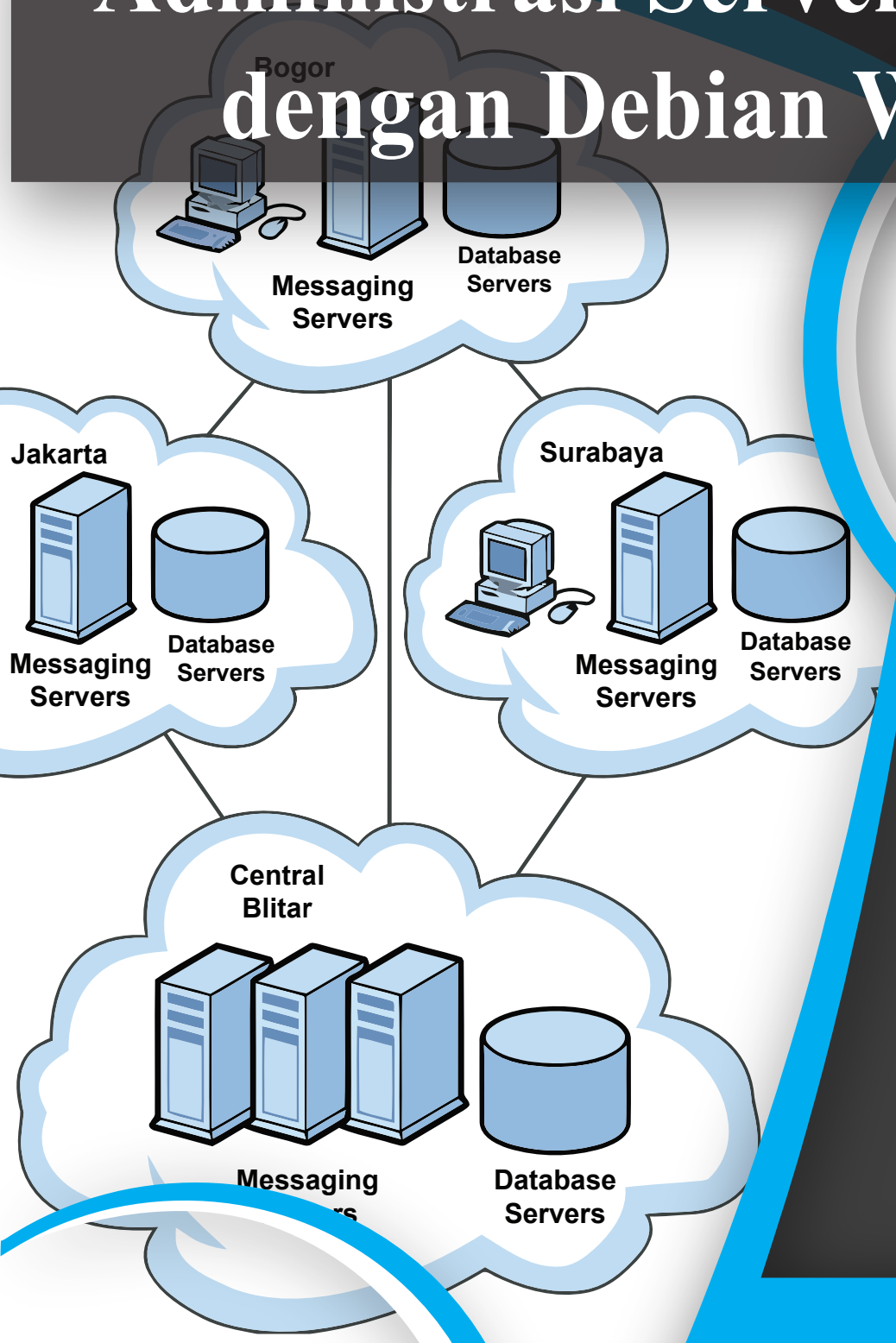
Administrasi Server Jaringan dengan Debian Wheezy



Komunitas IT SMKN 1 Nglegok

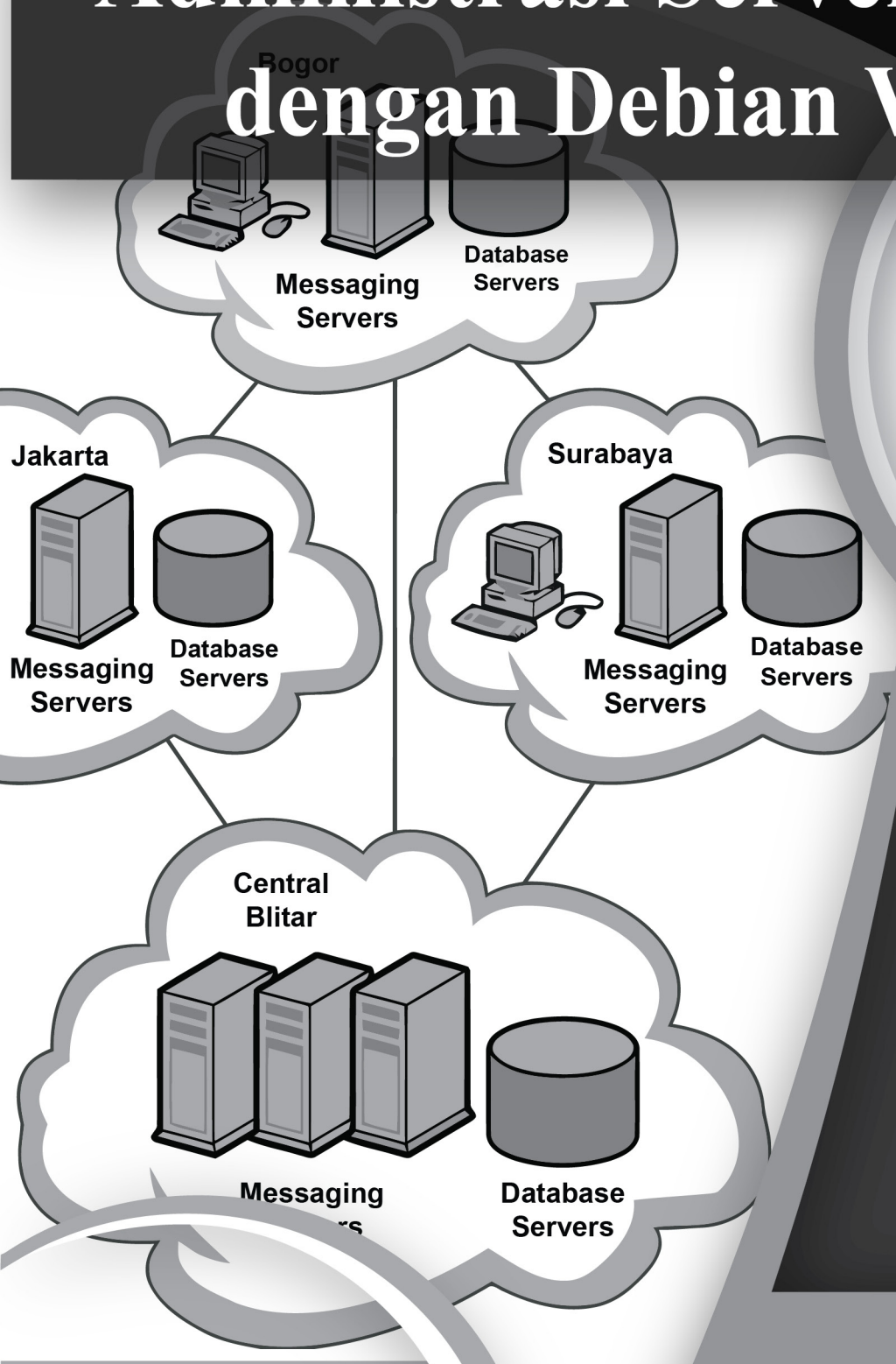
For KITS BOOK

Ahmad Rosid Komarudin

coretanbocahit.blogspot.com

kitsmkn1nglegok.com

Administrasi Server Jaringan dengan Debian Wheezy



Komunitas IT SMKN 1 Nglegok

For KITS BOOK

Ahmad Rosid Komarudin

coretanbocahit.blogspot.com

kitsmkn1nglegok.com

For KITS Book

Administrasi Server Jaringan dengan Debian Wheezy

Penulis : Ahmad Rosid Komarudin

Designer Cover : Beni Danang Nugraha

Pembimbing & Penasehat : Very Setiawan, S.Kom.

Supported by : Komunitas IT SMKN 1 Nglegok

Website : http://coretanbocahit.blogspot.com

http://kitsmkn1nglegok.com

Terbit : Juni 2016

Kata Pengantar

Alhamdulillahirobbil 'alamin, insyaallah dengan ridho Allah SWT ebook dengan judul "For KITS Book - Administrasi Server Jaringan dengan Debian Wheezy" ini selesai ditulis dan disusun.

Sholawat dan salam kupersembahkan untuk Baginda Rosul, Nabi Muhammad SAW Sang penerang dunia, penerang ilmu pengetahuan dan teknologi, serta panutan bagi umat di dunia.

Terimakasih untuk *Bapak* dan *Emak*, terimakasih atas pelajaran tentang misteri kehidupan yang tiada ahirnya ini, terimakasih sudah mengizinkanku untuk memilih jalan hidup yang kusukai, terimkasih sudah mengizinkanku berjam-jam didepan komputer setiap hari untuk menyelesaikan ebook ini. Semoga Allah memberi penghargaan yang setinggi-tingginya kepada *Panjenengan*, Amin..

Terimakasih untuk seluruh guru-guruku, guru pendidikan formal, guru ngaji, guru dalam kehidupan sehari-hari, guru-guruku di internet, seluruh orang yang telah memberi saya pandangan baru dalam kehidupan yang tidak mungkin kusebutkan satu persatu. Terimakasih sudah mendidik saya, terimakasih sudah menggembleng saya, terimakasih telah mengajarkan saya bagaimana cara belajar yang baik, terimakasih sudah mengajarkanku tentang hal-hal yang baru, terimakasih telah mengajarkan tentang indahnya berbagi.

Terimakasih terutama untuk guru besarku, Bapak Very Setiawan, S.Kom, terimakasih sudah mengajarkan saya tentang apa itu jaringan komputer, apa itu ip address, apa itu osi layer, apa itu server, terimakasih juga telah memberi nasehat-nasehat untukku, terimakasih sudah mengajariku hidup, terimakasih sudah meluangkan banyak waktu untuk kami (Komunitas IT). Terimakasih sudah mendukung kami, terimakasih sudah menjembatani kami untuk mengembangkan bakat kami, terimakasih sudah memberi kami tugas-tugas yang tak mudah, terimakasih sudah menghukum kami, terimakasih sudah mengingatkan kami saat kami salah, terimakasih pula telah menemani kami berjuang menggapai mimpi. ("*Maaf ya buat Nabila, Ayahnya jadi pulang malam terus... .. ..*").

Mas heru, mas farouq, mas fandi, mas naryo, mas ilham, mas topik, mas beni, mbak yesi, mbak aisy, mbak anis, seluruh seniorku yang tak mungkin kusebut semua. Terimakasih sudah membimbingku, terimakasih sudah membentuk mentalku, terimakasih sudah mau berbagi kepadaku.

Teman-teman seperjuanganku, bambang, nafi', nur zulianto, radit, nasrudin, fajar, rusmini, ebsi, elfinda, fina, lela, leni, nadia, nur fitria, melinda. Terimakasih sudah menemaniku berjuang, jangan pernah lupakan KITS!,

Terimakasih juga buat adik-adik kelasku yang tak mungkin kusebut satu-persatu, terimakasih sudah mendoakanku, terimakasih sudah mendukungku, terimakasih pula telah mengajarkanku bagaimana indahnya keluarga KITS ini.

Terimakasih kepada pembaca sekalian, yang telah bersedia menggunakan ebook ini sebagai referensi untuk belajar. Semoga ebook ini bisa bermanfaat untuk pembaca sekalian. Saya tidak minta imbalan apapun dari ebook ini, saya hanya minta tolong untuk senantiasa mengingatku, orang tuaku, guru-guruku, senior-seniorku, teman-temanku, dan seluruh guru-guru anda dalam doa anda. Semoga ilmu yang telah diajarkan oleh guru kita, ilmu yang kita pelajari, bisa bermanfaat, Amin...

Pesan saya untuk pembaca sekalian, jika anda sudah pintar, anda sudah bisa akan suatu hal, ada dua hal yang perlu anda perhatikan. "Pertama", jangan pernah lupakan jasa guru anda, jangan sekali-kali anda mengatakan bahwa anda belajar secara OTODIDAK, karena sesungguhnya anda tidaklah belajar sendiri! Banyak sekali guru-guru yang telah mengarahkan anda dalam belajar, banyak sekali guru-guru yang telah membuat artikel diinternet dan anda menggunakan artikel itu untuk belajar. "Kedua", Jangan lupa untuk mengajarkan kembali apa yang telah anda pelajari, karena jika anda tidak mengajarkannya kembali, itu artinya anda memutus rantai amal baik dari guru anda. Semoga Allah SWT senantiasa menjaga kita... Amin,,,

Ahmad Rosid Komarudin
ahmadrosid30121997@gmail.com

Daftar Isi

Memahami Buku Ini,?

Untuk memermudah pembaca dalam memahami buku ini sekaligus menghindari kesalahfahaman, kita akan membuat beberapa kesepakatan.

Buku ini disusun dengan tujuan memberikan referensi belajar menggunakan alat seminim mungkin. Dalam hal ini, kita akan bisa mempraktikan seluruh isi buku ini hanya dengan bantuan sebuah komputer dan sebuah aplikasi virtualisasi. Tentunya ada beberapa bab pengecualian yang memerlukan software tambahan, alat tambahan, atau bahkan komputer tambahan.

Sistem operasi server yang digunakan pada buku ini adalah Debian Wheezy dengan versi 7.6. Sedangkan sistem operasi client yang akan digunakan adalah ubuntu 14.04 32 bit. Namun akan ada pengecualian pada beberapa bab, yaitu jika ada perbedaan yang jauh antara client ubuntu dan windows, maka buku ini juga akan membahas penggunakan sistem operasi windows sebagai client. Versi windows yang akan digunakan adalah windows 7.

Nantinya, dalam praktik akan sering menggunakan sebuah perintah dasar, baik untuk melakukan konfigurasi di server, ataupun melakukan pengujian dari client. Oleh karena itu, kita akan menggunakan latar belakang hitam untuk perintah pada server dan latar belakang putih untuk perintah pada client. Hal ini dimaksudkan agar ada pembeda antara perintah yang dijalankan pada server dan perintah yang dijalankan pada client.

Dalam setiap bab atau sub bab, akan ada topologi yang dapat memermudah pembaca dalam memahami konsep materi. Sehingga pembaca dapat mengetahui konsep secara real meskipun dalam praktik hanya menggunakan aplikasi virtualisasi.

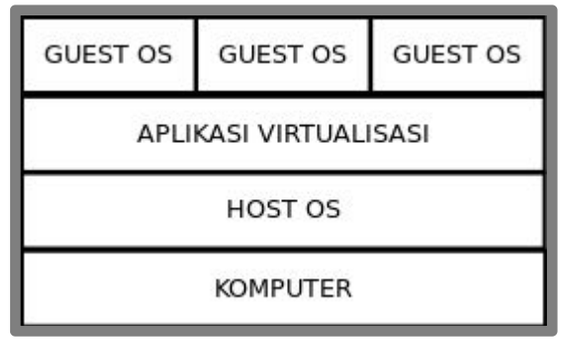
Jika ada suatu teks, syntak, ataupun script yang tidak terlalu penting dan terlalu panjang saat pembahasan, maka teks, syntak, ataupun script tersebut akan dihapus dan hanya diganti dengan tanda titik-tik (.....).

Tentu tidak mungkin untuk membahas materi dengan sangat detail dalam buku ini. Untuk itu, pembaca diharapkan bisa mempelajari lebih lanjut dari apa yang disampaikan pada buku ini dengan mencoba hal-hal baru sendiri (*Practice Make Better,* Kata salah satu guru besar saya).

Bab 1
Persiapan Praktik

Sudah disebutkan sebelumnya, bahwa kita hanya akan menggunakan sebuah komputer yang telah terinstall sistem operasi ubuntu 14.04 32 bit. Untuk membantu praktik, kita akan menginstall aplikasi virtualbox.

Virtualbox adalah salah satu aplikasi virtualiasasi yang sangat terkenal. Dengan bantuan aplikasi ini, kita bisa menjalankan lebih dari 1 sistem operasi secara bersamaan dalam sebuah komputer. Tentunya jumlah sistem operasi yang dapat dijalankan terbatas sesuai dengan spesifikasi komputer yang digunakan. Jika komputer yang digunakan mempunyai spesifikasi tinggi, akan semakin banyak sistem operasi yang bisa dijalankan, begitu juga sebaliknya. Berikut merupakan gambaran umum penerapan aplikasi virtualisasi dalam sebuah komputer.



Gambar 1.1 Konsep aplikasi virtualisasi

Dari gambar 1.1 tersebut, terlihat bahwa pondasi terbawah adalah sebuah komputer. Kemudian diatasnya adah sebuah host os, dimana host os ini merupakan sistem operasi yang diinstall didalam komputer kita. Selanjutnya, didalam sistem operasi kita (host os) akan diinstall sebuah aplikasi virtualisasi, yaitu virtualbox itu sendiri. Sedangkan didalam virtualbox kita baru akan menginstall sistem operasi virtual, atau yang biasa disebut dengan guest os.

Install Virtualbox di Ubuntu 14.04

Kita sudah memahami konsep dasar mengenai penggunaan aplikasi virtualiasasi. Selanjutnya kita akan belajar tentang cara menginstall virtualbox di sistem operasi ubuntu 14.04.

Untuk melakukannya, disarankan komputer kita terhubung dengan internet. Sebenarnya ada cara lain untuk menginstall virtualbox di ubuntu tanpa harus terhubung dengan internet. Namun cara termudah menurut saya adalah dengan terhubung dengan internet. Tenang saja, setelah kita menginstall virtualbox, kita bisa belajar tanpa ada akses internet (kecuali dibeberapa bab).

Buka terminal dengan kombinasi ctrl+alt+t, kemudian ketikkan perintah sebagai berikut.

```
admin@ubuntu:~$ sudo apt-get install virtualbox
[sudo] password for admin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  virtualbox-dkms virtualbox-qt
The following packages will be upgraded:
  virtualbox virtualbox-dkms virtualbox-qt
3 upgraded, 0 newly installed, 0 to remove and 722 not upgraded.
Need to get 18,6 MB of archives.
After this operation, 348 kB disk space will be freed.
Do you want to continue? [Y/n] Y
```

Gambar 1.2 Installasi virtualbox di Ubuntu 14.04

Selanjutnya, proses installasi akan berjalan. Berikut tampilan virtualbox setelah proses installasi selesai.

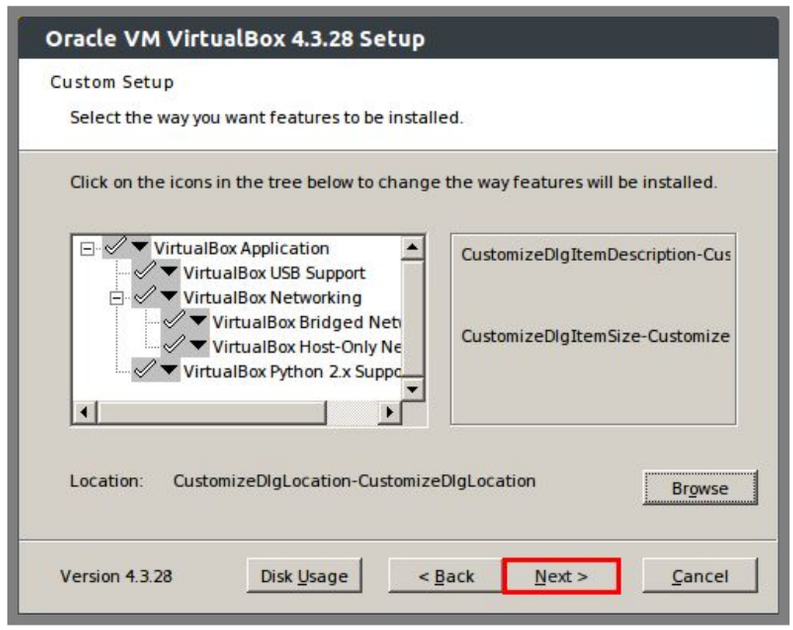
Gambar 1.3 Halaman utama virtualbox

Install Virtualbox di Windows 7

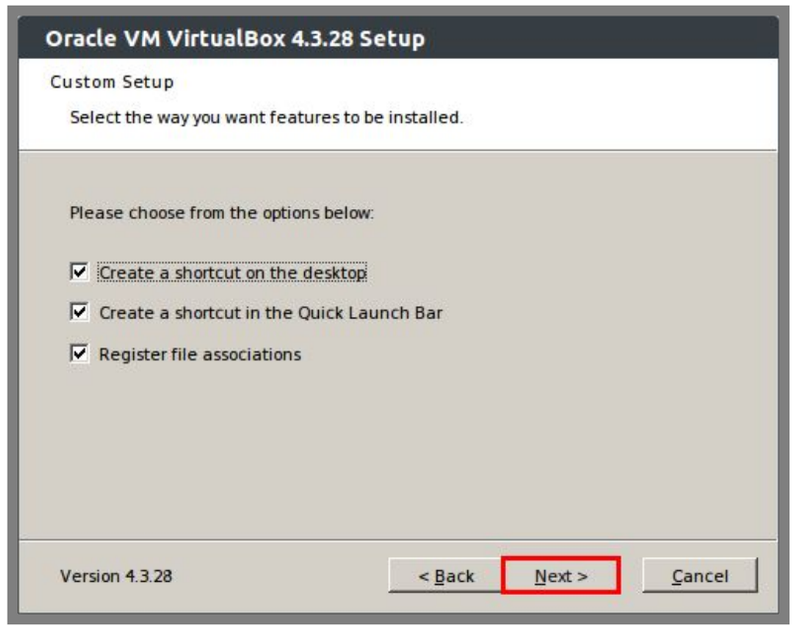
Karena ada perbedaan yang signifikan antara proses installasi virtualbox di ubuntu dan windows, maka akan dibahas juga bagaimana cara install virtualbox di windows.

Hal pertama yang harus dilakukan adalah download installer virtualbox dari website resmi virtualbox di www.virtualbox.org. Selanjutnya jalankan file installer tersebut.

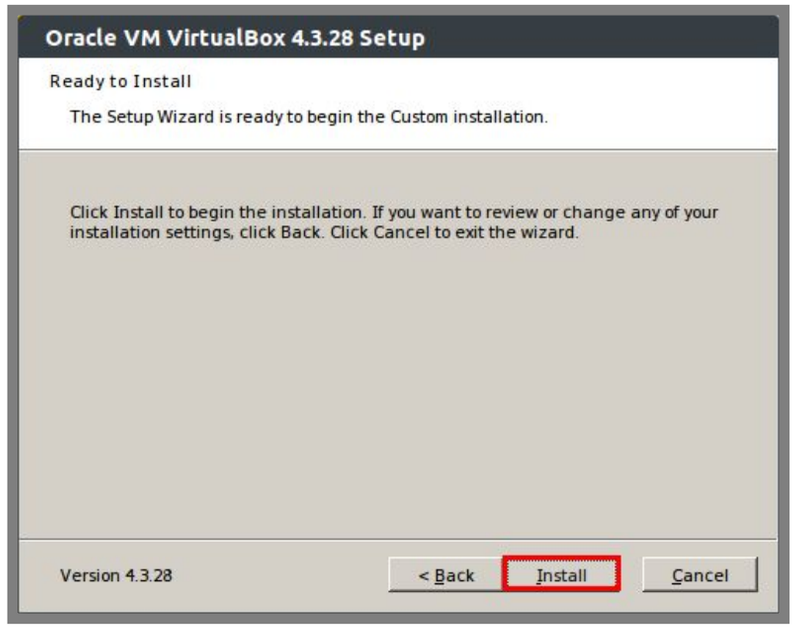
Gambar 1.4 Proses installasi virtualbox di windows



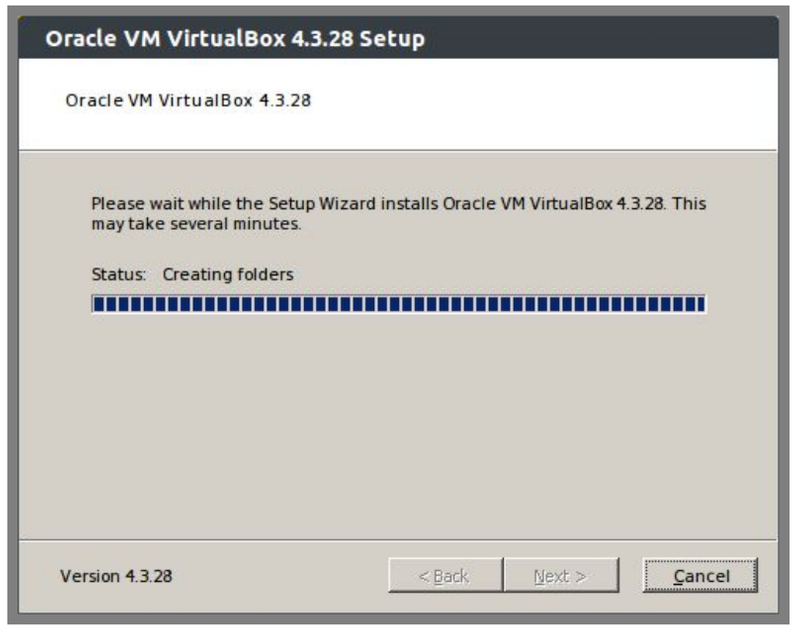
Gambar 1.5 Proses installasi virtualbox di windows



Gambar 1.6 Proses installasi virtualbox di windows



Gambar 1.6 Proses installasi virtualbox di windows



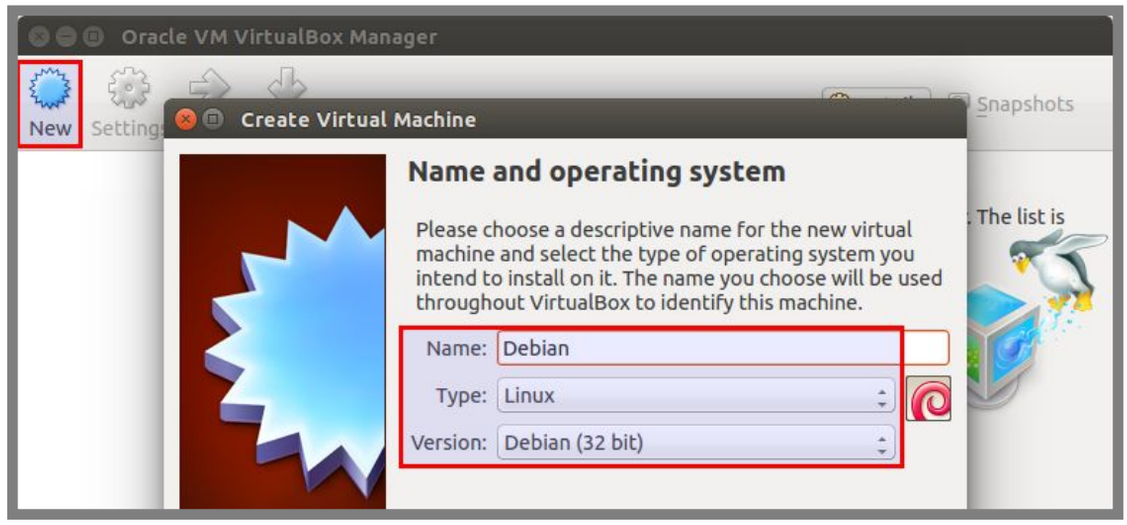
Gambar 1.7 Proses installasi virtualbox di windows

Gambar 1.8 Proses installasi virtualbox di windows

Membuat Virtual Machine

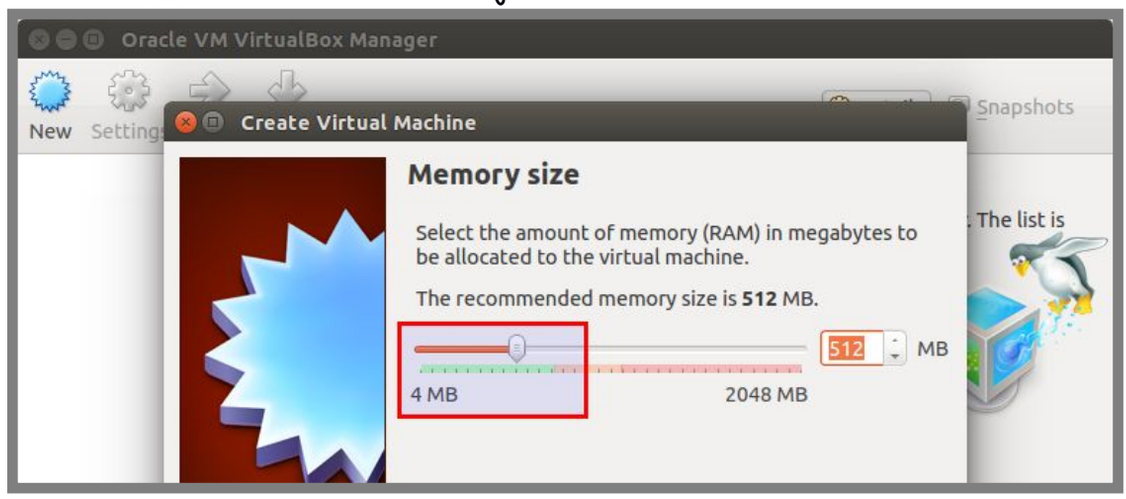
Pada dasarnya, sebuah sistem operasi diinstall didalam sebuah komputer. Konsep tersebut berlaku juga pada aplikasi virtualisasi. Jadii sebelum memulai installasi sebuah sistem operasi virtual, kita harus membuat sebuah komputer virtual. Dimana nantinya kita akan menginstall sistem operasi virtual didalam komputer virtual tersebut. Berikut langkah-langkahnya:

Klik shortcut New, kemudian isikan name, type, dan version sesuai dengan sistem operasi yang akan diinstall



Gambar 1.9 Menentukan nama dan type virtual machine

Konfigurasi ukuran RAM sesuai dengan kebutuhan. Pastikan tidak menyetting ukuran RAM melebihi batas warna hijau

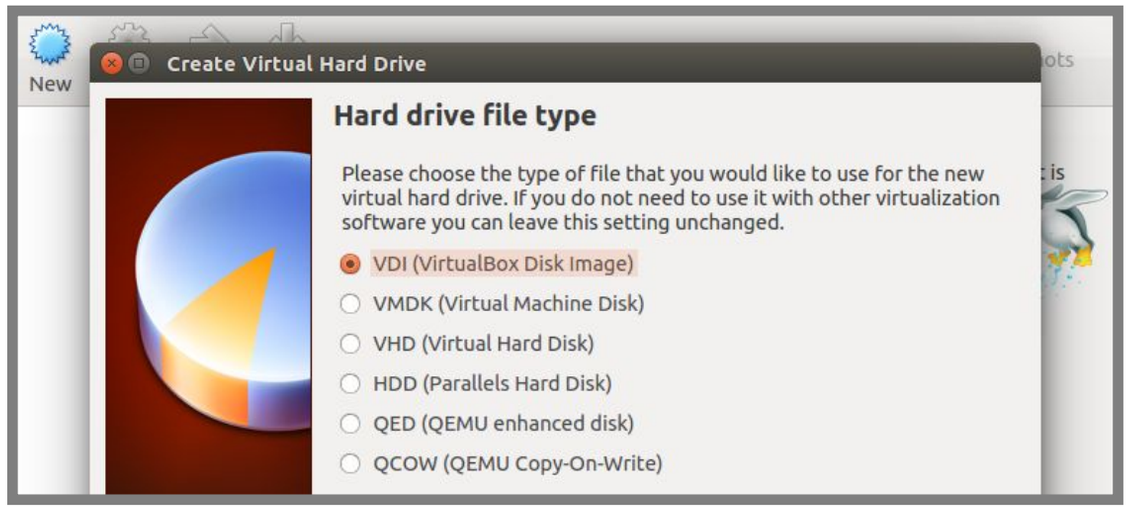


Gambar 1.10 Menentukan ukuran RAM pada virtual machine

Pilih create virtual harddrive now, untuk membuat harddisk virtual

Gambar 1.11 Membuat harddisk baru untuk virtual machine

Tentukan extensi file virtual harddisk yang diinginkan, jika belum terlalu mengerti bisa dipilih yang default saja, yaitu vdi

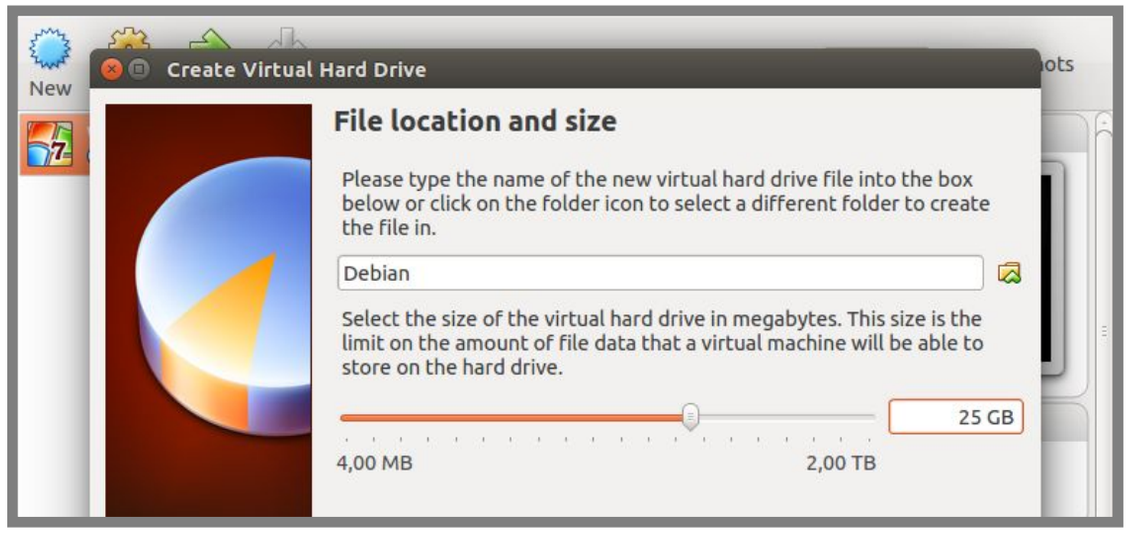


Gambar 1.12 Menentukan tipe virtual harddisk yang ingin dibuat

Pilih dynamiic allocated

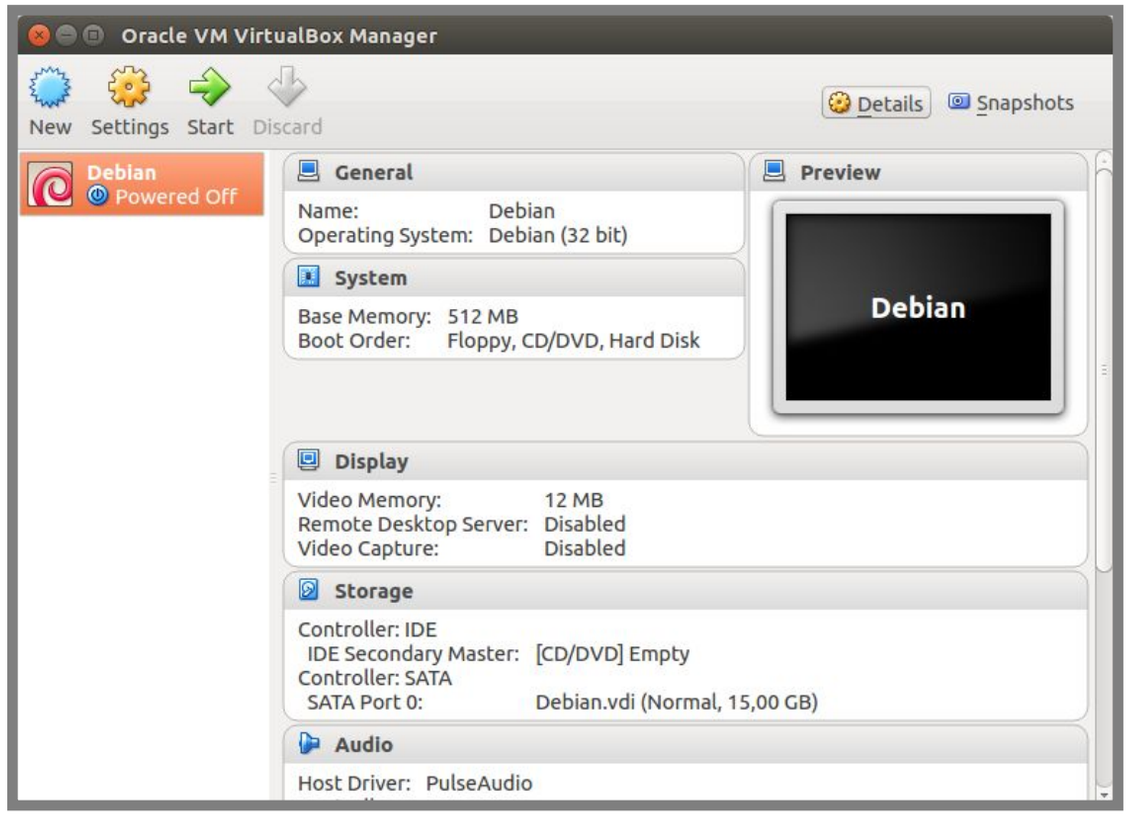
Gambar 1.13 Menentukan tipe penyimpanan harddisk virtual machine

Tentukan ukuran harddisk virtual yang akan dibuat



Gambar 1.14 Menentukan ukuran harddisk virtual

Sampai saat ini kita telah selesai membuat sebuah virtual machine. Berikut tampilan saat telah selesai



Gambar 1.15 Tampilan virtual machine yang telah berhasil dibuat

---END OF CHAPTER---

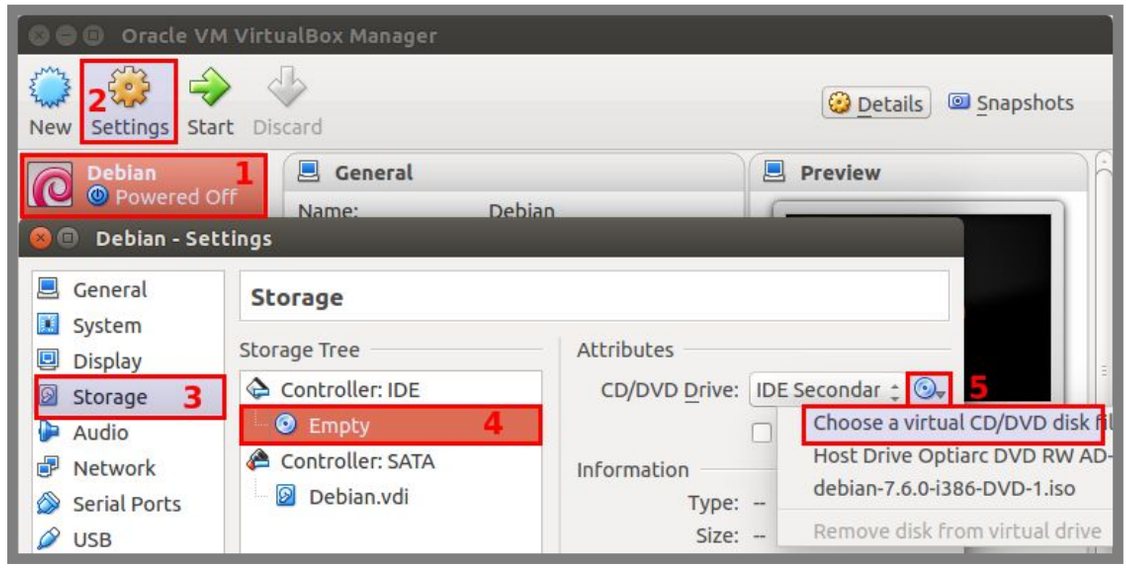
Bab 2

Installasi Sistem Operasi Debian 7.6

Telah disepakati sebelumnya, bahwa sistem operasi server yang digunakan adalah debian dengan versi 7.6.

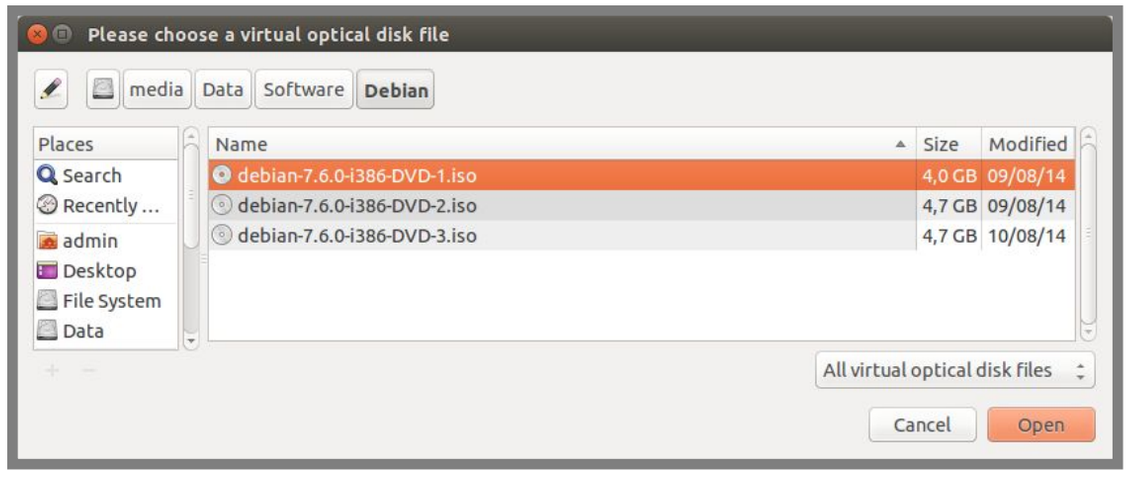
Kita telah membuat sebuah virtual machine di bab sebelumnya. Di bab ini kita akan membahas tentang langkah-langkah installasi sistem operasi server berbasis linux debian. Berikut langkah-langkahnya:

Konfigurasi cd/dvd untuk booting installasi



Gambar 2.1 Konfigurasi installer

Cari lokasi file iso debian berada, pilih file iso dvd 1

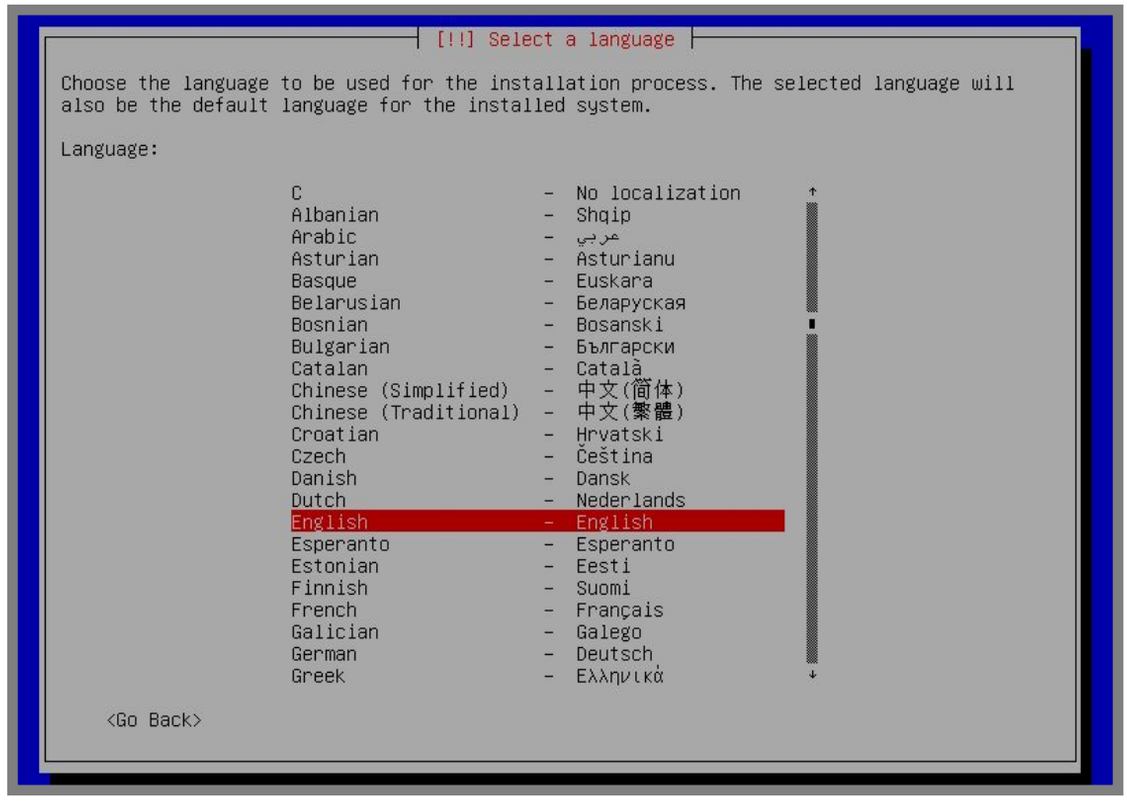


Gambar 2.2 Memilih file iso untuk installasi

Selanjutnya jalankan virtual machine dengan menekan tombol start, berikut tampilan pertama saat komputer booting dari installer debian 7.6. Jika ingin menginstall dengan metode text kita bisa pilih opsi install, namun jika ingin menginstall dengan metode GUI bisa memilih opsi graphical install. Pada buku ini akan dibahas installasi metode text

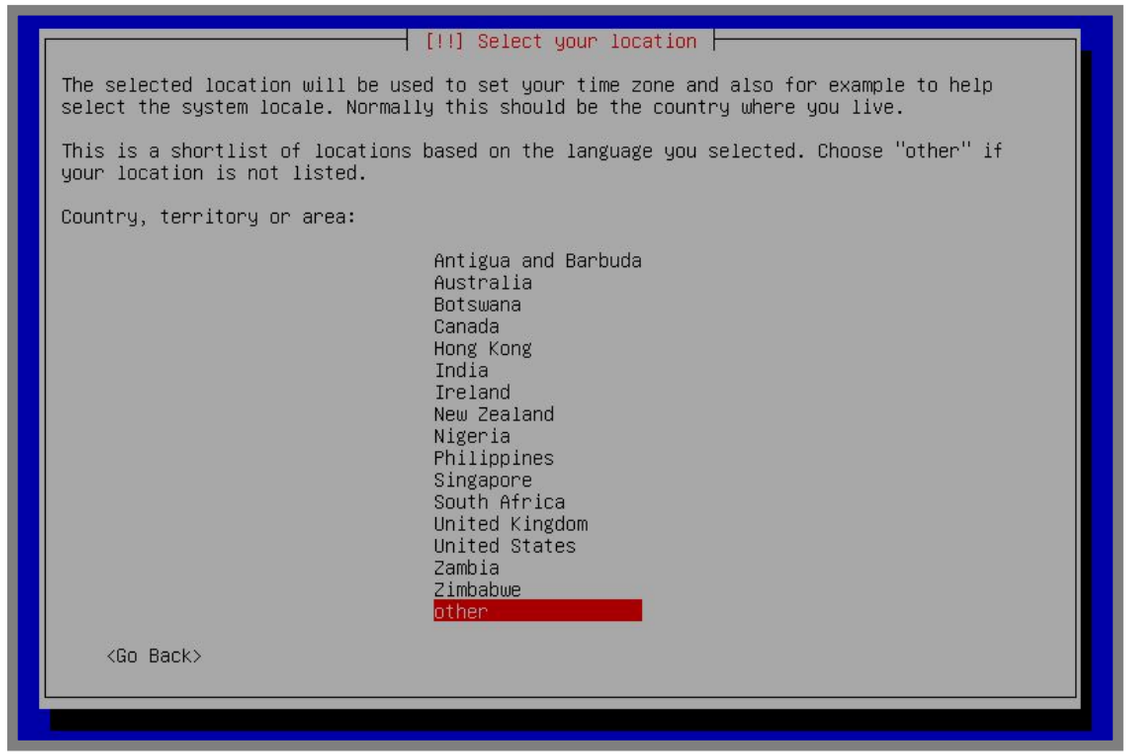
Gambar 2.3 Tampilan pertama proses instllasi

Pilih bahasa yang ingin digunakan untuk sistem operasi server



Gambar 2.4 Pemilihan bahasa untuk proses installasi dan sistem operasi

Pilih lokasi tempat kita tinggal. Karena indonesia tidak ada, kita pilih other

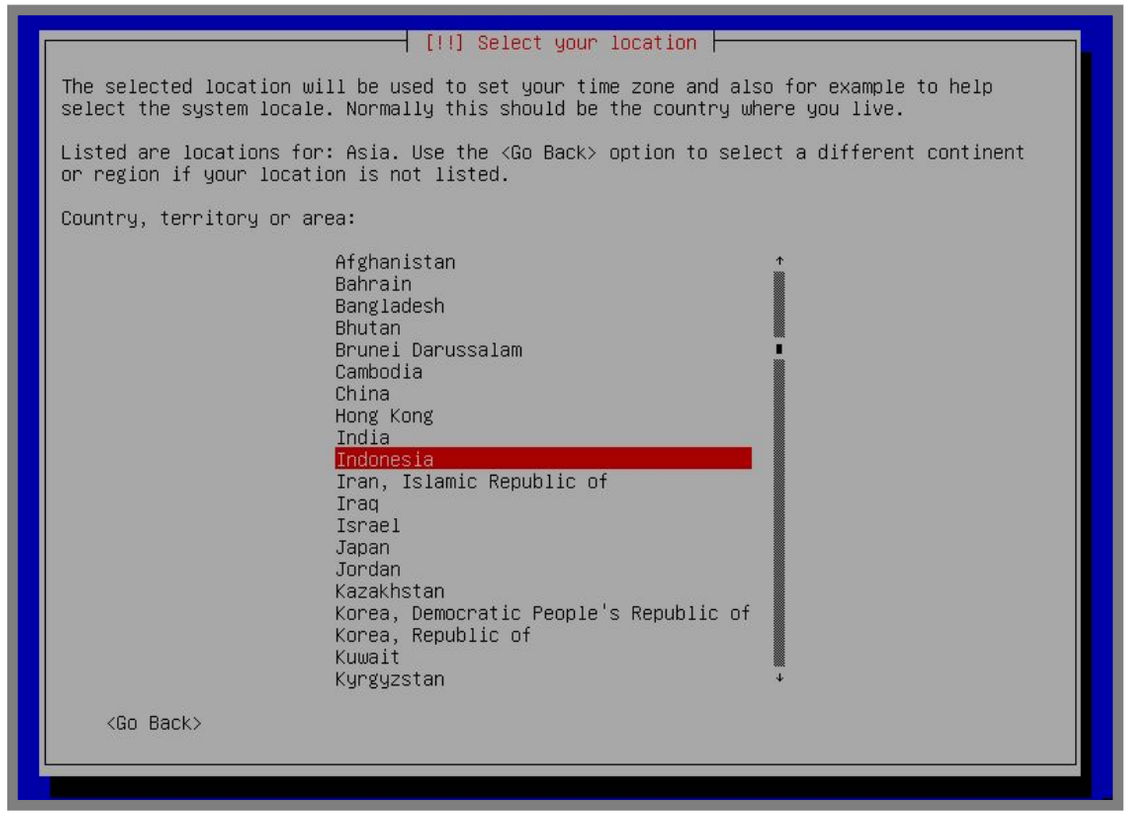


Gambar 2.5 Pemilihan negara

Selanjutnya pilih Asia

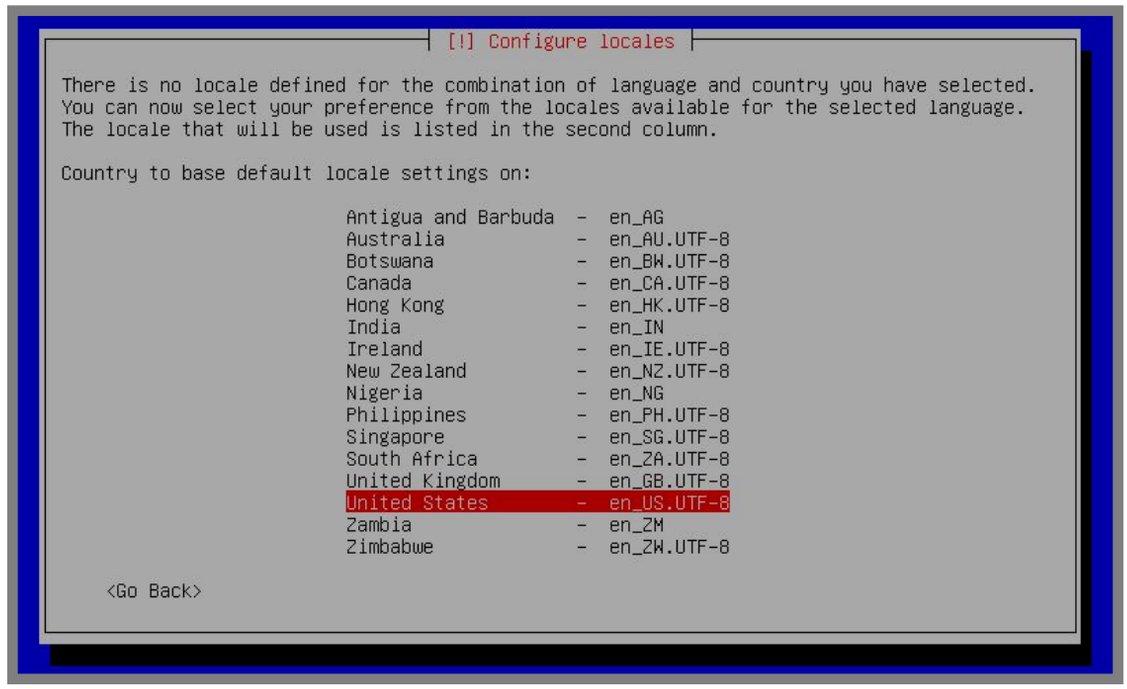
Gambar 2.6 Pemilihan benua

Pilih Indonesia



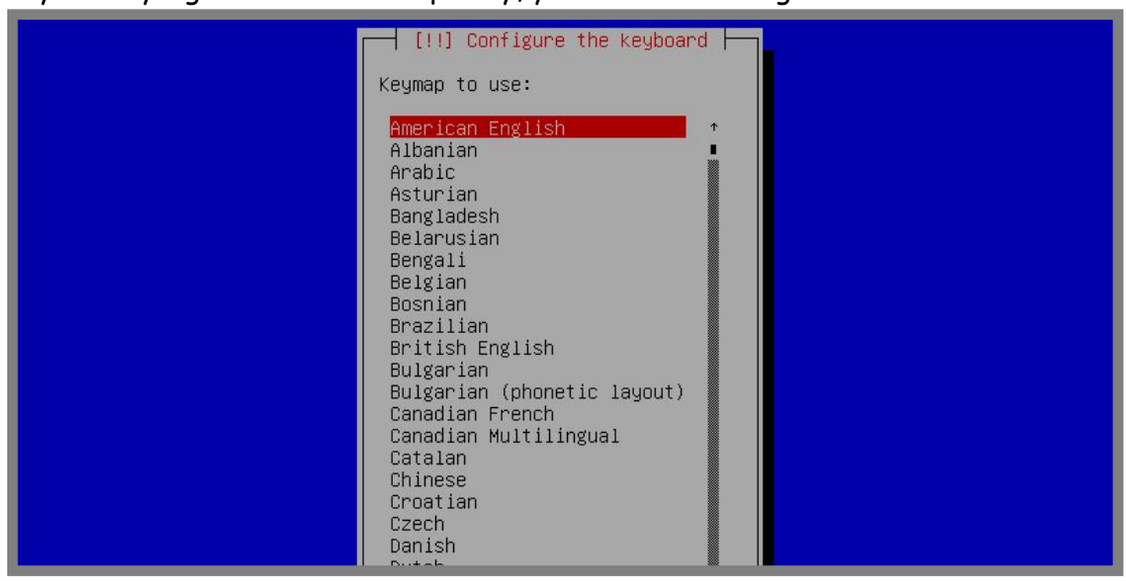
Gambar 2.7 Pemilihan negara

Pilih united states



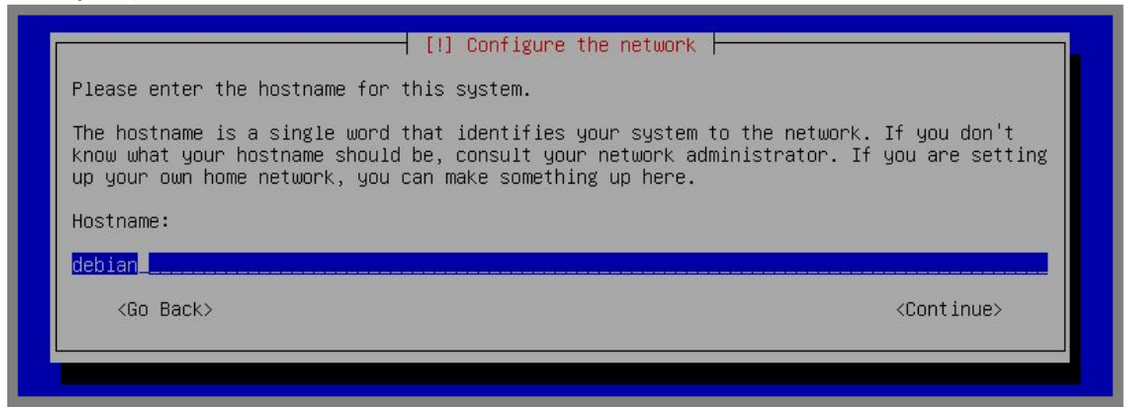
Gambar 2.8 Pemilihan negara untuk penggunaan bahasa

Pilih type kyboard sesuai dengan yang digunakan. Umumnya di Indonesia, type keyboard yang beredar adalah qwerty, yaitu American English



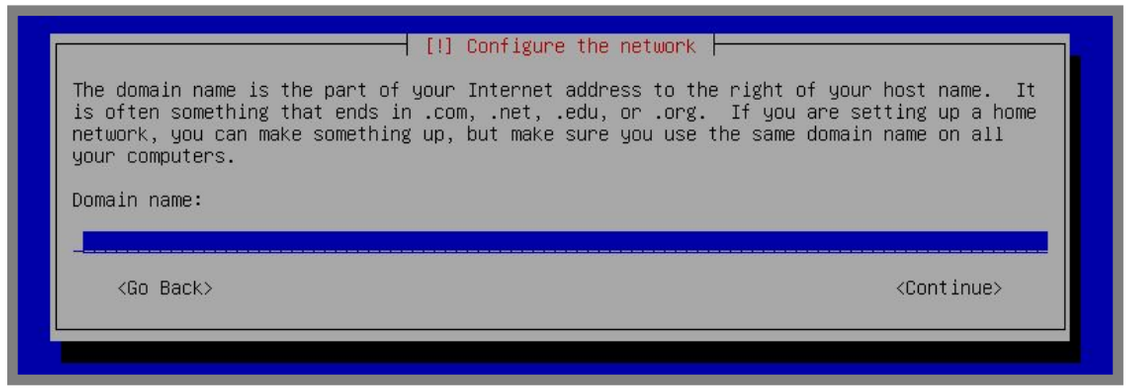
Gambar 2.9 Pemilihan type kyboard

Isikan hostname sesuai dengan yang diinginkan. Jika belum terlalu faham, bisa diisi default saja, yaitu debian (kita akan bahas materi hostname di bab selanjutnya). Kemudian enter



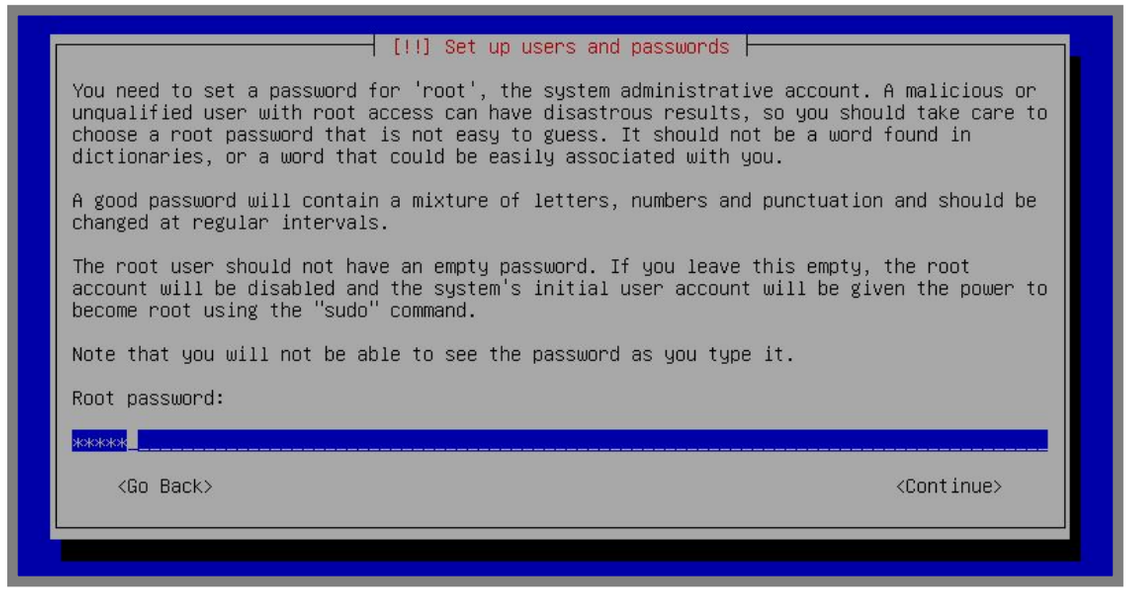
Gambar 2.10 Konfigurasi hostname

Isikan domain name, bisa dikosongkan saja jika belum tahu apa itu domain name (kita akan bahas materi tentang domain di bab selanjutnya). Kemudian enter



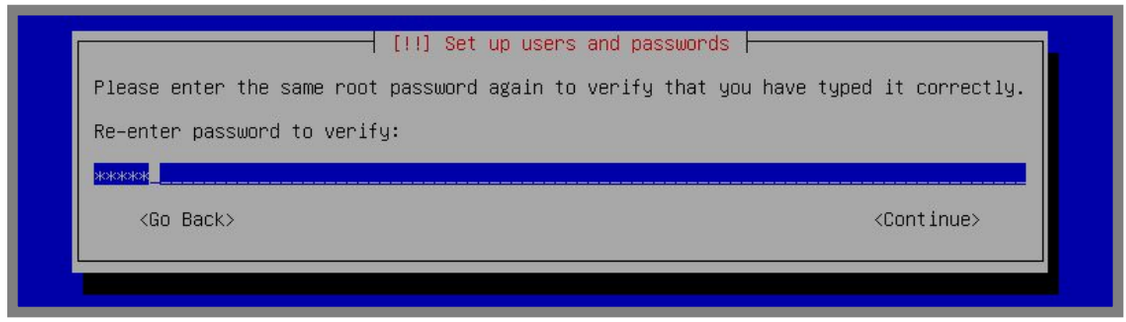
Gambar 2.11 Konfigurasi domain name

Isikan password untuk user root. Kita akan membahas lebih lanjut tentang apa itu user root pada bab selanjutnya, untuk saat ini kita hanya perlu memasukkan password saja. Kemudian enter



Gambar 2.12 Konfigurasi root password

Masukkan password untuk root sekali lagi



Gambar 2.13 Konfigurasi root password

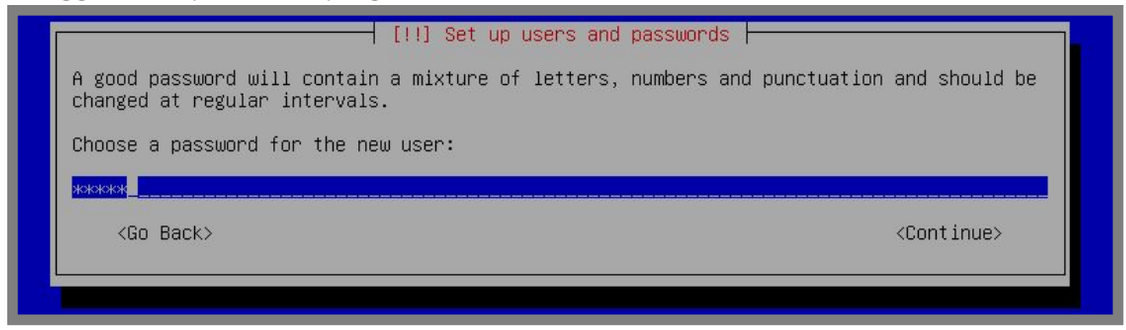
Isikan nama lengkap untuk user baru

Gambar 2.14 Nama lengkap untuk user baru yang akan dibuat

Isikan username untuk user baru

Gambar 2.15 Username untuk user baru yang akan dibuat

Masukkan password untuk user baru yang akan dibuat. Disarankan untuk menggunakan password yang berbeda dari user root.

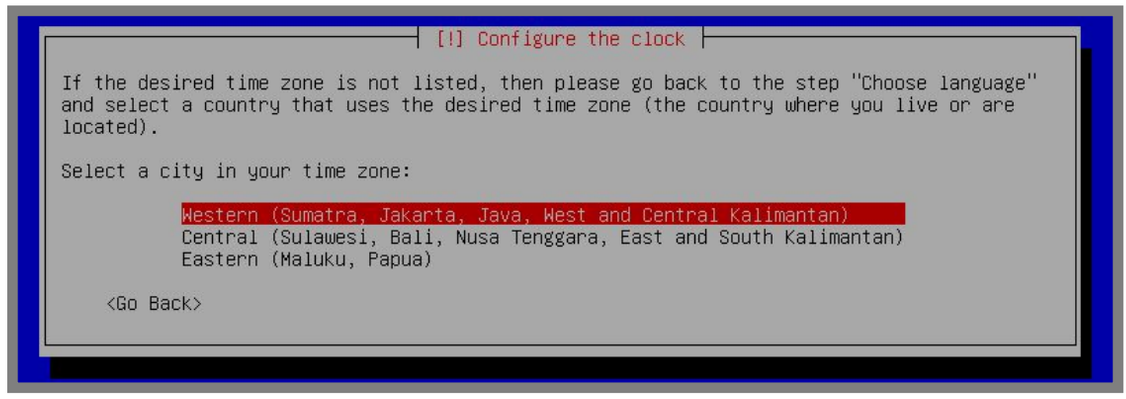


Gambar 2.16 Konfigurasi password untuk user baru

Masukkan password untuk user baru sekali lagi

Gambar 2.17 Konfigurasi password untk user baru

Pilih lokasi tempat tinggal anda untuk menentukan konfigurasi waktu



Gambar 2.18 Pemilihan lokasi waktu

Langkah selanjutnya adalah proses partisi. Secara garis besar, ada dua metode partisi yang umum digunakan saat installasi linux, yaitu manual dan otomatis.

Jika kita memilih otomatis, maka seluruh file yang ada di harddisk akan diformat, dan selanjutnya akan dibuatkan partisi dengan jumlah dan besar yang ditentukan oleh sistem. Namun jika memilih metode manual, kita akan lebih bebas menentukan jumlah partisi dan besar masing-masing partisi sesuai dengan yang kita butuhkan.

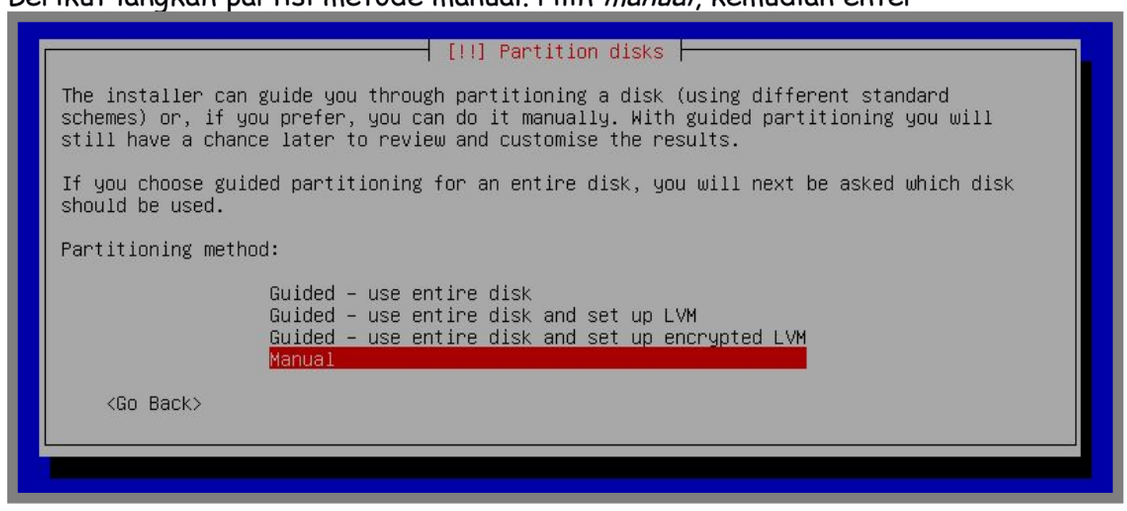
Sistem operasi linux memerlukan minimal dua partisi, yaitu swap dan root. Partisi swap nantinya berfungsi sebagai virtual memory, dalam artian partisi ini akan menjadi RAM tambahan bagi komputer. Partisi ini, umumnya berukuran 2x (dua kali) ukuran RAM. Selanjutnya partisi root berfungsi untuk menyimpan seluruh file sistem milik sistem operasi linux. Dalam linux, partisi root disimbolkan dengan back slash (/).

Selain dua partisi yang wajib ada pad linux (root dan swap), kita bisa membuat beberapa partisi opsional, seperti /usr, /home, /var, dll. Pada bab ini kita akan praktik membuat tiga partisi, yaitu root (/), swap, dan /home.

Masing-masing partisi opsional mempunyai fungsi yang berbeda-beda. Seperti /home digunakan untuk menyimpan data data milik user, /var digunakan untuk menyimpan beberapa file system, dll. Pembahasan materi tentang fungsi masing-masing partisi akan dijelaskan lebih lanjut pada bab selanjutnya.

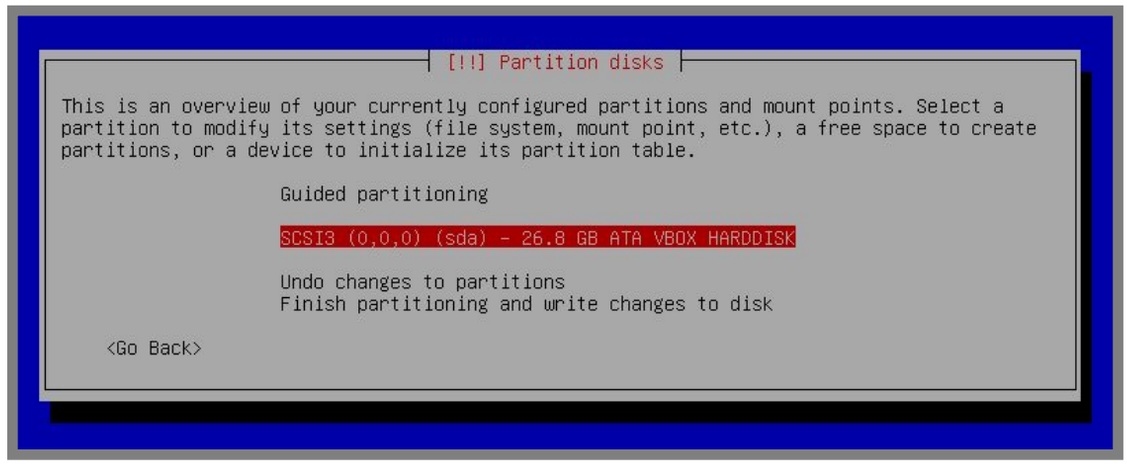
Sebenarnya untuk pemula, disarankan untuk membuat dua partisi saja, yaitu root (/) dan swap. Namun karena nantinya ada sebuah pembahasan materi yang membutuhkan partisi /home, jadi kita menambahkan sebuah partisi opsional yang akan kita buat, yaitu /home.

Berikut langkah partisi metode manual. Pilih *manual*, kemudian enter



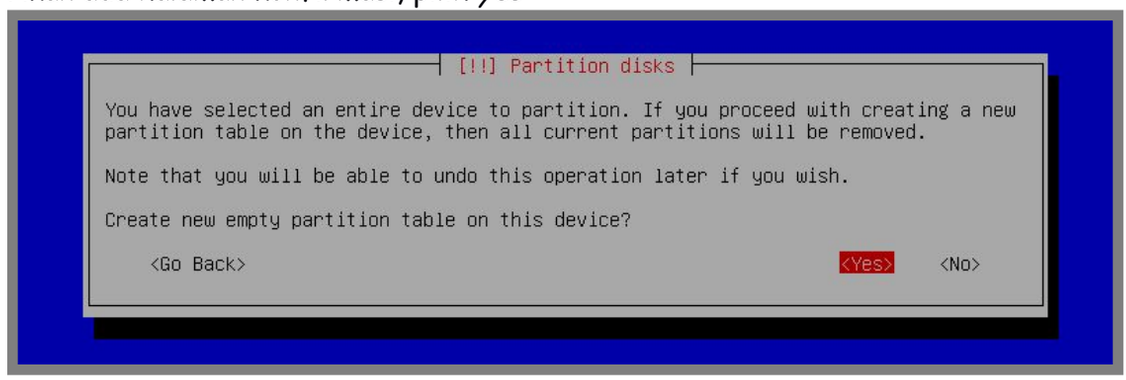
Gambar 2.19 Pemilihan metode partisi

Pilih harddisk yang akan dipartisi, kemudian enter



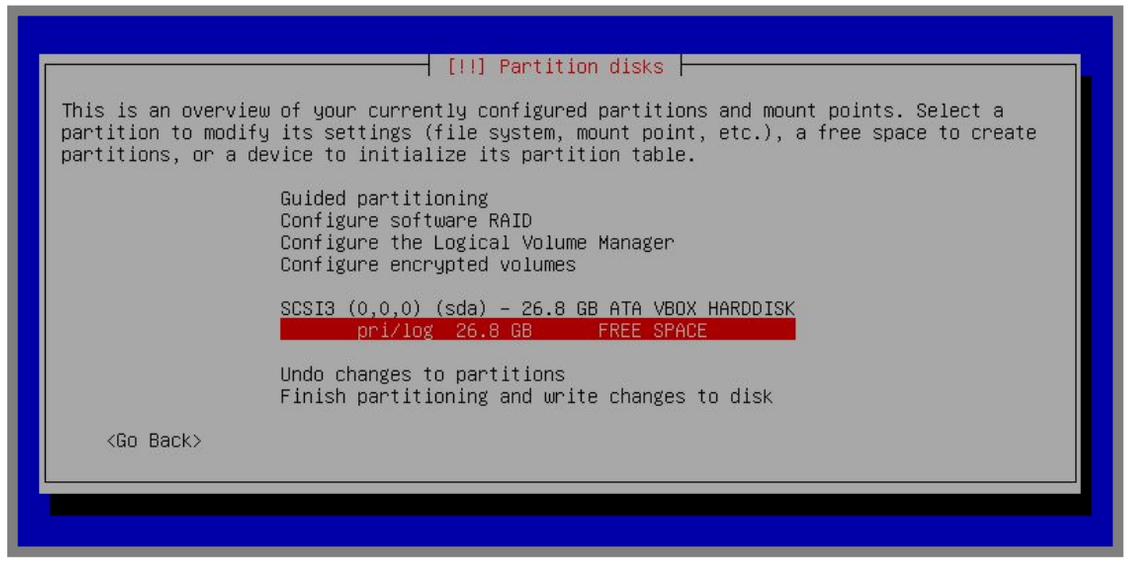
Gambar 2.20 Pemilihan harddisk yang akan dipartisi

Akan ada halaman konfirmasi, pilih *yes*



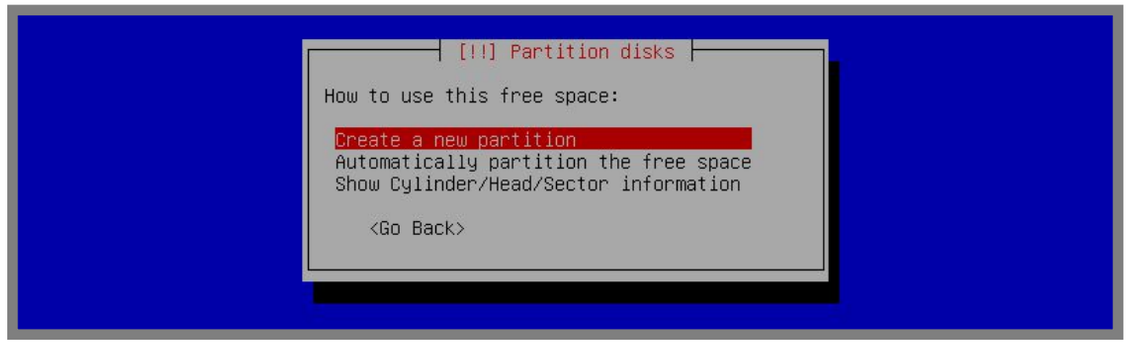
Gambar 2.21 Konfirmasi pembuatan tabel partisi

Pilih pada tabel partisi dengan label *free space* untuk membuat partisi baru, enter



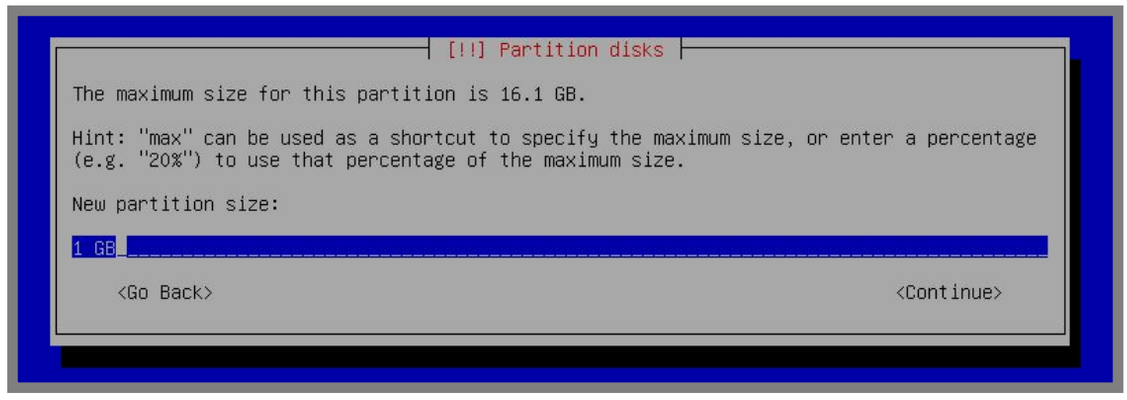
Gambar 2.22 Proses pembuatan partisi baru

Pilih *create a new partition* untuk membuat partisi



Gambar 2.23 Proses pembuatan partisi baru

Masukkan ukuran partisi untuk swap, karena RAM virtual machine tadi 512M, maka ukuran partisi swap yang ideal adalah 1G, enter



Gambar 2.24 Penentuan ukuran partisi swap

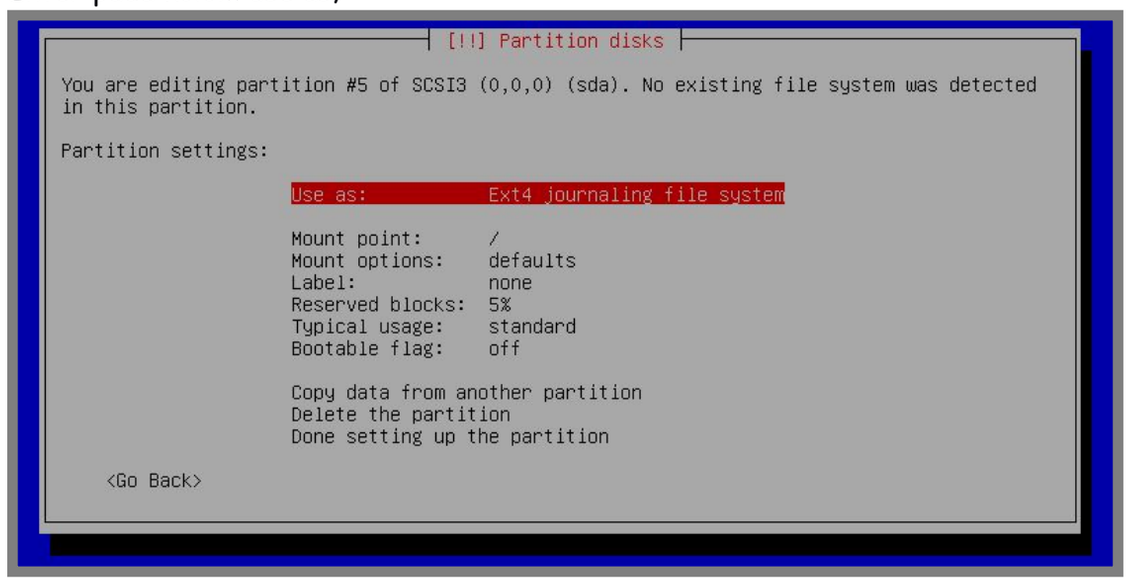
Untuk partisi swap, kita cukup menggunakan type *logical*

Gambar 2.25 Pemilihan tipe partisi swap

Untuk location, gunakan *beginning*

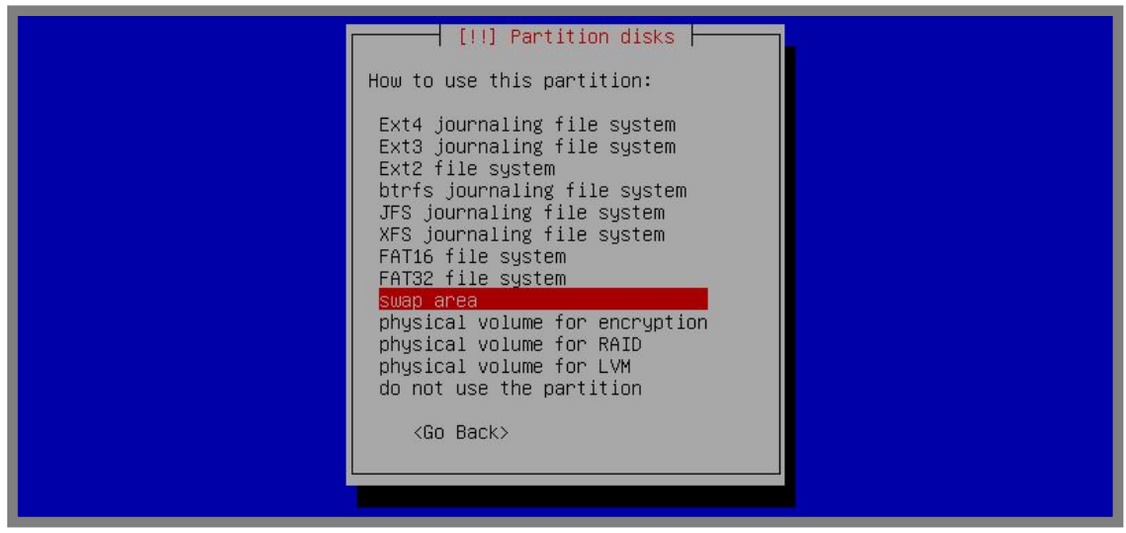
Gambar 2.26 Pemilihan lokasi partisi swap

Enter pada kolom *use as*,



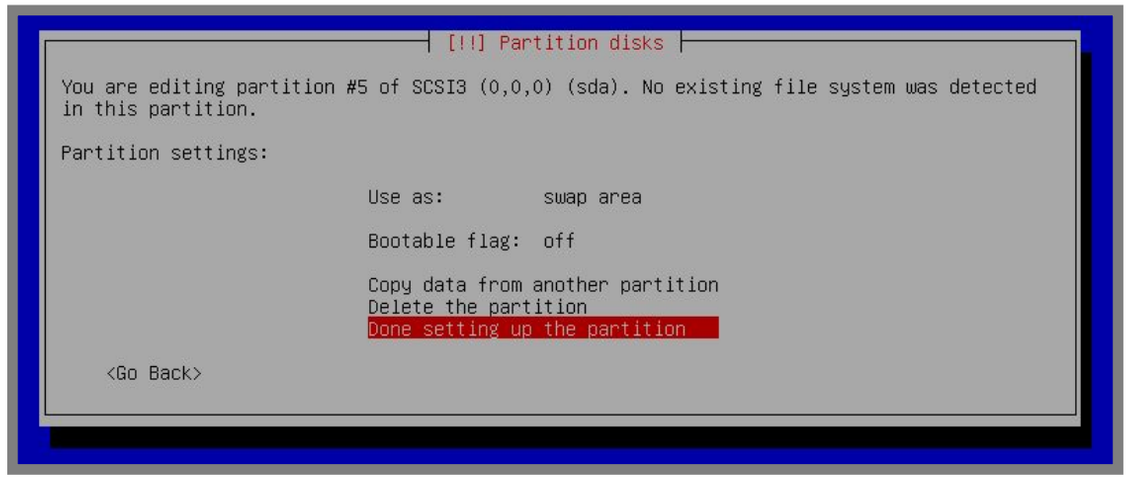
Gambar 2.27 Merubah partisi

Pilih *swap area*, enter



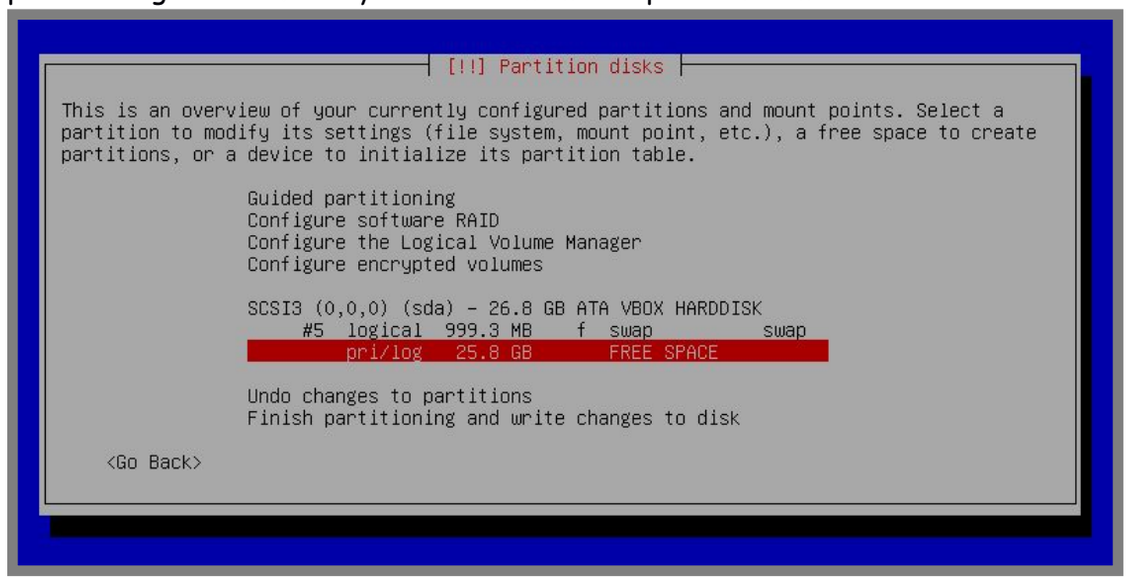
Gambar 2.28 Merubah partisi menjadi swap

Terahir pilih *done seting up the partition*



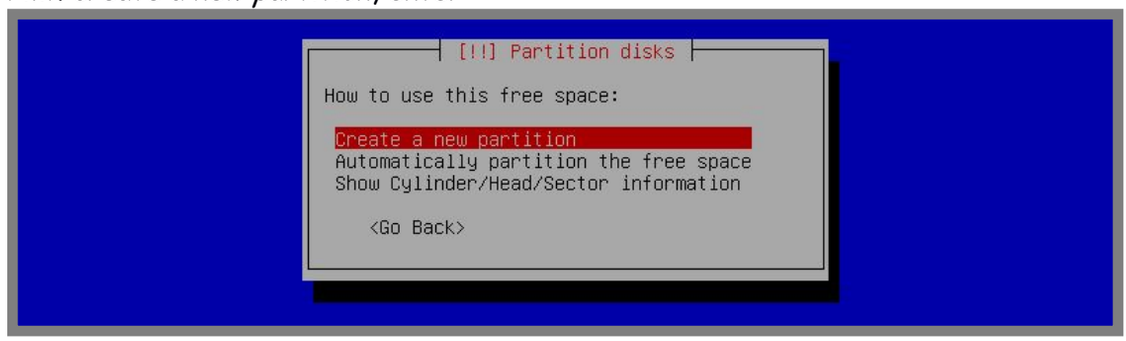
Gambar 2.29 Proses ahir pembuatan partisi swap

Sampai saat ini kita sudah selesai membuat partisi swap, selanjutnya pilih tabel partisi dengan label *free space* untuk membuat partisi */home*



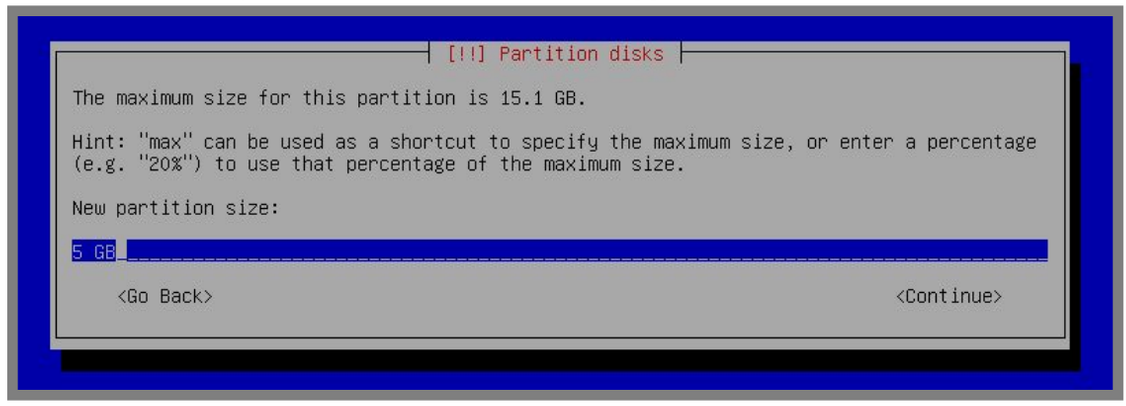
Gambar 2.30 Proses awal pembuatan partisi /home

Pilih *create a new partition*, enter



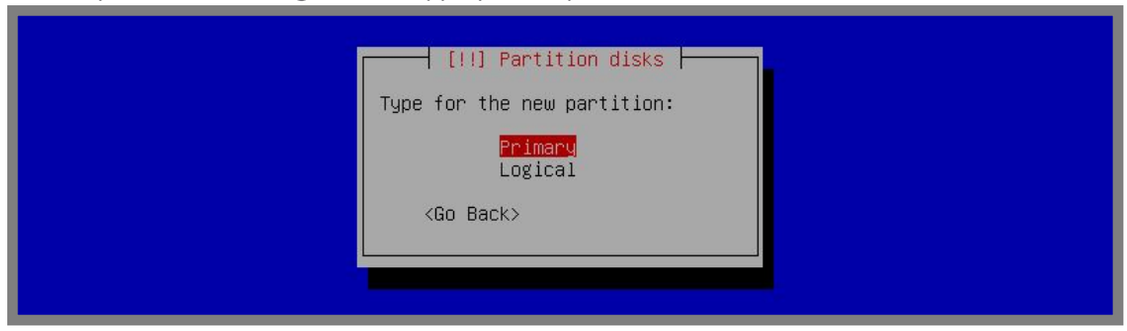
Gambar 2.31 Proses pembuatan partisi /home

Tentukan ukuran partisi /home, disini kita menggunakan 5G



Gambar 2.32 Menentukan ukuran partisi /home

Untuk partisi /home, gunakan type *primary*,

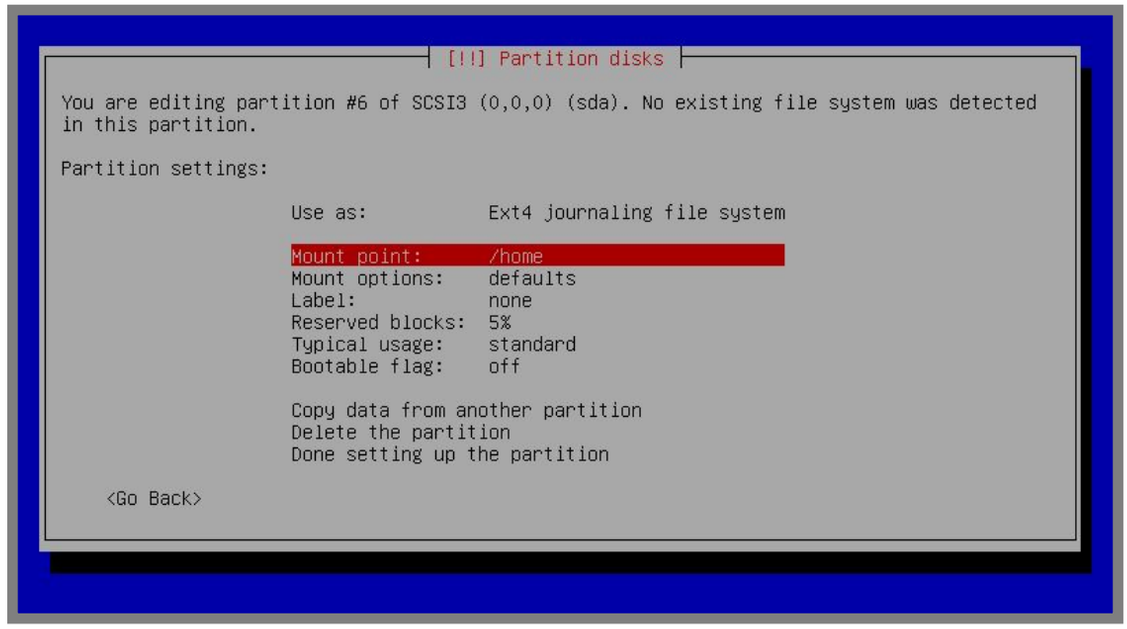


Gambar 2.33 Menentukan tipe yang akan digunakan oleh partisi /home

Gunakan *begginning*

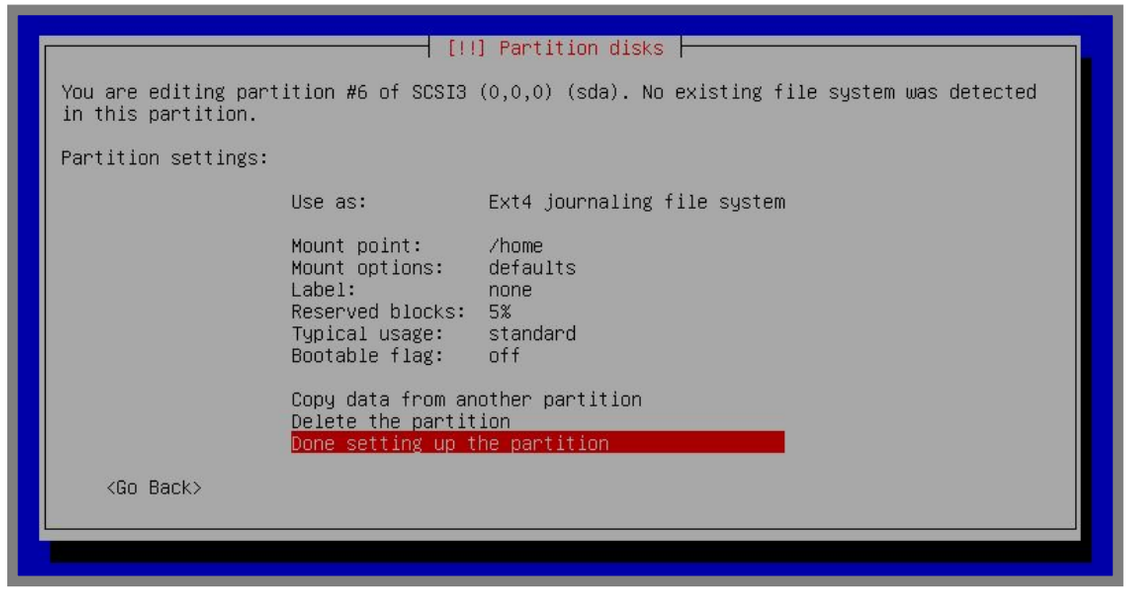
Gambar 2.34 Pemilihan lokasi partisi /home

Enter pada kolom *mount point* dan rubah menjadi */home*



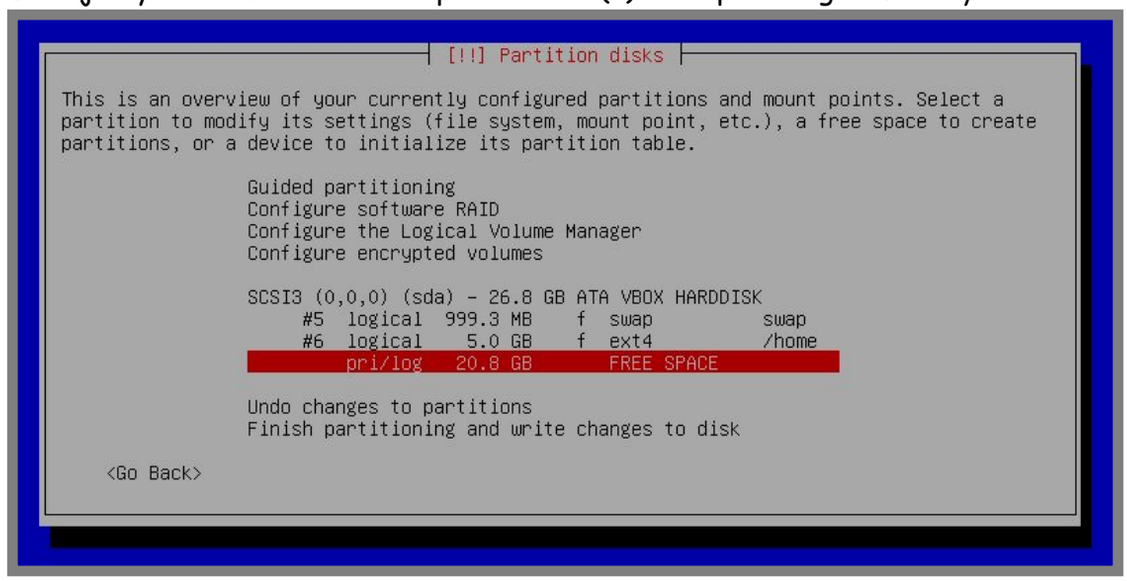
Gambar 2.35 Merubah partisi menjadi /home

Pilih *done setting up the partition*



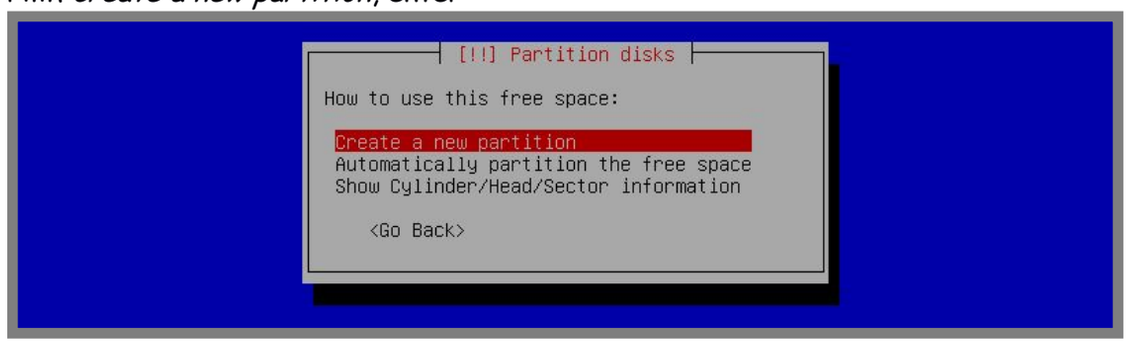
Gambar 2.35 Proses ahir pembuatan partisi /home

Selanjutnya kita akan membuat partisi root (/). Pilih pada bagian *free space*



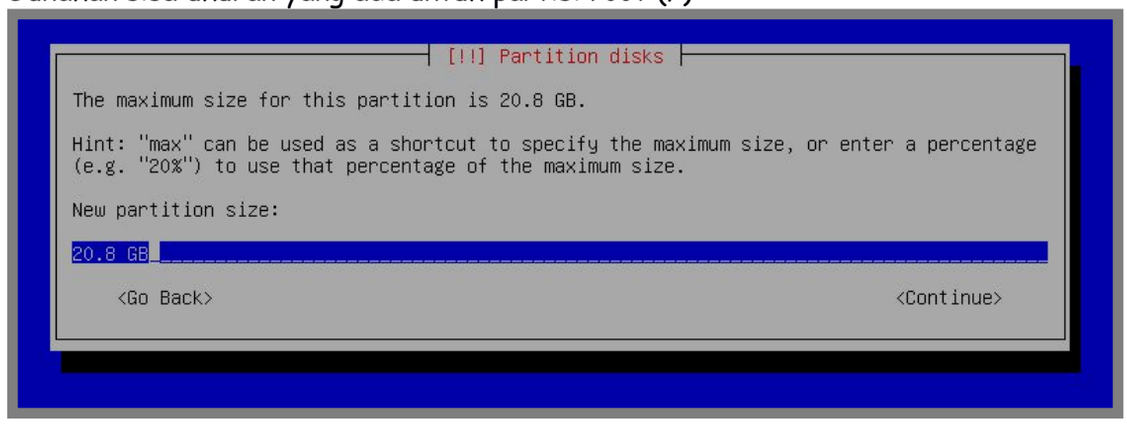
Gambar 2.36 Proses awal pembuatan partisi / (root)

Pilih *create a new partition*, enter



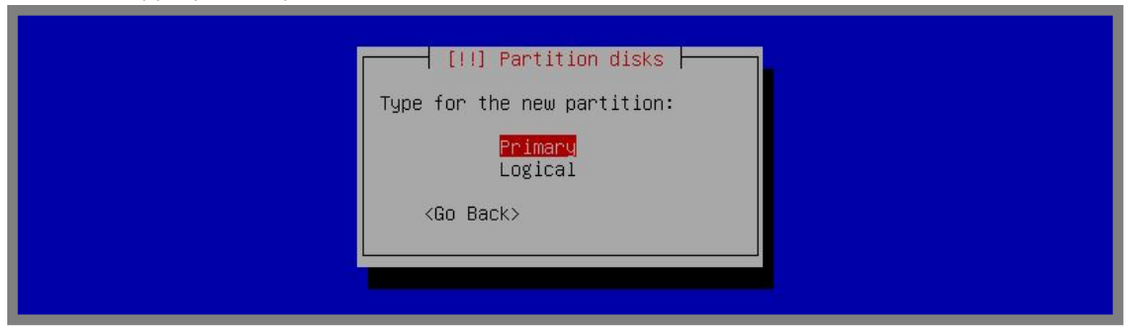
Gambar 2.37 Proses pembuatan partisi /

Gunakan sisa ukuran yang ada untuk partisi root (/)



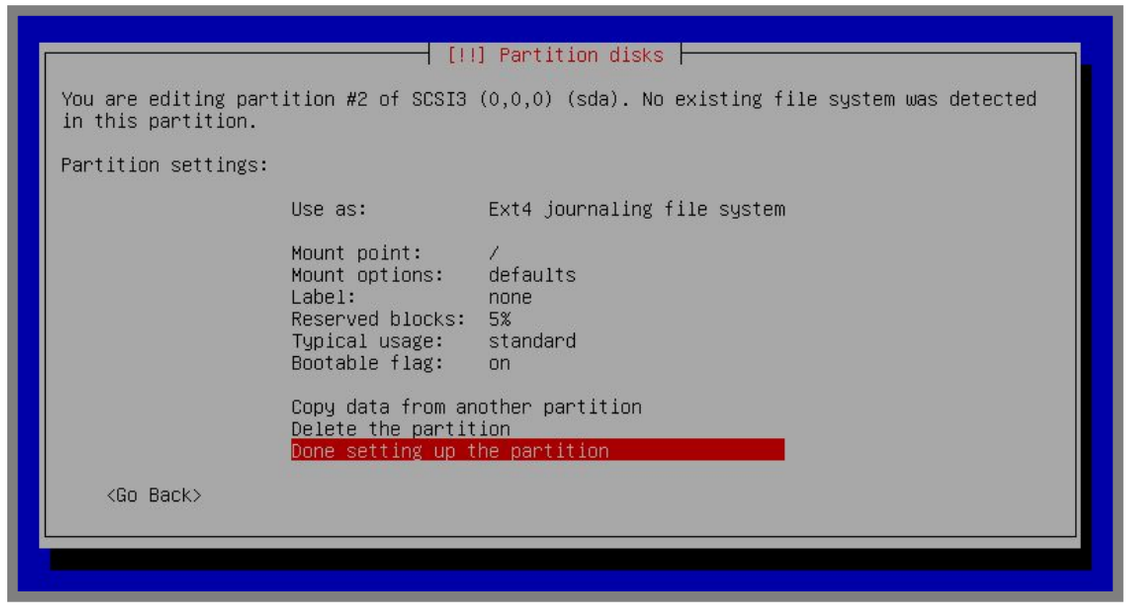
Gambar 2.38 Penentuan ukuran partisi /

Gunakan type *primary*



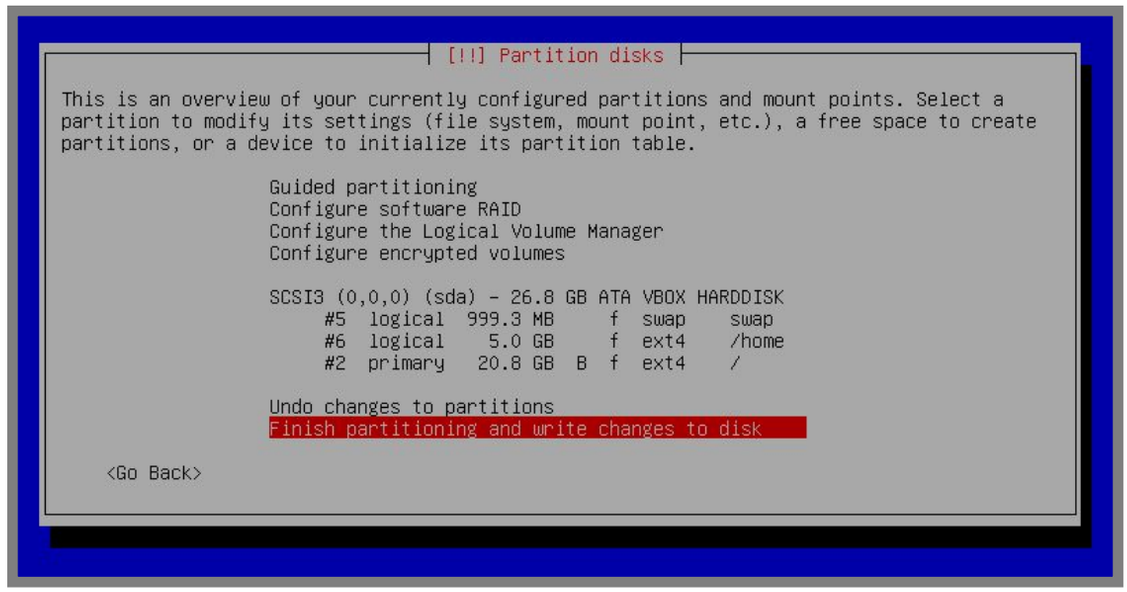
Gambar 2.39 Menentukan tipe partisi /

Pastikan parameter pada *bootable flag* adalah *on*. Selanjutnya pilih *done set up the partition*



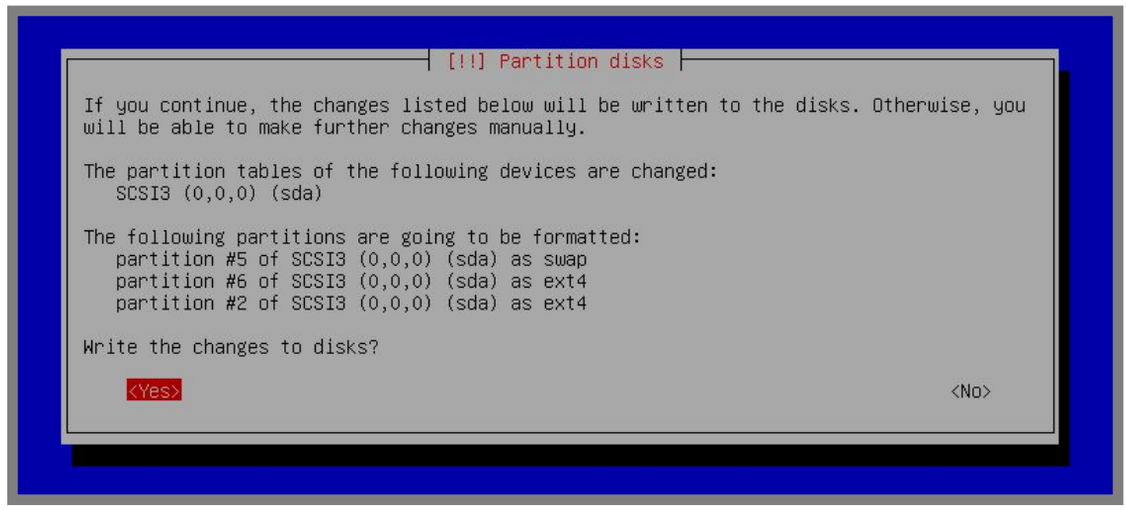
Gambar 2.40 Proses ahir pembuatan partisi /

Sampai saat ini kita sudah selesai membuat tiga partisi yang kita kehendakai. Pilih *finish partitioning and write changes to disk* untuk melanjutkan



Gambar 2.41 Finishing proses partisi

Akan ada halaman verifikasi, pilih *yes*

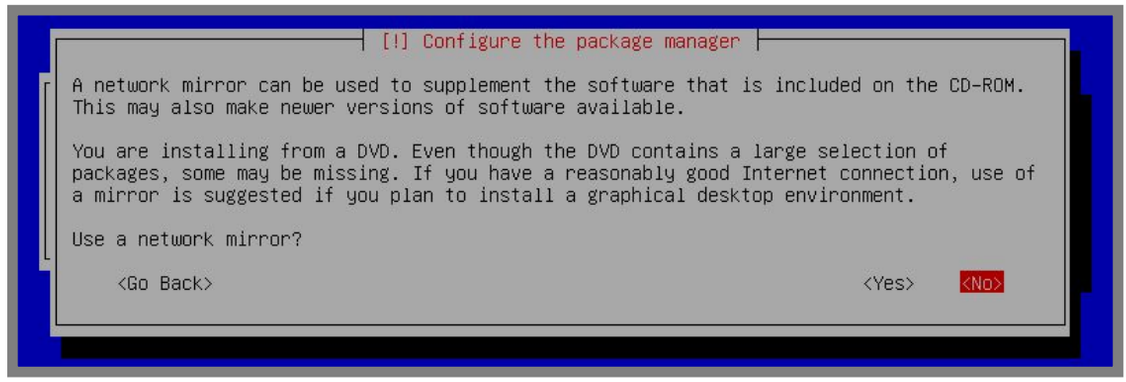


Gambar 2.42 Verifikasi hasil partisi

Pilih no jika ada pertanyaan scan another dvd

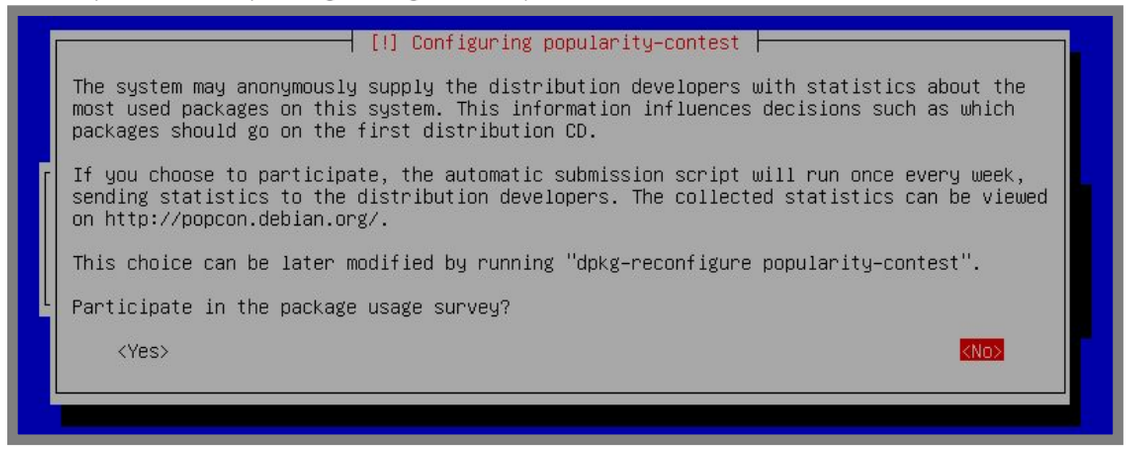
Gambar 2.43 Scan dvd untuk repository

Use a network mirror? *No*, enter



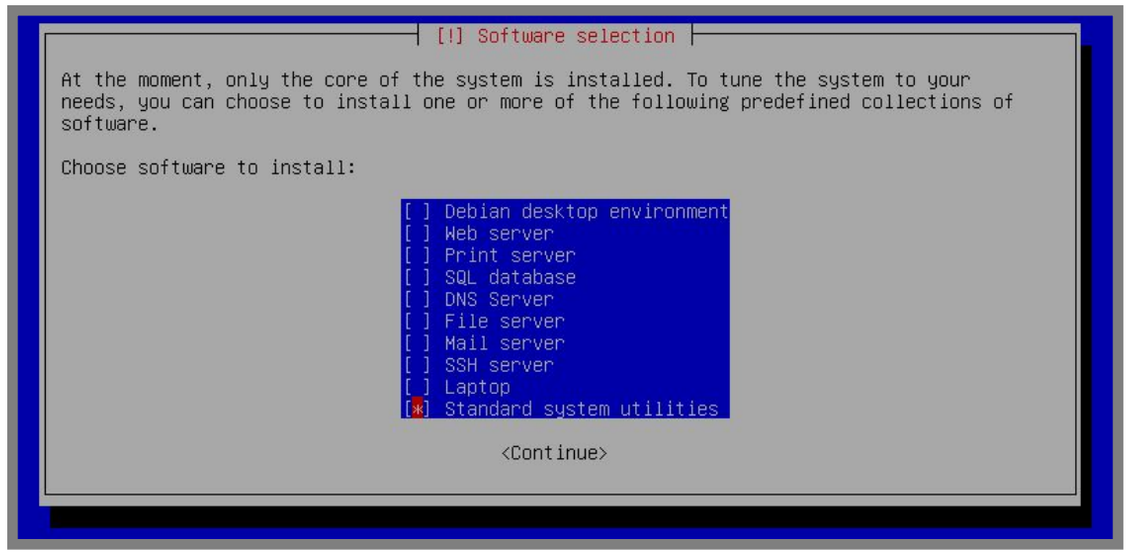
Gambar 2.44 Pertanyaan untuk repo lokal

Participate in the package usage survey? *No*,



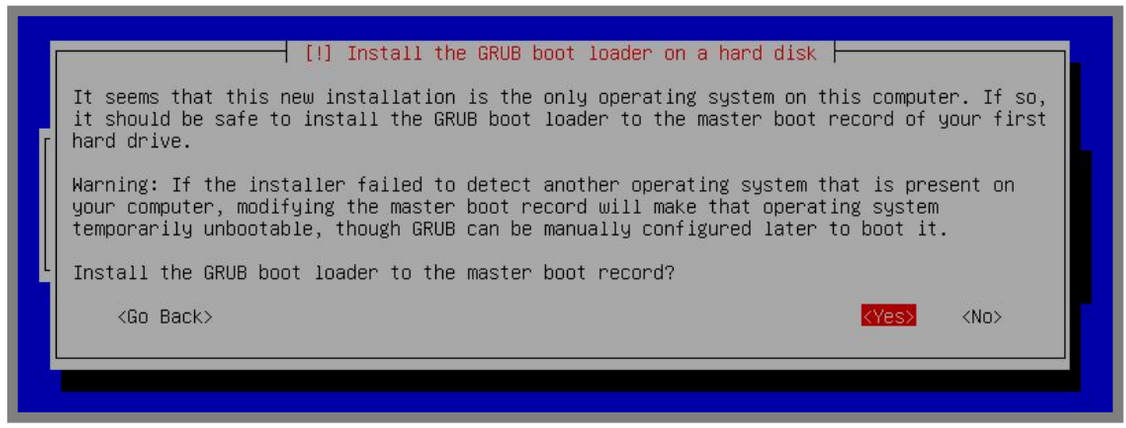
Gambar 2.45 Pertanyaan untuk suerver penggunaan

Langkah selanjutnya adalah menentukan paket apa saja yang ingin diinstall. Jika ingin menginstall server berbasis text, cukup centang pada *Standard system utilities*. Namun jika ingin install server berbasis GUI, centang *Debian desktop environment* dan *Standard system utilities*. Untuk menambahkan atau menghilangkan centang, gunakan tombol spasi pada kyboard. Kemudian enter,



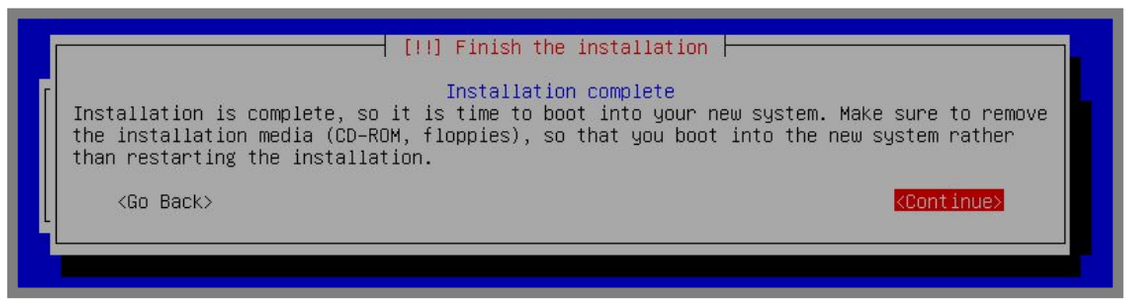
Gambar 2.46 Pemilihan aplikasi yang diinstall

Install grub boot loader? Yes



Gambar 2.47 Pertanyaan install grub boot loader

Terahir, akan ada perintah untuk restart, pilih *continue* dan enter



Gambar 2.48 Installasi telah selesai

Untuk pengujian, silahkan login menggunakan user yang telah dibuat saat proses installasi. Perhatikan bahwa password yang kita ketikkan di kyboard tidak akan ditampilkan di layar.

Gambar 2.49 Login ke server

---END OF CHAPTER---

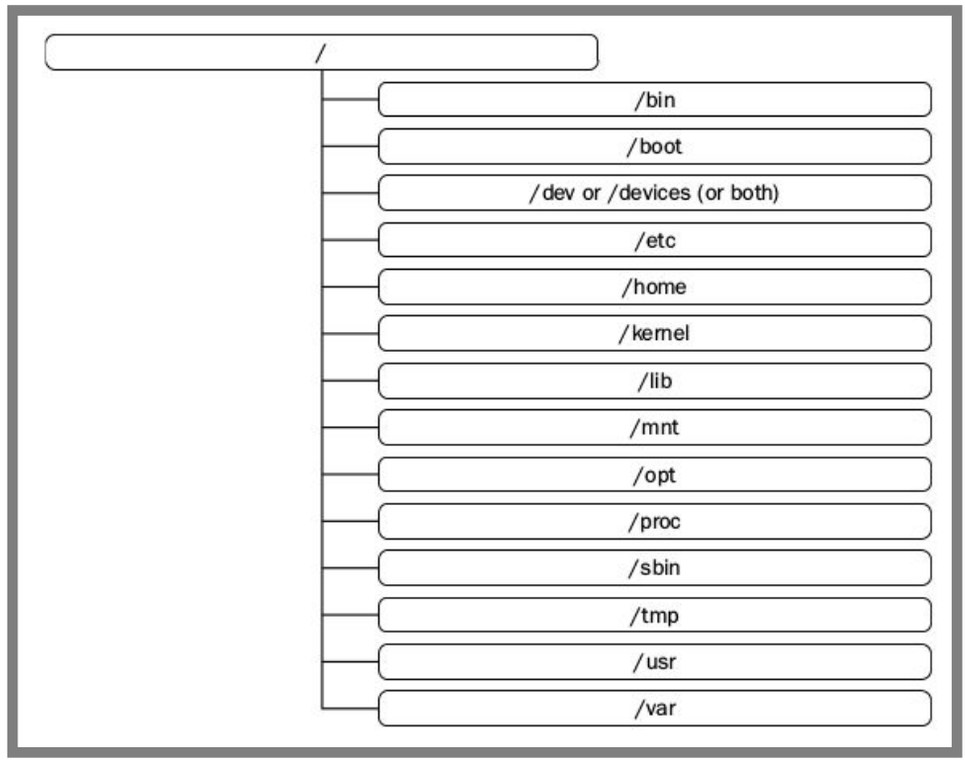
Bab 3
Pengetahuan Dasar Linux

Sebelum memulai melakukan konfigurasi server menggunakan linux debian, kita harus mengetahui dan paham mengenai dasar-dasar linux. Banyak sekali dasar-dasar linux yang harus dipahami untuk melakukan konfigurasi server, namun buku ini akan membahasnya secara singkat. Berikut dasar-dasar linux yang akan dibahas pada bab ini:

1. Struktur direktori/folder linux
2. Perintah dasar linux
3. Managemen user dan group di linux
4. Direktori & file permission di linux
5. Text editor di linux

Struktur Direktori/Folder Linux

Struktur direktori di linux menggunakan konsep hirarki. Dengan direktori root (/) sebagai direktori dasar bagi seluruh direktori yang ada di linux. Dengan kata lain, seluruh direktori yang ada di sistem operasi linux berada dibawah direktori root (/). Berikut gambaran umum struktur direktori di linux:



Gambar 3.1 Struktur direktori linux

Gambar diatas merupakan struktur direktori linux pada umumnya. Mungkin akan ada sedikit perbedaan antara beberapa distro linux yang beredar. Namun secara garis besar struktur direktori, dan fungsi masing-masing direktori tetaplah sama meskipun berbeda distro.

Berikut fungsi masing-masing direktori dilinux:

Direktori	Deskripsi
/	Direktori utama dari seluruh direktori yang berada di linux. Direktori ini berisi sub direktori yang mempunyai fungsi masing-masing dalam menjalankan proses sistem operasi.
/bin	Berisi file-file binari (executable) untuk digunakan oleh sistem. File binari adalah sebuah file aplikasi atau program dasar di linux.
/boot	Berisi file untuk keperluan booting sistem operasi
/dev	Berisi file-file device, seperti *cdrom* (CD-ROM Device), *sda1* (Harddisk Device), fd*0* (Removable Device), dll. Penamaan device bisa berbeda-beda antar distro linux.
/etc	Berisi file-file konfigurasi, seperti *passwd* (File untuk konfigurasi username password), *hosts* (File untuk konfigurasi hosts di jaringan), dll.
/home	Berisi file-file atau folder pribadi milik user, seperti Documents, Downloads, Music, Videos, Pictures, dll.
/kernel	Berisi file-file kernel. Kernel adalah file-file penting yang menjadi pondasi sebuah sistem operasi.
/lib	Berisi file-file library dan file-file yang berhubungan dengan kernel.
/mnt atau /media	Digunakan untuk memount file temporary, seperti *cdrom* (DVD/CD-ROM), *fd0* (FlashDisk), *sda1* (Harddisk), dll.
/proc	Berisi file-file yang mempunyai informasi tentang proses yang sedang dijalankan oleh sistem operasi. Direktori ini mempunyai isi yang dinamis (berubah-ubah).
/sbin	Sama dengan /bin, hanya saja yang disimpan adalah file-file binari yang berfungsi untuk administrasi sistem, seperti *ifconfig* (untuk managemen interface), dan *fdisk* (untuk managemen disk).
/tmp	Berisi file-file temporary yang digunakan selama sistem operasi berjalan.
/usr	Direktori ini bisa digunakan oleh seluruh user untuk berbagai kepentingan, seperti sharing folder.
/var	Berisi file-file variable, seperti file print, log, mail, dll.

Kita tidak perlu menghafal seluruh fungsi dari masing-masing direktori. Pemahaman yang paling penting sebenarnya adalah konsep sistem hirarki dari direktori root (/) itu saja. Jadi misal kita ingin mengedit file *passwd* didalam *etc*, maka kita harus masuk ke direktori *root (/)* dulu, kemudian *etc*, baru membuka file *passwd* (*/etc/passwd*).

Perintah Dasar Linux

Pemahaman mengenai perintah dasar di linux mutlak diperlukan untuk melakukan administrasi server jaringan berbasis linux. Karena banyak sekali hal-hal yang tidak bisa dilakukan pada mode GUI, dalam artian harus dilakukan menggunakan perintah text.

Banyak sekali perintah dasar dalam linux yang dapat kita gunakan untuk mempermudah pekerjaan kita. Namun pada bab ini akan dijelaskan beberapa perintah saja yang sering digunakan. Untuk lebih memahami masing-masing perintah, pada bab ini juga akan diberikan contoh penggunaan perintah.

Sebelum mengenal perintah-perintah dasar di linux, ada baiknya kita memahami tentang *bash* di linux. Bash adalah sebuah baris yang selalu mengawali perintah di linux. Seperti contoh berikut:

```
forkits@debian:~$
```

Gambar 3.2 Tampilan *bash* di linux

Berikut penjelasan dari masing-masing syntak yang ada pada bash

Syntak	Deskrisi
forkits	Syntak pertama, pada contoh diatas ditunjukkan dengan tulisan forkits. Syntak ini menunjukkan user yang sedang digunakan. Pada contoh diatas menunjukkan bahwa user yang sedang aktif adalah forkits.
debian	Selanjutnya, syntak yang ditunjukkan tulisan debian adalah hostname dari komputer kita. Hostname adalah nama dari sebuah komputer pada jaringan.
~	Tanda (~) yang terletak setelah (:) menunjukkan lokasi direktori dimana user sedang berada. Tanda (~) menunjukkan bahwa user sedang berada di direktori home miliknya. Direktori home dari sebuah user biasanya berada di */home/nama_user*. Contoh diatas menunjukkan bahwa direktori aktif adalah */home/forkits*
$	Syntak terahir menunjukkan bahwa user kita adalah user biasa atau user root. Jika tandanya adalah ($) seperti contoh diatas, menandakan bahwa user yang sedang aktif adalah user biasa. Sedangkan jika tandanya adalah (#) menunjukkan bahwa user yang sedang aktif adalah user root. Perbedaan user biasa dan user root akan dibahas di sub bab berikutnya.

Tentunya jika berbeda komputer, maka bash yang dimiliki akan berbeda. Tergantung dari user, hostname, dan lokasi direktori aktif.

Selanjutnya kita akan belajar beberapa perintah dasar yang sering digunakan untuk melakukan administrasi server linux.

pwd

Perintah ini berfungsi untuk menunjukkan lokasi direktori dimana user sedang berada. Atau sering disebut direktori aktif.

```
forkits@debian:~$ pwd
/home/forkits
forkits@debian:~$
```

Gambar 3.3 Penggunaan perintah *pwd*

ls

Perintah ini digunakan untuk melihat isi dari sebuah direktori. Berikut contoh dari penggunaan dari perintah ini.

```
forkits@debian:~$ ls
forkits@debian:~$ ls /
bin    etc          lib          mnt    root   selinux  tmp   vmlinuz
boot   home         lost+found   opt    run    srv      usr
dev    initrd.img   media        proc   sbin   sys      var
forkits@debian:~$ ls /home/
forkits
forkits@debian:~$ ls -a /home/
.  ..  forkits
forkits@debian:~$ ls -l /home/
total 4
drwxr-xr-x 2 forkits forkits 4096 Apr  9 07:02 forkits
forkits@debian:~$
```

Gambar 3.4 Penggunaan perintah *ls*

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
ls	Perintah tersebut artinya adalah kita melihat isi dari direktori aktif. Yaitu direktori home dari forkits (~), atau lebih tepatnya di */home/forkits*
ls /	Perintah tersebut artinya kita melihat isi dari direktori root (/)
ls /home/	Perintah tersebut artinya kita melihat isi dari direktori home yang ada dibawah direktori root (/)
ls -a /home/	Perintah ini sama dengan perintah sebelumnya, hanya saja perintah ini akan menunjukkan seluruh isi dari direktori, termasuk file atau direktori hidden. File/direktori hidden di linux ditandai dengan namanya yang didahului titik (.)
ls -l /home/	Perbedaan dari perintah sebelumnya adalah bahwa perintah ini akan menunjukkan isi dari direktori beserta detail keterangannya.

cd

Perintah ini digunakan untuk berpindah dari direktori aktif ke direktori lainnya. Berikut contoh penggunaan perintah ini.

```
forkits@debian:~$ pwd
/home/forkits
forkits@debian:~$ cd /etc/network/
forkits@debian:/etc/network$ cd run/
forkits@debian:/etc/network/run$ cd ..
forkits@debian:/etc/network$ cd
forkits@debian:~$ pwd
/home/forkits
forkits@debian:~$
```

Gambar 3.5 Contoh penggunaan perintah *cd*

Berikut penjelasan masing-masing perintah diatas

Perintah	Deskripsi
pwd	Perintah ini digunakan untuk melihat direktori aktif
cd /etc/network	Digunakan untuk berpindah ke direktori network yang ada didalam direktori etc, dan direktori etc berada didalam direktori root (/etc/network). Ingat konsep struktur direktori di linux, bahwa seluruh direktori berada dibawah direktori root (/). Hasil dari perintah ini ditunjukkan pada perubahan bash dibaris berikutnya.
cd run/	Perintah ini digunakan untuk berpindah ke direktori run yang ada didalam direktori network yang ada didalam direktori etc yang ada didalam direktori root (/etc/network/run). Karena sebelumnya kita telah berada di direktori /etc/network, maka kita tidak perlu mengetikkan struktur direktori lengkap (/etc/network/run) melainkan cukup dengan (run/). Hasil dari perintah ini ditunjukkan pada perubahan bash dibaris berikutnya.
cd ..	Perintah ini digunakan untuk keluar satu direktori dari direktori aktif. Perhatikan contoh diatas, direktori aktif sebelumnya adalah /etc/network/run. Setelah mengetikkan perintah ini, direktori aktif berpindah ke /etc/network (ditunjukkan oleh perubahan bash)
cd	Perintah ini digunakan untuk kembali ke direktori home. Perhatikan perintah pwd terahir, terlihat bahwa direktori aktif adalah /home/forkits

mkdir

Perintah ini digunakan untuk membuat sebuah direktori. Berikut contoh penggunaan dari perintah ini.

```
forkits@debian:~$ su
Password:
root@debian:/home/forkits# mkdir /home/linux
root@debian:/home/forkits# ls /home/
forkits  linux
root@debian:/home/forkits# cd /home/
root@debian:/home# mkdir debian
root@debian:/home# ls
debian  forkits  linux
root@debian:/home#
```

Gambar 3.6 Contoh penggunaan perintah *mkdir*

Berikut penjelsan masing-masing perintah diatas

Perintah	Deskripsi
su	Perintah ini digunakan untuk berpindah dari user biasa menjadi user root. Hal ini dikarenakan perintah mkdir tidak bisa sembarangan digunakan oleh user biasa. Perhatikan perbedaan tanda pada bash setelah menjadi user root (#).
mkdir /home/linux	Perintah ini digunakan untuk membuat sebuah direktori dengan nama linux didalam direktori home yang ada didirektori root (/).
ls /home/	Perintah ini digunakan untuk melihat isi direktori /home
cd /home/	Perintah ini digunakan untuk berpindah ke direktori /home
mkdir debian	Perintah ini memiliki fungsi yang sama dengan perintah sebelumnya, yaitu membuat sebuah direktori didalam direktori home dengan nama debian. Perbedaannya hanya penulisan perintahnya saja, dimana kita tidak perlu menulis secara spesifik (mkdir /home/debian), kita cukup menulis (mkdir debian) karena kita telah berada di direktori /home
ls	Perintah ini juga memiliki fungsi yang sama dengan perintah sebelumnya, yaitu melihat isi direktori /home. Sekali lagi kita tidak perlu menulis perintah spesifik (ls /home), melainkan cukup (ls) karena kita telah berada di direktori /home

rmdir

Perintah ini digunakan untuk menghapus sebuah direktori. Berikut contoh penggunaan perintah ini

```
root@debian:/etc# ls /home/
debian  forkits  linux
root@debian:/etc# rmdir /home/debian
root@debian:/etc# ls /home/
forkits  linux
root@debian:/etc# cd /home/
root@debian:/home# rmdir linux
root@debian:/home# ls
forkits
root@debian:/home#
```

Gambar 3.7 Contoh penggunaan perintah *rmdir*

Berikut penjelasan masing-masing perintah diatas

Perintah	Deskripsi
ls /home/	Perintah ini digunakan untuk melihat isi direktori /home/. Kita harus menulisnya spesifik, karena saat ini kita sedang berada di direktori /etc/. Jika kita hanya menulis (ls) saja, maka yang akan dilihat adalah isi direktori /etc, bukan isi direktori /home
rmdir /home/debian	Perintah ini digunakan untuk menghapus direktori debian yang berada di direktori /home. Sekali lagi perintah ini harus ditulis spesifik, karena saat ini kita masih berada di direktoir /etc
ls /home/	Perintah ini digunakan untuk melihat isi direktori /home
cd /home/	Perintah ini digunakan untuk berpindah ke direktori /home
rmdir linux	Perintah ini memiliki fungsi yang sama dengan perintah sebelumnya, yaitu menghapus direktori linux yang ada didalam direktori home. Perbedaannya hanya penulisan perintahnya saja, dimana kita tidak perlu menulis secara spesifik (rmdir /home/linux), kita cukup menulis (rmdir linux) karena kita telah berada di direktori /home
ls	Perintah ini juga memiliki fungsi yang sama dengan perintah sebelumnya, yaitu melihat isi direktori /home.

Perintah ini hanya bisa digunakan untuk menghapus direktori yang kosong. Jika direktori yang akan dihapus tidak kosong, perintah ini tidak akan berguna. Untuk menghapus direktori yang ada isinya, bisa menggunakan perintah rm yang akan dibahas di halaman selanjutnya.

touch

Perintah ini digunakan untuk membuat sebuah file. Berikut contoh penggunaan perintah ini.

```
root@debian:/home# mkdir debian
root@debian:/home# ls
forkits    debian
root@debian:/home# touch debian/linux.txt
root@debian:/home# ls debian/
linux.txt
root@debian:/home# touch /media/debian.txt
root@debian:/home# ls /media/
cdrom  cdrom0  debian.txt
root@debian:/home#
```

Gambar 3.8 Contoh penggunaan perintah *touch*

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
mkdir debian	Perintah ini digunakan untuk membuat direktori debian di dalam direktori /home
ls	Perintah ini digunakan untuk melihat isi direktori /home, hal ini karena saat ini kita berada di direktori /home
touch debian/linux.txt	Perintah ini digunakan untuk membuat file linux.txt didalam direktori /home/debian/linux.txt. Namun kita tidak perlu menulis perintah spesifik (touch /home/debian/linux.txt), kita cukup menulis (touch debian/linux.txt) karena saat ini kita telah berada di direktori /home
ls debian/	Perintah ini digunakan untuk melihat isi direktori /home/debian. Namun kita cukup menulis (ls debian/) karena saat ini kita telah berada didirektori /home
touch /media/debian.txt	Perintah ini digunakan untuk membuat sebuah file debian.txt didalam direktori /media. Berbeda dengan perintah sebelumnya, disini kita harus menulis spesifik, karena saat ini kita sedang berada didirektori /home
ls /media/	Perintah ini digunakan untuk melihat isi direktori /media

rm

Perintah ini digunakan untuk menghapus sebuah file ataupun direktori. Berikut contoh penggunaan perintah ini

```
root@debian:/home# ls /media/
cdrom  cdrom0  debian.txt
root@debian:/home# rm /media/debian.txt
root@debian:/home# ls debian/
linux.txt
root@debian:/home# rm debian/ -rf
root@debian:/home# ls
forkits
root@debian:/home#
```

Gambar 3.9 Contoh penggunaan perintah *rm*

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
ls /media/	Perintah ini digunakan untuk melihat isi direktori /media. Perhatikan bahwa ada sebuah file dengan nama debian.txt yang telah kita buat dipembahasan sebelumnya
rm /media/debian.txt	Perintah ini digunakan untuk menghapus file debian.txt yang ada di direktori /media/
ls debian/	Perhatikan bahwa didalam direktori /home/debian terdapat sebuah file dengan nama linux.txt
rm debian/ -rf	Perintah rm dengan option (-rf) dibelakang seperti diatas, digunakan untuk menghapus sebuah direktori yang tidak kosong. Perhatikan bahwa direktori debian masih berisi file linux.txt
ls	Digunakan untuk melihat isi direktori /home, perhatikan bahwa direktori debian sudah hilang setelah dihapus dengan perintah sebelumnya.

cp

Perintah ini digunakan untuk melakukan copy file ataupun folder. Berikut contoh penggunaan dari perintah ini.

```
root@debian:/home# mkdir unix linux
root@debian:/home# ls
forkits  linux  unix
root@debian:/home# touch unix/bsd.txt
root@debian:/home# cp linux/ /media/ -rf
root@debian:/home# cp unix/bsd.txt /media/
root@debian:/home# ls /media/
bsd.txt  cdrom cdrom0 linux
root@debian:/home# ls
forkits  linux  unix
root@debian:/home#
```

Gambar 3.10 Contoh penggunaan perintah *cp*

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
mkdir unix linux	Perintah ini digunakan untuk membuat dua direktori didalam direktori /home, yaitu direktori unix dan linux.
ls	Perintah ini digunakan untuk melihat isi direktori /home. Perhatikan bahwa telah ada dua direktori yang baru saja dibuat dengan perintah sebelumnya.
touch unix/bsd.txt	Perintah ini digunakan untuk membuat sebuah file dengan nama bsd.txt didalam direktori /home/unix
cp linux/ /media/ -rf	Perintah ini digunakan untuk copy direktori linux yang ada di /home ke direktori /media. Ingata bahwa jika ingin copy direktori, harus menyertakan option (-rf)
cp unix/bsd.txt /media/	Perintah ini digunakan untuk copy file bsd.txt yang ada didirektori /home/unix ke direktori /media. Perhatikan bahwa kita tidak perlu menyertakan option (-rf) jika hanya ingin copy sebuah file.
ls /media/	Perintah ini digunakan untuk melihat isi direktori /media. Perhatikan bahwa direktori linux dan file bsd.txt yang telah kita copy dengan perintah sebelumnya sudah ada.
ls	Perintah ini digunakan untuk melihat isi direktori /home. Perhatikan bahwa dua direktori yang kita buat masih ada.

mv

Perintah ini digunakan untuk memindahkan sebuah file ataupun direktori. Berikut contoh penggunaan dari perintah ini

```
root@debian:/home# ls
forkits  linux  unix
root@debian:/home# ls unix/
bsd.txt
root@debian:/home# mv unix/bsd.txt /mnt/
root@debian:/home# ls unix/
root@debian:/home# mv linux/ /mnt/
root@debian:/home# ls
forkits  unix
root@debian:/home# ls /mnt/
bsd.txt  linux
root@debian:/home#
```

Gambar 3.11 Contoh penggunaan perintah *mv*

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
ls	Perintah ini digunakan untuk melihat isi direktori /home. Perhatikan bahwa telah ada dua direktori yang telah dibuat saat pembahsan perintah cp
ls unix/	Perintah ini digunakan untuk melihat isi direktori /home/unix. Perhatikan bahwa telah ada sebuah file dengan nama bsd.txt yang telah dibuat sebelumnya.
mv unix/bsd.txt /mnt/	Perintah ini digunakan untuk memindahkan file bsd.txt yang ada di direktori /home/unix ke direktori /mnt
ls unix/	Perintah ini digunakan untuk melihat isi direktori /home/unix. Perhatikan bahwa file bsd.txt telah tidak ada. Hal ini karena file tersebut sudah dipindahkan ke direktori /mnt
mv linux/ /ment/	Perintah ini digunakan untuk memindahkan direktori linux yang ada di /home ke direktori /mnt
ls	Perintah ini digunakan untuk melihat isi direktori /home. Perhatikan bahwa direktori linux sudah tidak ada. Hal ini karena direktori linux sudah dipindahkan ke direktori /mnt dengan perintah sebelumnya
ls /mnt/	Perintah ini digunakan untuk melihat isi direktori /mnt. Perhatikan bahwa disana telah ada direktori linux dan file bsd.txt.

cat

Perintah ini digunakan untuk melihat isi dari suatu file. Berikut contoh penggunaan perintah cat untuk melihat isi file /etc/network/interfaces

```
root@debian:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet dhcp
root@debian:~#
```

Gambar 3.12 Contoh penggunaan perintah *cat*

grep

Perintah ini digunakan untuk mencari karakter, kata, atau kalimat tertentu dengan suatu kata kunci. Berikut contoh penggunaan perintah ini

```
root@debian:~# cat /etc/network/interfaces | grep dhcp
iface eth0 inet dhcp
root@debian:~# ls /etc/network/
if-down.d  if-post-down.d  if-pre-up.d  if-up.d  interfaces  run
root@debian:~# ls /etc/network/ | grep interfaces
interfaces
root@debian:~# ls /etc/network/ | grep linux
root@debian:~#
```

Gambar 3.13 Contoh penggunaan perintah *grep*

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
cat /etc/network/interfaces | grep dhcp	Perintah ini digunakan untuk melihat isi file /etc/network/interfaces. Perhatikan bahwa keluarannya hanya satu baris yang terdapat kata dhcp, hal ini berkat perintah grep. Perhatikan bahwa karakter (|) memungkinkan untuk menjalankan lebih dari satu perintah dalam waktu yang sama.
ls /etc/network/	Perintah ini digunakan untuk isi direktori /etc/network.
ls /etc/network/ | grep interfaces	Perhatikan bahwa dengan tambahan perintah grep, maka hanya file yang dengan nama interfaces saja yang ditampilkan

ls /etc/network/ | grep linux	Perhatikan bahwa perintah ini tidak menghasilkan keluaran apapun. Hal ini dikarenakan tidak ada file/direktori dengan kata kunci linux didalam direktori /etc/network/

man

Perintah ini digunakan untuk menampilkan manual dari suatu perintah. Jadi misalkan suatu saat kita lupa apa itu fungsi dari perintah */s*, kita bisa melihat manualnya dengan perintah *man ls*. Berikut contoh keluaran dari perintah *man ls*

```
LS(1)                                                    User Commands
LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by
default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort is
speci-
       fied.

       Mandatory arguments to long options are mandatory for short
options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
 Manual page ls(1) line 1 (press h for help or q to quit)
```

Gambar 3.14 Contoh penggunaan perintah *man*

Untuk membaca halaman selanjutnya dari manual tersebut, gunakan tombol spasi pada kyboard, sedangkan jika ingin keluar dari manual, tekan tombol q pada kyboar.

Managemen User dan Group di Linux

Sistem operasi linux merupakan sistem operasi multiuser, artinya bisa menangani lebih dari satu user dalam waktu yang bersamaan. Dengan kata lain, satu komputer dengan sistem operasi linux bisa digunakan dua orang atau lebih dengan user yang berbeda-beda dalam waktu yang sama.

Secara garis besar, user dibedakan menjadi dua, yaitu user root dan user biasa. Dimana user root mempunyai simbol (#) dan user biasa mempunyai simbol ($) pada bash. Perbedaan antara keduanya adalah dalam hal hak akses. User root memiliki hak akses maksimal didalam sistem operasi, artinya user root bisa melakukan apa saja dalam sistem operasi. Sedangkan user biasa normalnya hanya bisa melakukan modifikasi file atau direktori yang berada didalam direktori home miliknya saja (*/home/nama_user*).

Fungsi utama sebuah group adalah memudahkan dalam memberikan hak akses kepada sejumlah user (Hak akses akan dibahas di sub bab berikutnya). Misal dalam sebuah sistem operasi mempunyai user dengan nama tkj1, tkj2, tkj3, dan tkj4. Administrator (pemegang user root) menginginkan agar keempat user tadi memiliki hak akses yang sama. Tentunya akan sangat merepotkan jika harus memberikan hak akses kepada masing-masing user. Oleh karena itu, kita bisa langsung memasukkan keempat user tersebut kedalam sebuah group dengan nama tkj, sehingga kita cukup mengatur hak akses pada group tkj saja dan keempat user tersebut sudah memiliki hak akses sesuai dengan yang diberikan kepada group tkj.

Berikut beberapa perintah dasar yang bisa digunakan untuk melakukan managemen user dan group

useradd

```
root@debian:~# useradd tkj1
root@debian:~# passwd tkj1
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@debian:~#
```

Gambar 3.15 Contoh penggunaan perintah *useradd* dan *passwd*

Perintah diatas digunakan untuk membuat sebuah user dengan nama tkj1. Selanjutnya perintah *passwd* digunakan untuk memberikan password pada user tkj1. Perhatikan bahwa penulisan password tidak ditampilkan di layar.

adduser

```
root@debian:~# adduser tkj2
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
Adding user `tkj2' ...
Adding new group `tkj2' (1002) ...
Adding new user `tkj2' (1002) with group `tkj2' ...
Creating home directory `/home/tkj2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for tkj2
Enter the new value, or press ENTER for the default
    Full Name []: TKJ 2
    Room Number []: R2
    Work Phone []: 534
    Home Phone []: 231
    Other []:
Is the information correct? [Y/n] Y
root@debian:~# ls /home/
forkits  tkj2
root@debian:~#
```

Gambar 3.16 Contoh penggunaan perintah *adduser*

Perintah ini juga digunakan untuk membuat sebuah user. Hanya saja perintah ini akan meminta detail dari user yang akan dibuat (termasuk password), berbeda dengan perintah sebelumnya (useradd) yang sama sekali tidak meminta detail dari user yang dibuat. Perbedaan lainnya adalah bahwa jika menggunakan perintah ini, maka user yang dibuat otomatis akan dibuatkan sebuah home direktori di /home/nama_user. Perhatikan perintah *ls /home/* menunjukkan bahwa terdapat direktori dengan nama tkj2.

su

```
root@debian:~# su tkj2
tkj2@debian:/root$ su
Password:
root@debian:~#
```

Gambar 3.17 Contoh penggunaan perintah *su*

Perintah ini digunakan untuk berpindah ke user lain. Format penulisan dari perintah ini adalah *su nama_user*. Perhatikan perintah pertama (su tkj2) menunjukkan perintah untuk berpindah ke user tkj2. Perhatikan bash pada baris kedua (tkj2@debian:/root$), terlihat bahwa usernya sudah berubah menjadi tkj2. Begitu juga tandanya juga berubah dari (#) menjadi ($) yang menandakan bahwa tkj2 adalah user biasa. Selanjutnya jika kita hanya menggunakan perintah *su* saja, itu tandanya kita ingin berpindah ke user root (lihat perintah kedua).

userdel

Perintah ini digunakan untuk menghapus user. Berikut contoh penggunaan perintah ini.

```
root@debian:~# userdel tkj2
userdel: user tkj2 is currently used by process 2569
root@debian:~# su tkj2
tkj2@debian:/root$ su
Password:
root@debian:~# userdel -rf tkj2
userdel: user tkj2 is currently used by process 2569
userdel: tkj2 mail spool (/var/mail/tkj2) not found
root@debian:~# su tkj2
No passwd entry for user 'tkj2'
root@debian:~# passwd tkj2
passwd: user 'tkj2' does not exist
root@debian:~# ls /home/
forkits
root@debian:~#
```

Gambar 3.18 Contoh penggunaan *userdel*

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
userdel tkj2	Perintah ini digunakan untuk menghapus user tkj2. Seharusnya perintah ini bisa berjalan jika user tkj2 tidak sedang login, namun karena saat ini tkj2 sedang login (login saat pembahasan perintah su), maka perintah ini tida berjalan.
su tkj2	Perintah ini membuktikan bahwa user tkj2 belum terhapus oleh perintah sebelumnya (userdel tkj2). Hal ini karena kita masih berhasil login sebagai user tkj2.
su	Perintah ini digunakan untuk berpindah ke user root. Hal ini karena hanya user root saja yang bisa melakukan perintah userdel

userdel -rf tkj2	Dengan menggunakan option (-rf), akan menghapus user tkj2 meskipun masih dalam keadaan login. Option (-f) artinya force yang menghapusnya dengan paksa, sedangkan option (-r) menghapus home direktori dari user tkj2.
su tkj2	Perhatikan bahwa kita tidak bisa login sebagai user tkj2, namun peringatannya adalah tidak ada password untu user tkj2.
passwd tkj2	Karena perintah sebelumnya mengeluarkan peringatann bahwa tidak ada password untuk tkj2, kita akan coba memberikan password untuk tkj2. Namun ternyata keluar sebuah peringatan bahwa user tkj2 tidak ada. Ini menandakan bahwa perintah untuk menghapus user tkj2 diatas telah berhasil
ls /home/	Perintah ini digunakan untuk melihat isi direktori /home. Perhatikan bahwa home direktori dari tkj2 sudah tidak ada.

groupadd

Perintah ini digunakan untuk membuat sebuah group. Berikut contoh penggunaan perintah ini saat mencoba membuat sebuah group dengan nama tkj.

```
root@debian:~# groupadd tkj
root@debian:~#
```

Gambar 3.19 Contoh penggunaan perintah *groupadd*

groupdel

Perintah ini digunakan untuk menghapus sebuah group. Berikut contoh penggunaan perintah ini saat mencoba menghapus sebuah group dengan nama tkj yang telah dibuat sebelumnya

```
root@debian:~# groupdel tkj
root@debian:~#
```

Gambar 3.20 Contoh penggunaan perintah *groupdel*

groups

Perintah ini digunakan untuk melihat keanggotaan suatu user terhadap group.

```
root@debian:~# groups tkj1
tkj1 : tkj1
root@debian:~#
```

Gambar 3.21 Contoh penggunaan perintah *groups*

Sebelumnya kita telah membuat user dengan nama tkj1. Perhatikan bahwa tkj1 merupakan anggota dari group tkj1. Hal ini karena saat kita membuat sebuah user, maka otomatis akan dibuatkan sebuah group dengan nama yang sama dengan user yang dibuat, dan otomatis user yang dibuat akan menjadi anggota dari group tersebut.

adduser & groups

Seperti contoh kasus yang dibahas di awal sub bab, misal kita mempunyai user tkj1, tkj2, tkj3, dan tkj4. Kita ingin memasukkann keempat user tersebut ke group dengan nama tkj. Diasumsikan kita telah membuat empat user dengan nama tkj1, tkj2, tkj3, dan tkj4 serta sebuah group dengan nama tkj.

```
root@debian:~# adduser tkj1 tkj
Adding user `tkj1' to group `tkj' ...
Adding user tkj1 to group tkj
Done.
root@debian:~# groups tkj1
tkj1 : tkj1 tkj
root@debian:~#
```

Gambar 3.22 Memasukkan user kedalam group

Perintah pertama (adduser tkj1 tkj) artinya memasukkan user tkj1 ke group tkj. Dibuktikan dengan perintah kedua (groups tkj1) bahwa saat ini user tkj1 juga merupakan anggota dari group tkj. Jika ingin memasukkan user tkj2, tkj3, dan tkj4 ke group tkj, perintah yang digunakan sama.

Direktori & File Permission di Linux

Pada sub bab ini, akan dibahas mengenai hak akses suatu user atau group terhadap file dan direktori. Dari segi tingkat hak akses, hak akses itu sendiri dibedakan menjadi tiga, yaitu read (r), write (w), dan executable (x). Sedangkan dari segi pemilik hak akses, hak akses dibedakan menjadi tiga juga, yaitu user/owner (u), groups (g), dan other (o).

Sebelum melangkah lebih jauh, kita akan belajar membaca hak akses suatu file atau direktori. Perhatikan perintah berikut

```
root@debian:/boot# ls -l
total 13676
-rw-r--r-- 1 root root  1577218 Apr 23  2014 System.map-3.2.0-4-486
-rw-r--r-- 1 root root   134707 Apr 23  2014 config-3.2.0-4-486
drwxr-xr-x 3 root root    12288 Apr  9  07:02 grub
-rw-r--r-- 1 root root 9759836 Apr  9  06:44 initrd.img-3.2.0-4-486
-rw-r--r-- 1 root root 2512736 Apr 23  2014 vmlinuz-3.2.0-4-486
root@debian:/boot#
```

Gambar 3.23 Analisa file atau direktori

Perintah diatas adalah perintah melihat isi direktori /boot dengan detail (ls -l). Berikut penjelasan masing-masing detail yang diberikan

Simbol	Deskripsi
-	Simbol ini menunjukkan type dari file atau direktori. Jika simbolnya adalah (-), berarti adalah file. Sedangkan jika simbolnya adalah (d), berarti direktori (perhatikan file ke 3)
rw-	Simbol ini adalah hak akses untuk user atau owner (pemilik). Simbol tersebut menunjukkan bahwa user/owner memiliki hak akses read dan write. Jika saja simbolnya (rwx), maka user memiliki hak akses penuh, yaitu read, write, dan executable.
r--	Simbol ini menunjukkan hak akses group. Artinya group yang memiliki file ini hanya memiliki hak akses read.
r--	Simbol ini menunjukkan hak akses untuk other (bukan user/owner, juga bukan anggota group). Artinya other memiliki hak akses read saja pada file ini.
root	Simbol ini menunjukkan user/owner (pemilik) dari file ini. Artinya file ini adalah file milik user root, jadi user root memiliki hak akses (rw-)
root	Simbol ini menunjukkan group pemilik file ini. Artinya file ini adalah milik group root, jadi user apa saja yang merupakan anggota group root akan memiliki hak akses (r--)
Apr 23 2014	Simbol ini menunjukkan waktu terahir kali file/direktori dibuat atau dimodifikasi.
System.map-3.2.0-4-486	Simbol ini menunjukkan nama dari file/direktori.

chown

Perintah ini digunakan untuk merubah kepimilikan sutau file/direktori terhadap user dan group. Berikut contoh penggunaan perintah ini

```
root@debian:/# cd /mnt/
root@debian:/mnt# ls -l
total 4
-rw-r--r-- 1 root root    0 Apr 10 05:46 bsd.txt
drwxr-xr-x 2 root root 4096 Apr 10 05:46 linux
root@debian:/mnt# chown forkits:tkj bsd.txt
root@debian:/mnt# ls -l
total 4
-rw-r--r-- 1 forkits tkj    0 Apr 10 05:46 bsd.txt
drwxr-xr-x 2 root    root 4096 Apr 10 05:46 linux
root@debian:/mnt#
```

Gambar 3.24 Contoh penggunaan perintah *chown*

Berikut penjelasan masing-masing perintah diatas

Perintah	Deskripsi
cd /mnt/	Perintah ini digunakan untuk masuk ke direktori /mnt/
ls -l	Perintah ini digunakan untuk melihat isi direktori /mnt/ dengan detail. Perhatikan bahwa ada sebuah file dengan nama bsd.txt dan direktori linux. File bsd.txt adalah milik user root dan group root
chown forkits:tkj bsd.txt	Perintah ini digunakan untuk merubah kepemilikan file bsd.txt menjadi milik user forkits dan group tkj. (user forkits dan group tkj telah dibuat sebelumnya).
ls -l	Perintah ini digunakan untuk melihat isi direktori /mnt dengan detail. Perhatikan bahwa saat ini file bsd.txt sudah milik user forkits dan group tkj

chmod

Perintah ini digunakan untuk merubah hak akses suatu file/direktori. Perintah ini memanfaatkan konversi bilangan biner ke desimal yang merepresentasikan hak akses rwx (read, write, executable). Perhatikan tabel berikut.

Hak akses	rwx	rw-	r-x	r--	-wx	-w-	--x	---
Biner	111	110	101	100	011	010	001	000
Desimal	7	6	5	4	3	2	1	0

Tabel diatas menunjukkan hubungan antara hak akses, biner, dan desimal. Berikut tabel husus yang menunjukkan konversi angka biner ke desimal.

Biner	111	110	101	100	011	010	001	000
Rumus	$2^2+2^1+2^0$	$2^2+2^1+2^0$	$2^2+2^1+2^0$	$2^2+2^1+2^0$	$2^2+2^1+2^0$	$2^2+2^1+2^0$	$2^2+2^1+2^0$	$2^2+2^1+2^0$
Desimal	4+2+1	4+2	4+1	4	2+1	2	1	0

Pemberian hak akses menggunakan perintah chmod memiliki urutan **UGO**, yaitu user (u), group (g), dan other (o). Sehingga jika kita ingin memberikan hak akses read write dan executable untuk user, read dan executable untuk group, dan read untuk other, maka kita menggunakan angka 754. 7 untuk user (rwx), 5 untuk group (r-x), dan 4 untuk other (r--).
Berikut contoh penggunaan perintah ini.

```
root@debian:/mnt# ls -l
total 4
-rw-r--r-- 1  forkits  tkj      0 Apr 10 05:46 bsd.txt
drwxr-xr-x 2 root     root 4096 Apr 10 05:46 linux
root@debian:/mnt# chmod 754 bsd.txt
root@debian:/mnt# ls -l
total 4
-rwxr-xr-- 1  forkits tkj      0 Apr 10 05:46 bsd.txt
drwxr-xr-x 2 root     root 4096 Apr 10 05:46 linux
root@debian:/mnt# touch linux/index.html
root@debian:/mnt# ls -l linux/
total 0
-rw-r--r-- 1 root root 0 Apr 10 14:21 index.html
root@debian:/mnt# chmod 777 linux/ -R
root@debian:/mnt# ls -l
total 4
-rwxr-xr-- 1  forkits tkj      0 Apr 10 05:46 bsd.txt
drwxrwxrwx 2 root     root 4096 Apr 10 14:21 linux
root@debian:/mnt# ls -l linux/
total 0
-rwxrwxrwx 1 root root 0 Apr 10 14:21 index.html
root@debian:/mnt#
```

Gambar 3.25 Contoh penggunaan perintah *chmod*

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
ls -l	Perintah ini digunakan untuk melihat isi direktori /mnt. Perhatikan bahwa file bsd.txt memiliki hak akses rw-r--r-- (644). Yaitu rw- untuk user (6), r-- untuk group (4), dan r-- untuk other (4).

chmod 754 bsd.txt	Perintah ini digunakan untuk mengganti hak akses file bsd.txt menjadi 754, yaitu rwx untuk user (7), r-x untuk group (5), dan r-- untuk other (4). Perhatikan hasilnya pada perintah *ls -l* berikutnya.
ls -l	Perhatikan bahwa saat ini file bsd.txt telah memiliki hak akses 754 (rwxr-xr--)
touch linux/index.html	Membuat file dengan nama index.html didalam direktori /mnt/linux
ls -l linux/	Perintah ini digunakan untuk melihat isi direktori /mnt/linux dengan detail. Perhatikan bahwa file index.html yang baru saja kita buat memiliki hak akses rw-r--r-- (644).
chmod 777 linux/ -R	Perintah ini digunakan untuk merubah hak akses direktori /mnt/linux menjadi 777 (rwxrwxrwx). Option (-R) diahir perintah menunjukkan bahwa perintah ini juga diberlakukan kepada seluruh file/direktori yang berada didalam direktori linux. Dalam artian, file index.html yang berada didalam direktori /mnt/linux juga akan berubah hak aksesnya.
ls -l	Perhatikan bahwa direktori /mnt/linux sudah memiliki hak akses 777 (rwxrwxrwx)
ls -l linux/	Perhatikan bahwa file index.html yang berada di direktori /mnt/linux juga mempunyai hak akses 777 (rwxrwxrwx), padahal kita tidak memberikan perintah untuk merubah hak akses file ini. Hak akses file ini berubah akibat option (-R) pada perintah (chmod 777 linux/ -R).

Text Editor di Linux

Sesuai namanya, text editor di linux digunakan untuk melakukan modifikasi suatu file. Selain itu, text editor juga bisa digunakan untuk membuat sekaligus memodifikasi suatu file.

Ada beberapa text editor yang bisa kita gunakan di linux, diantaranya vi, vim, pico, nano, ex, dll. Namun pada sub bab ini, hanya akan dibahas text editor yang sering dipakai oleh pemula, yaitu nano/pico. Secara garis besar, text editor pico dan nano memiliki kemiripan, baik dari segi fungsi, tool yang tersedia, dan kecepatan.

Berikut contoh penggunaan text editor nano

```
root@debian:~# touch linux.txt
root@debian:~# ls
linux.txt
root@debian:~# nano linux.txt
```

Gambar 3.26 Contoh penggunaan perintah n*ano*

Terlihat bahwa ada tiga perintah pada gambar diatas. Perintah pertama (touch linux.txt) digunakan untuk membuat sebuah file dengan nama linux.txt. Terlihat bahwa perintah tersebut berhasi, terbukti dengan perintah (ls). Perintah terahir digunakan untuk mengedit isi file linux.txt dengan text editor nano. Setelah menjalankan perintah tersebut, maka akan muncul sebuah halaman editor, kita bebas menambahkan atau menghapus isi file seperti berikut

Gambar 3.27 Proses edit file dengan *nano*

Selanjutnya, untuk menyimpan hasil perubahan yang telah dilakukan, gunakan tombol kombinasi *ctrl+x*, selnjutnya *y*, dan terahir *enter*. Perhatikan gambar dibawah ini, terlihat bahwa saat ini file linux.txt telah berisi sebaris tulisan seperti yang telah ditambahkan saat menggunakan editor nano

```
root@debian:~# cat linux.txt
Ini isi dari file linux.txt
root@debian:~#
```

Gambar 3.28 Melihat perubahan isi file linux.txt

---END OF CHAPTER---

Bab 4
Konfigurasi Dasar Debian Server

Pada bab ini akan dibahas beberapa konfigurasi dasar yang perlu dilakukan untuk membuat sebuah server berbasis linux debian. Konfigurasi dasar yang akan dibahas yaitu, konfigurasi ip address, hostname, resolver, dan repository. Selain hal-hal tersebut, pada bab ini juga akan dibahas tentang penggunaan network adapter di virtualbox.

Managemen Interface

Untuk melakukan managemen interface, di linux ada sebuah perintah yang sangat berguna, yaitu *ifconfig*. Perintah ini dapat digunakan untuk melihat, menonaktifkan, ataupun mengakitfkan interface.

Pada sistem operasi linux, penamaan interface dimulai dengan eth0, eth1, eth2 dst. Jika hanya ada satu interface yang terpasang dikomputer, maka namanya adalah eth0, jika ada dua maka interface berikutnya mempunyai nama eth1, begitu juga seterusnya. Berikut contoh penggunaan perintah ifconfig

```
root@debian:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:74:c8:8c
          inet addr:192.168.56.2 Bcast:192.168.56.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe74:c88c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6157 (6.0 KiB)  TX bytes:5207 (5.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
root@debian:~#
```

Gambar 4.1 Contoh penggunaan perintah *ifconfig*

Perintah diatas digunakan untuk melihat kondisi interface di linux. Selanjutnya, untuk menonaktifakan sebuah interface, kita juga bisa menggunakan perintah ifconfig, berikut contoh penggunaannya

```
root@debian:~# ifconfig eth0 down
root@debian:~# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
root@debian:~#
```

Gambar 4.2 Menonaktifkan interface eth0

Perhatikan bahwa setelah interface eth0 dinonaktifkan, maka interface eth0 tidak akan terlihat saat kita menggunakan perintah ifconfig untuk melihat kondisi interface. Selanjutnya berikut contoh penggunaan perintah ifconfig untuk mengaktifkan interface

```
root@debian:~# ifconfig eth0 up
root@debian:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:74:c8:8c
          inet addr:192.168.56.2 Bcast:192.168.56.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe74:c88c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6157 (6.0 KiB)  TX bytes:5207 (5.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
root@debian:~#
```

Gambar 4.3 Mengaktifkan kembali interface eth0

Konfigurasi IP Address

Kita telah sedikit mengenal perintah ifconfig untuk melakukan managemen dasar interface, seperti melihat kondisi, menonakifkan, dan mengaktifkan sebuah interface. Namun perintah ifconfig tidak sebatas itu saja. Perintah ini juga bisa digunakan untuk melakukan konfigurasi ip address. Berikut contoh penggunaan ifconfig untuk konfigurasi ip address.

```
root@debian:~# ifconfig eth0 192.168.56.10/24
root@debian:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:74:c8:8c
          inet addr:192.168.56.10 Bcast:192.168.56.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe74:c88c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:140 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18554 (18.1 KiB)  TX bytes:14166 (13.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
root@debian:~#
```

Gambar 4.4 Konfigurasi ip address sementara

Perhatikan bahwa setelah menggunakan perintah pertama, ip address pada interface eth0 berganti sesuai dengan perintah tersebut saat dicek dengan perintah ifconfig (ditunjukkan tulisan warna merah).

Namun ada satu kelemahan jika kita melakukan konfigurasi ip address dengan cara diatas, yaitu konfigurasi ip address akan hilang saat komputer direstart. Untuk konfigurasi ip address secara permanen kita perlu mengedit file */etc/network/interfaces* dengan salah satu text editor, misal *nano*.

```
root@debian:~# nano /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.56.2
netmask 255.255.255.0
gateway 192.168.56.1
root@debian:~# service networking restart
[....] Running /etc/init.d/networking restart is deprecated because it may not
[warnble some interfaces ... (warning).
[ ok ] Reconfiguring network interfaces...done.
root@debian:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:74:c8:8c
          inet addr:192.168.56.2 Bcast:192.168.56.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe74:c88c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:442 errors:0 dropped:0 overruns:0 frame:0
          TX packets:278 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:44817 (43.7 KiB)  TX bytes:35237 (34.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
root@debian:~#
```

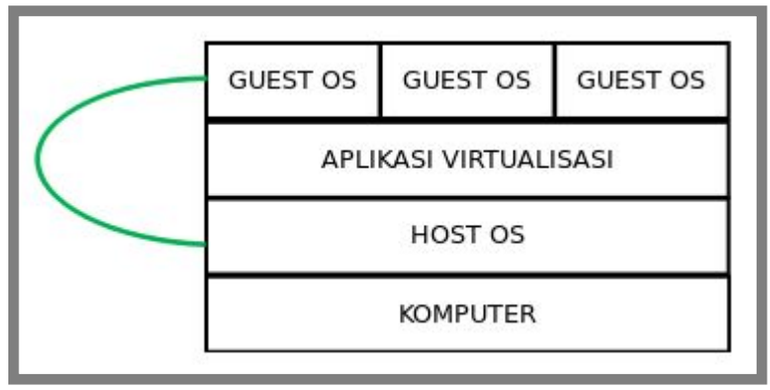
Gambar 4.5 Konfigurasi ip address permanen

Perhatikan baik-baik konfigurasi pada file */etc/network/interface* (tulisan warna hijau). Syntak yang menunjukkan konfigurasi adalah baris yang tidak diawali tanda pagar (#). Sedangkan baris yang diawali tanda pagar (#) hanya sebuah komentar. Setelah melakukan konfigurasi file tersebut, jangan lupa untuk merestart service network dengan perintah seperti diatas (*service networking restart*). Perhatikan hasil konfigurasi pada perintah terahir (*ifconfig*).

Network Adapter di Virtualbox

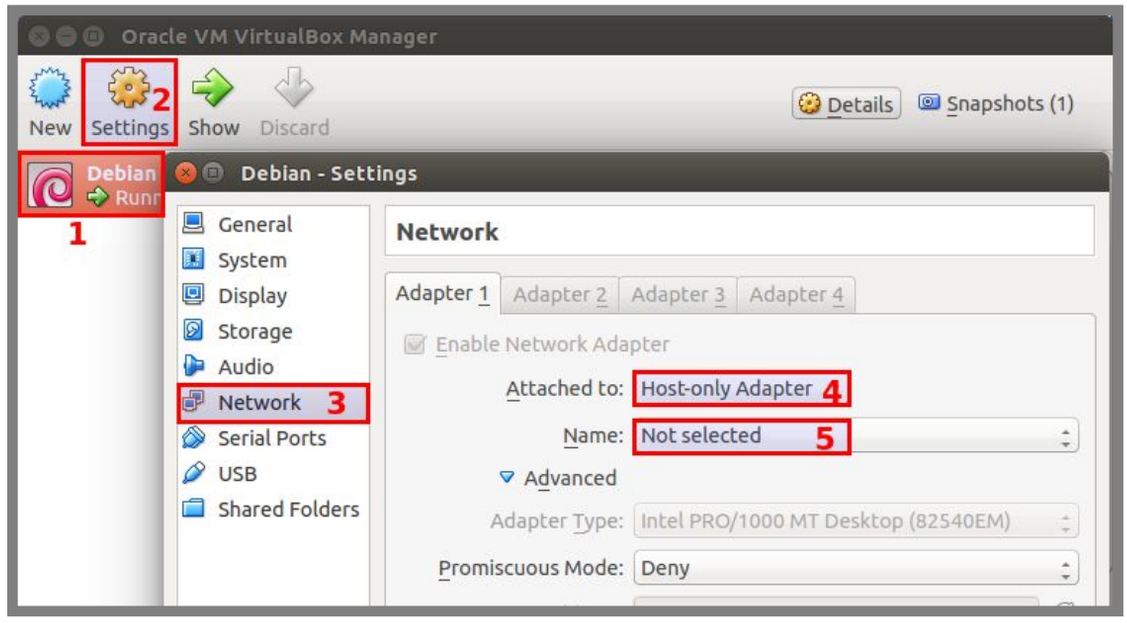
Ada beberapa hal tambahan yang harus dilakukan jika kita belajar menggunakan sebuah aplikasi virtualisasi. Hal ini tidak diperlukan jika kita belajar menggunakan dua komputer atau lebih. Hal tersebut adalah mengenai konfigurasi network adapter.

Jika kita belajar menggunakan dua komputer, untuk menghubungkannya kita hanya perlu menggunakan sebuah kabel LAN. Namun tidak demikian jika kita menggunakann aplikasi virtualisasi (virtualbox). Kita harus melakukan konfigurasi untuk menghubungkan komputer nyata (host os) dengan komputer virtual (guest os). Berikut contoh kasus yang akan kita praktikkan pada sub bab ini.



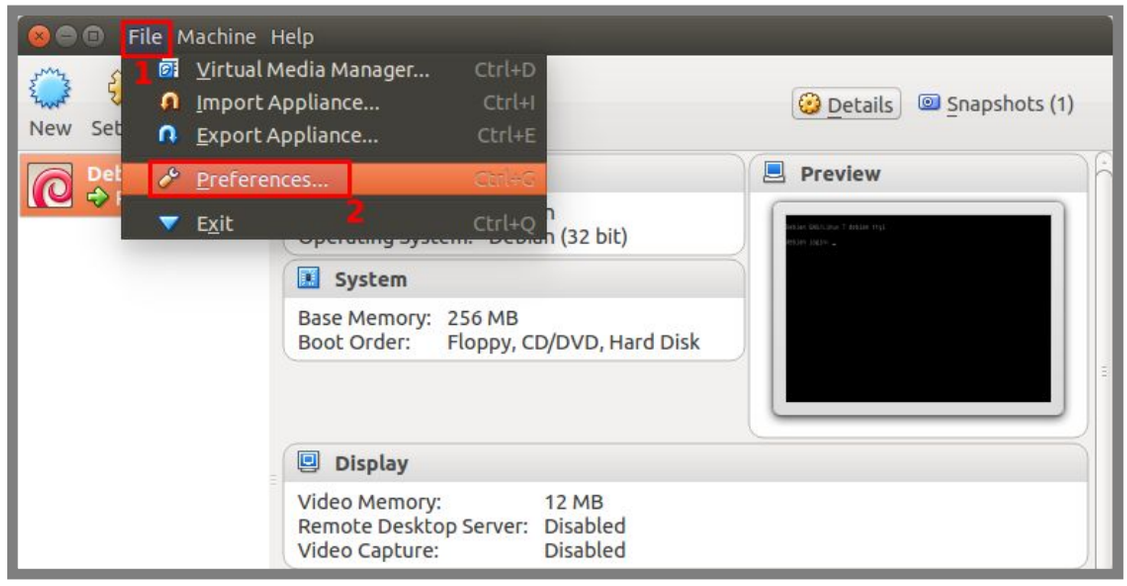
Gambar 4.6 Konsep network adapter di virtualbox

Perhatikan gambar diatas, terlihat bahwa kita berusaha menghubungkan antara sistem operasi yang berjalan diatas komputer (host os) dengan sistem operasi yang berjalan diatas aplikasi virtualisasi (guest os). Berikut konfigurasi yang diperlukan

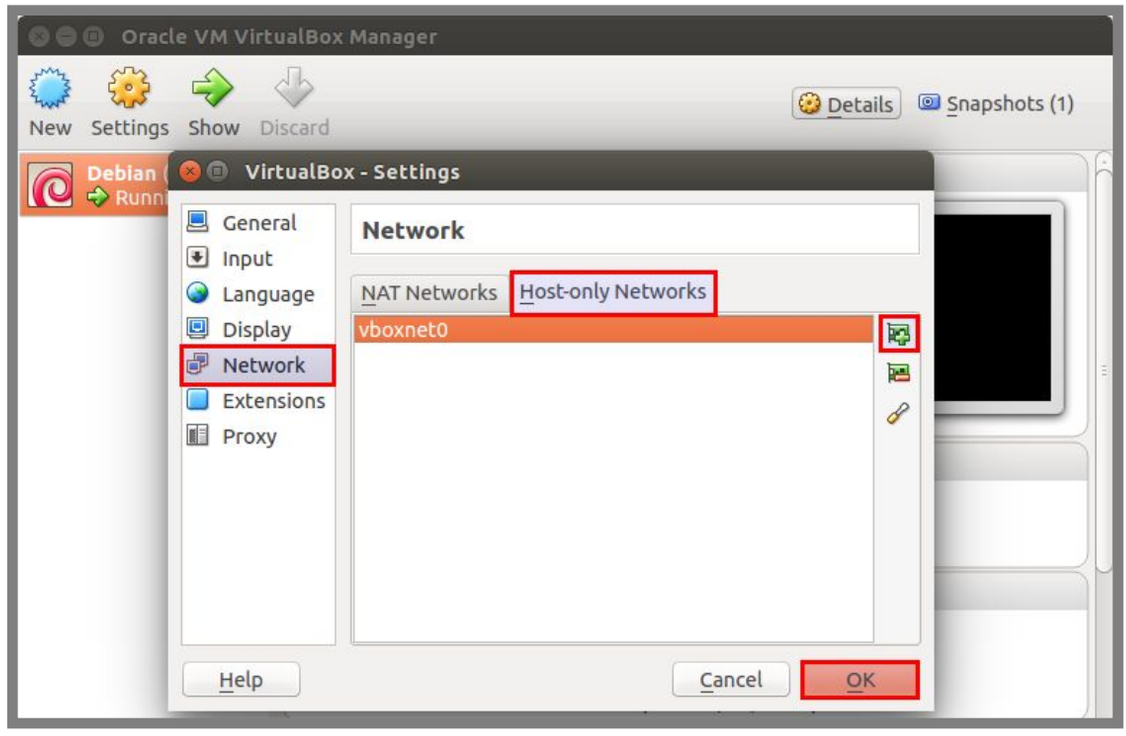


Gambar 4.7 Konfigurasi network adapter di virtul machine

Perhatikan bahwa ada sedikit error pada langkah kelima. Error diatas tidak akan ditemukan jika host os yang digunakan adalah windows, namun jika host os yang digunakan adalah ubuntu, kemungkinan besar akan muncul masalah seperti diatas. Untuk mengatasinya kita cancel dulu langkah diatas dan ikuti langkah berikut

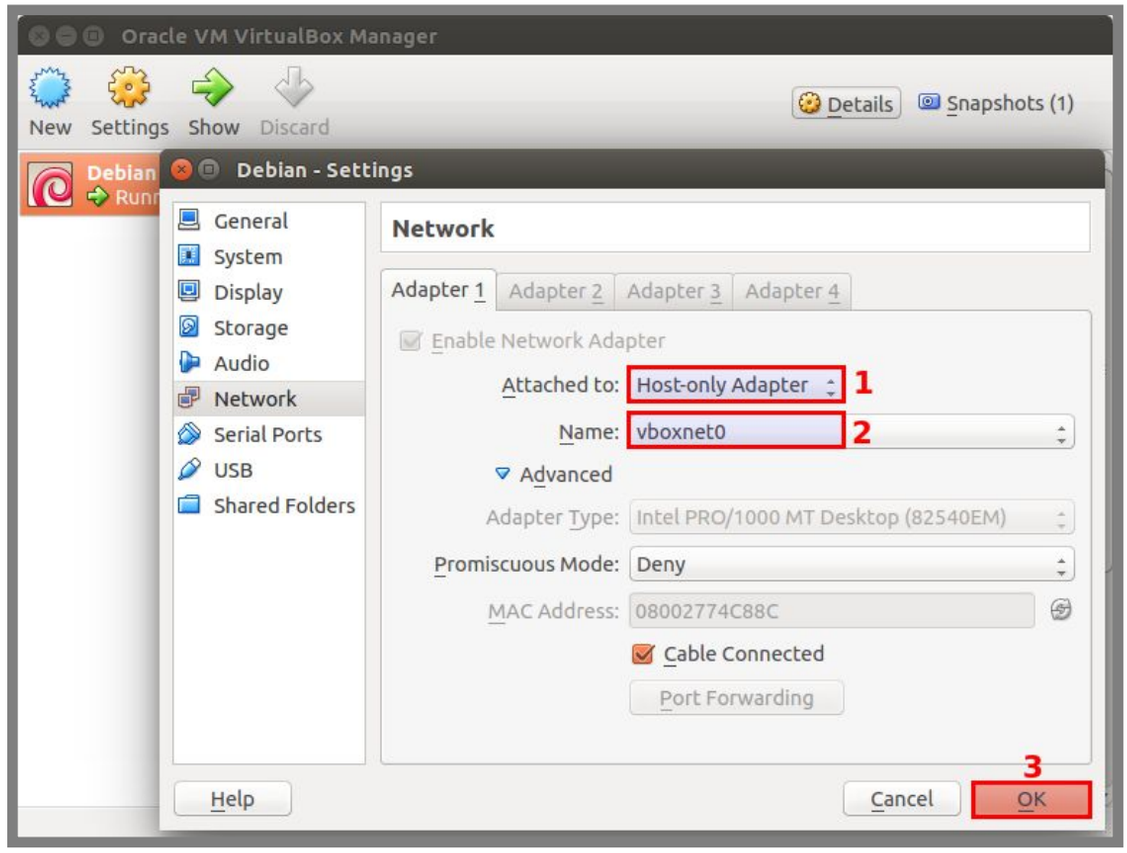


Gambar 4.8 Konfigurasi network adapter lanjutan



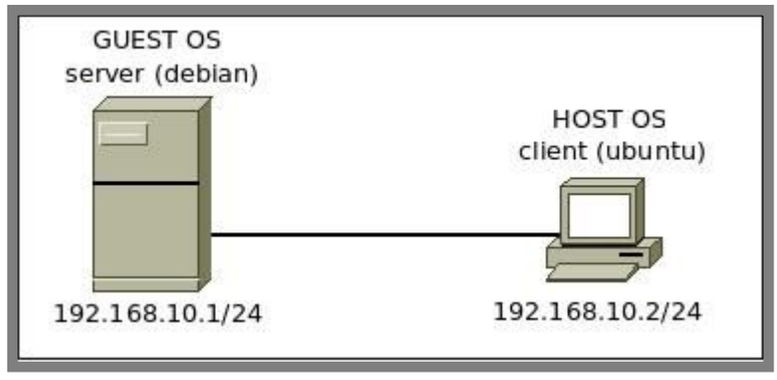
Gambar 4.9 Menambahkan host-only adapter di virtualbox

Selanjutnya kita harus mengulangi langkah yang sebelumnya gagal seperti berikut.



Gambar 4.10 Konfigurasi network adapter di virtual machine

Saat ini antara host os dan guest os sudah saling berhubungan. Langkah terahir agar kedua sistem operasi benar-benar berhubungan adalah melakukan konfigurasi ip address pada server dan client dengan ip address yang satu jaringan. Berikut topologi yang akan kita gunakan



Gambar 4.11 Topologi jaringan untuk praktik konfigurasi ip address

Berikut konfigurasi yang perlu dilakukan pada host os (client).

```
admin@ubuntu:~$ sudo ifconfig vboxnet0 192.168.10.2/24
[sudo] password for admin:
admin@ubuntu:~$ ifconfig vboxnet0
vboxnet0   Link encap:Ethernet   HWaddr 0a:00:27:00:00:00
           inet addr:192.168.10.2 Bcast:192.168.10.255 Mask:255.255.255.0
           inet6 addr: fe80::800:27ff:fe00:0/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:287 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 (0.0 B)   TX bytes:20745 (20.7 KB)
admin@ubuntu:~$
```

Gambar 4.12 Konfigurasi ip address di client

Berikut konfigurasi yang perlu dilakukan pada guest os (server).

```
root@debian:~# ifconfig eth0 192.168.10.1/24
root@debian:~# ifconfig eth0
eth0       Link encap:Ethernet   HWaddr 08:00:27:74:c8:8c
           inet addr:192.168.10.1 Bcast:192.168.10.255 Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fe74:c88c/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
           RX packets:97 errors:0 dropped:0 overruns:0 frame:0
           TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:11803 (11.5 KiB)   TX bytes:8625 (8.4 KiB)
root@debian:~#
```

Gambar 4.13 Konfigurasi ip address di server

Ingat, bahwa konfigurasi ip address dengan metode diatas hanya bersifat sementera. Sengaja digunakan metode diatas agar lebih ringkas, teman-teman bisa menggunakan metode permanen untuk praktik. Intinya mau metode manapun sebenarnya sama saja, tinggal menyesuaikan kondisi saja. Untuk membuktikan keberhasilan konfigurasi diatas, kita bisa mengujinya dengan perintah ping

```
root@debian:~# ping 192.168.10.2
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_req=1 ttl=64 time=0.999 ms
64 bytes from 192.168.10.2: icmp_req=2 ttl=64 time=0.403 ms
64 bytes from 192.168.10.2: icmp_req=3 ttl=64 time=0.627 ms
64 bytes from 192.168.10.2: icmp_req=4 ttl=64 time=0.679 ms
```

Gambar 4.14 Contoh ping dari server ke client

Konfigurasi IP Address Alias

Hampir sama halnya dengan pembasan pada sub bab sebelumnya. Namun pada sub bab ini akan dibahas tentang cara agar sebuah interface bisa memiliki lebih dari satu ip address. Sebagai contoh kasus, kita akan mengkonfigurasi interface eth0 pada debian server dengan ip address 192.168.10.1/24 dan 172.16.10.1/24. Berikut konfigurasi yang perlu dilakukan

```
root@debian:~# nano /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.10.1
netmask 255.255.255.0

auoto eth0:0
iface eth0:0 inet static
address 172.16.10.1
netmask 255.255.255.0
root@debian:~# service networking restart
[....] Running /etc/init.d/networking restart is deprecated because it may not
[warnble some interfaces ... (warning).
[ ok ] Reconfiguring network interfaces...done.
root@debian:~# ifconfig
eth0        Link encap:Ethernet  HWaddr 08:00:27:74:c8:8c
            inet addr:192.168.10.1 Bcast:192.168.10.255 Mask:255.255.255.0
            inet6 addr: fe80::a00:27ff:fe74:c88c/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:302 errors:0 dropped:0 overruns:0 frame:0
            TX packets:217 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:28767 (28.0 KiB)  TX bytes:26853 (26.2 KiB)

eth0:0      Link encap:Ethernet  HWaddr 08:00:27:74:c8:8c
            inet addr:172.16.10.1  Bcast:172.16.10.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo          Link encap:Local Loopback
            ..........
            ..........
            ..........
root@debian:~#
```

Gambar 4.15 Konfigurasi ip address alias

Konfigurasi DNS Resolver

Konfigurasi ini dilakukan jika komputer terhubung dengan internet atau ada sebuah server yang bertindak sebagai dns server. Untuk saat ini kita hanya akan mengkonfigurasi dns resolver agar menggunakan open dns yang ada di internet. Materi tentang dns server akan kita bahas di bab selanjutnya. Berikut langkah-langkah untuk melakukan konfigurasi dns resolver.

```
root@debian:~# nano /etc/resolv.conf
nameserver 180.131.144.144
nameserver 180.131.145.145
```

Gambar 4.16 Konfigurasi dns resolver

Konfigurasi Hostname

Hostname adalah nama sebuah komputer yang digunakan saat terhubung dalam suatu jaringan. Contoh kasus, misal kita menginginkan agar nama (hostname) komputer kita adalah *forkits,* maka langkah yang perlu dilakukan adalah sebagai berikut

```
root@debian:~# nano /etc/hostname
forkits
root@debian:~# nano /etc/hosts
127.0.0.1       localhost
127.0.1.1       forkits.forkits.com     forkits

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
root@debian:~# service hostname.sh start
root@debian:~# hostname -f
forkits.forkits.com
root@debian:~# su
root@forkits:~#
```

Gambar 4.17 Konfigurasi hostname

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
nano /etc/hostname	Perintah ini digunakan untuk mengedit file /etc/hostname. Modifikasi isi file tersebut sesuai dengan hostname yang diinginkan

nano /etc/hosts	Perintah ini digunakan untuk merubah file /etc/hosts. Perhatikan bahwa kita hanya melakukan perubahan pada baris kedua saja. (forkits) adalah hostname yang diinginkan, sedangkan (forkits.com) adalah domain yang digunakan. Penjelasan tentang domain akan dibahas pada bab dns server. Untuk saat ini kita bisa menggunakan domain apa saja.
service hostname.sh start	Perintah ini digunakan untuk merestart service hostname.
hostname -f	Perintah ini digunakan untuk melihat hasil konfigurasi hostname yang baru saja dilakukan. Terlihat bahwa keluaran dari perintah ini telah sesuai dengan hostname yang diinginkan
su	Perintah ini hanya digunakan agar hostname yang ditunjukkan pada *bash* berubah. Perhatikan perbedaan bash sebelumnya dengan bash setelah perintah ini.

Konfigurasi Repository

Repository adalah sebuah alamat (url) yang menunjukkan lokasi penyimpanan file-file installer saat komputer kita akan menginsatall sebuah aplikasi.

Jika kita ingat-ingat, saat kita akan menginstall suatu aplikasi di sistem operasi windows. Maka hal pertama yang kita lakukan adalah download file installer yang biasanya berextensi .exe. Selanjutnya kita tinggal menjalankan file install tersebut.

Tidak demikian dalam sistem operasi linux. Jika pada sistem operasi linux, kita harus melakukan konfigurasi repsoitory. Seperti yang telah disebutkan diatas, bahwa repository adalah alamat (url) yang menunjukkan lokasi penyimpanan file-file installer. Nantinya saat komputer akan melakukan installasi sebuah aplikasi, maka sistem operasi akan membaca file konfigurasi repository dan selanjutnya langsung mengakses lokasi repository yang ditunjukkan (oleh konfigurasi repository). Lokasi repository itu sendiri bermacam-macam, bisa di jaringan (internet), DVD, maupun di harddisk.

Berikut konfigurasi repository jika lokasi repository berada di jaringan (internet).

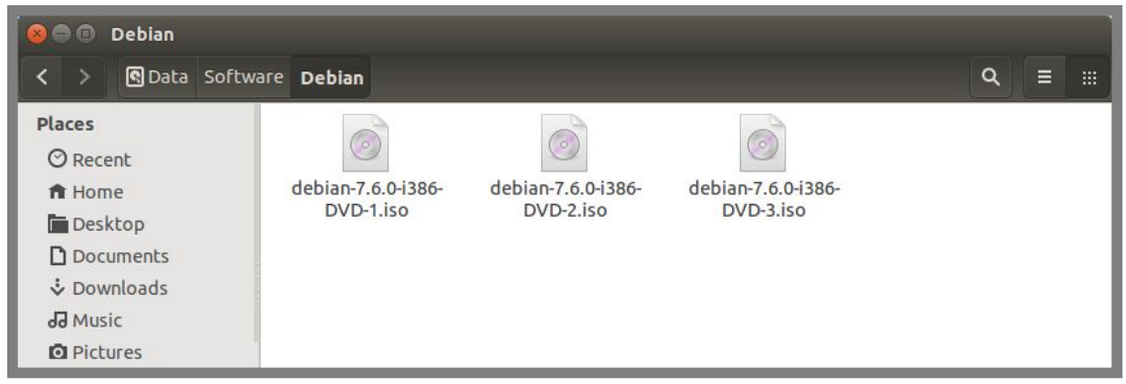
```
root@forkits:~# nano /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD Binar$

#deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD Binar$

deb http://security.debian.org/ wheezy/updates main contrib
deb-src http://security.debian.org/ wheezy/updates main contrib

# wheezy-updates, previously known as 'volatile'
# A network mirror was not selected during install.  The following entries
# are provided as examples, but you should amend them as appropriate
# for your mirror of choice.
#
deb http://ftp.debian.org/debian/ wheezy-updates main contrib
deb-src http://ftp.debian.org/debian/ wheezy-updates main contrib
root@forkits:~# apt-get update
Get:1 http://ftp.debian.org wheezy-updates Release.gpg [1554 B]
Get:2 http://ftp.debian.org wheezy-updates Release [151 kB]
Get:3 http://security.debian.org wheezy/updates Release.gpg [1554 B]
Get:4 http://ftp.debian.org wheezy-updates/main Sources [5516 B]
Get:5 http://security.debian.org wheezy/updates Release [102 kB]
Get:6 http://ftp.debian.org wheezy-updates/contrib Sources [14 B]
Get:7 http://ftp.debian.org wheezy-updates/main i386 Packages [7050 B]
Get:8 http://ftp.debian.org wheezy-updates/contrib i386 Packages [14 B]
Get:9 http://ftp.debian.org wheezy-updates/contrib Translation-en [14 B]
Get:10 http://ftp.debian.org wheezy-updates/main Translation-en [4879 B]
.....
.....
.....
Fetched 1038 kB in 2min 59s (5786 B/s)
Reading package lists... Done
root@forkits:~#
```

Gambar 4.18 Konfigurasi repository online

Perhatikan gambar diatas, perintah pertama (*nano /etc/apt/sources.list*) digunakan untuk mengedit konfigurasi repsitory. Repository yang digunakan adalah baris yang didepannya tidak ada tanda pagar (#). Sedangkan baris yang didepannya terdapat tanda pagar (#) adalah komentar. Selanjutnya perintah kedua (*apt-get update*) digunakan untuk memperbarui konfigurasi repository.

Sesuai dengan namanya (repository jaringan/internet), kita harus selalu terhubung dengan internet saat akan menginstall sebuah aplikasi jika menggunakan metode repository ini.

Tentu sangat merepotkan bagi orang-orang yang baru belajar jika harus selalu terhubung dengan internet. Maka dari itu kita bisa konfigurasi repository menggunakan DVD. Namun untuk menggunakan metode repository ini, kita membutuhkan file iso debian binary 1-3, kita bisa download ketiga file binary tersebut di internet.



Gambar 4.19 Menyiapkan dvd untuk repository

Berikut langkah-langkah yang perlu dilakukan

```
root@debian:~# nano /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD Binar$

#deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD Binary$

#deb http://security.debian.org/ wheezy/updates main contrib
#deb-src http://security.debian.org/ wheezy/updates main contrib

# wheezy-updates, previously known as 'volatile'
# A network mirror was not selected during install.  The following entries
# are provided as examples, but you should amend them as appropriate
# for your mirror of choice.
#
#deb http://ftp.debian.org/debian/ wheezy-updates main contrib
#deb-src http://ftp.debian.org/debian/ wheezy-updates main contrib
root@forkits:~#
```

Gambar 4.20 Menonaktifkan repository online

Yang dilakukan pada langkah diatas adalah menonaktifkan seluruh repository (memberikan tanda pagar pada semua baris repository).

Langkah selanjutnya, masukkan dvd debian binary 1, kemudian gunakan perintah seperti berikut.

```
root@forkits:~# mkdir /media/cd
root@forkits:~# mount /dev/cdrom /media/cd
mount: block device /dev/sr0 is write-protected, mounting read-only
root@forkits:~# apt-cdrom add
Using CD-ROM mount point /media/cd/
Identifying.. [a6b0f6ac390153bdcd38af3853fad113-2]
Scanning disc for index files..
Found 2 package indexes, 0 source indexes, 2 translation indexes and 0
signatures
This disc is called:
'Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD Binary-1
20140712-13:02'
Reading Package Indexes... Done
Reading Translation Indexes... Done
Writing new source list
Source list entries for this disc are:
deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD Binary-1
20140712-13:02]/ wheezy contrib main
Repeat this process for the rest of the CDs in your set.
root@forkits:~#
```

Gambar 4.20 Menambahkan dvd 1 ke repository

Berikut penjelasan dari masing-masing perintah diatas

Perintah	Deskripsi
mkdir /media/cd	Perintah ini digunakan untuk membuat direktori *cd* didalam direktori */media/*
mount /dev/cdrom /media/cd	Perintah ini digunakan untuk mount cdrom ke direktori */media/cd*
apt-cdrom add	Perintah ini digunakan untuk menambahkan baris repository dvd binary 1 ke file konfigurasi repository (/etc/apt/soruces.list). Perhatikan teks warna merah, terlihat bahwa dvd binary 1 telah berhasil ditambahkan.

Selanjutnya masukkan dvd debian binary 2, kemudian gunakan perintah sebagai berikut.

```
root@forkits:~# umount /media/cd
root@forkits:~# mount /dev/cdrom /media/cd
mount: block device /dev/sr0 is write-protected, mounting read-only
root@forkits:~# apt-cdrom add
Using CD-ROM mount point /media/cd/
Identifying.. [cb0d296b67a07d78e88cceca3caf51d8-2]
Scanning disc for index files..
.....
.....
.....
Writing new source list
Source list entries for this disc are:
deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD Binary-2
20140712-13:02]/ wheezy contrib main
Repeat this process for the rest of the CDs in your set.
root@forkits:~#
```

Gambar 4.21 Menambahkan dvd 2 ke repository

Perbedaan langkah diatas dengan sebelumnya hanya pada perintah pertama (umount /media/cd). Dimana perintah tersebut digunakan untuk mengumount cdrom yang sebelumnya (dvd binary 1). Perhatikan bahwa dvd binary 2 juga telah berhasil ditambahkan ke konfigurasi repository (teks warna merah). Selanjutnya masukkan dvd debian binary 3 dan lakukan langkah yang sama.

```
root@forkits:~# umount /media/cd
root@forkits:~# mount /dev/cdrom /media/cd
mount: block device /dev/sr0 is write-protected, mounting read-only
root@forkits:~# apt-cdrom add
Using CD-ROM mount point /media/cd/
Identifying.. [3d6580863edf0350b9bfabcb143290bb-2]
Scanning disc for index files..
.....
.....
.....
Writing new source list
Source list entries for this disc are:
deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD Binary-3
20140712-13:02]/ wheezy contrib main
Repeat this process for the rest of the CDs in your set.
root@forkits:~#
```

Gambar 4.22 Menambahkan dvd 3 ke repository

Langkah terahir yang perlu dilakukan adalah melakukan update,

```
root@forkits:~# apt-get update
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-3 20140712-13:02] wheezy Release.gpg
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-2 20140712-13:02] wheezy Release.gpg
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-1 20140712-13:02] wheezy Release.gpg
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-3 20140712-13:02] wheezy Release
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-2 20140712-13:02] wheezy Release
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-1 20140712-13:02] wheezy Release
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-3 20140712-13:02] wheezy/contrib i386 Packages/DiffIndex
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-3 20140712-13:02] wheezy/main i386 Packages/DiffIndex
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-2 20140712-13:02] wheezy/contrib i386 Packages/DiffIndex
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-2 20140712-13:02] wheezy/main i386 Packages/DiffIndex
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-1 20140712-13:02] wheezy/contrib i386 Packages/DiffIndex
Ign cdrom://[Debian GNU/Linux 7.6.0 _Wheezy_ - Official i386 DVD
Binary-1 20140712-13:02] wheezy/main i386 Packages/DiffIndex
Reading package lists... Done
root@forkits:~#
```

Gambar 4.23 Melakukan update repository

Kelemahan repository metode dvd seperti diatas adalah, kita harus sering melepas dan mengganti dvd binary yang satu dengan dvd binary lain saat proses installasi suatau aplikasi seperti ini

```
root@forkits:~# apt-get install bind9
.....
.....
.....
After this operation, 1257 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
Media change: please insert the disc labeled
'Debian GNU/Linux 7.6.0_Wheezy_-Official i386 DVD Binary-1 20142-13:02'
in the drive '/media/cdrom/' and press enter
```

Gambar 4.24 Contoh installasi aplikasi

Terlihat pada gambar diatas bahwa saat melakukan installasi aplikasi bind9, kita diminta untuk memasukkan dvd debian binary 1 (teks warna merah). Hal seperti itu juga akan terus terjadi saat kita menginstall aplikasi yang membutuhkan dvd binary 2, ataupun binary 3.

Hal seperti itu tentu akan sangat merepotkan, oleh sebab itu kita akan belajar mengatur repository agar menggunakan harddisk. Untuk menggunakan repository dari harddisk, kita harus mengcopy seluruh file dvd debian binary 1-3 kedalam harddisk kita. Kita akan melakukannya secara bertahap, pertama masukkan dvd binary 1 dan gunakan perintah berikut

```
root@forkits:~# mkdir /repo
root@forkits:~# mkdir /repo/dvd1
root@forkits:~# mount /dev/cdrom /media/cd
mount: block device /dev/sr0 is write-protected, mounting read-only
root@forkits:~# cp /media/cd/* /repo/dvd1/ -rf
root@forkits:~# ls /repo/dvd1/
README.html          autorun.inf  firmware     isolinux     tools
README.mirrors.html  css          g2ldr        md5sum.txt   win32-loader.ini
README.mirrors.txt   debian       g2ldr.mbr    pics
README.source        dists        install      pool
README.txt           doc          install.386  setup.exe
root@forkits:~#
```

Gambar 4.25 Copy file dvd 1 ke harddisk untuk repository lokal

Berikut keterangan dari masing-masing perintah diatas

Perintah	Deskripsi
mkdir /repo	Digunakan untuk membuat direktori /repo. Direktori ini nantinya akan kita gunakan untuk menyimpan seluruh repository dvd binary 1-3.
mkdir /repo/dvd1	Perintah ini digunakan untuk membuat direktori dvd1 didalam direktori /repo. Direktori ini nantinya akan kita gunakan untuk menyimpan repository dvd binary 1.
mount /dev/cdrom /media/cd	Perintah ini digunakan untuk mount cdrom ke direktori /media/cd.
cp /media/cd/* /repo/dvd1/ -rf	Perintah ini digunakan untuk mengcopy seluruh isi file /media/cd/ ke direktori /repo/dvd1. Ingat bahwa kita telah memount dvd binary 1 ke direktori /media/cd dengan perintah sebelumnya. Jadi perintah ini digunakan untuk mengcopy seluruh file dvd binary 1 ke direktori /repo/dvd1. Proses copy cukup lama.
ls /repo/dvd1	Perintah ini digunakan untuk melihat hasil copy yang telah kita lakukan sebelumnya.

Selanjutnya masukkan dvd binary 2 dan lakukan langkah yang sama dengan sebelumnya.

```
root@forkits:~# mkdir /repo/dvd2
root@forkits:~# umount /media/cd
root@forkits:~# mount /dev/cdrom /media/cd
mount: block device /dev/sr0 is write-protected, mounting read-only
root@forkits:~# cp /media/cd/* /repo/dvd2/ -rf
root@forkits:~# ls /repo/dvd2/
README.html          README.mirrors.txt  css   dists md5sum.txt  pool
README.mirrors.html README.txt           debian firmware  pics
root@forkits:~#
```

Gambar 4.26 Copy file dvd 2 ke harddisk untuk repository lokal

Berikutnya masukkan dvd binary 3 dan lakukan langkah yang sama seperti diatas

```
root@forkits:~# mkdir /repo/dvd3
root@forkits:~# umount /media/cd
root@forkits:~# mount /dev/cdrom /media/cd
mount: block device /dev/sr0 is write-protected, mounting read-only
root@forkits:~# cp /media/cd/* /repo/dvd3/ -rf
root@forkits:~# ls /repo/dvd3/
README.html          README.mirrors.txt  css   dists md5sum.txt  pool
README.mirrors.html README.txt           debian firmware  pics
root@forkits:~#
```

Gambar 4.27 Copy file dvd 3 ke harddisk untuk repository lokal

Saat ini kita telah mempunyai seluruh file repository didalam harddisk. Yang perlu dilakukan selanjutnya adalah melakukan konfigurasi repository dan update.

```
root@forkits:~# nano /etc/apt/sources.list
deb file:/repo/dvd1 wheezy main contrib
deb file:/repo/dvd2 wheezy main contrib
deb file:/repo/dvd3 wheezy main contrib
root@forkits:~# apt-get update
Ign file: wheezy Release.gpg
Ign file: wheezy Release.gpg
Ign file: wheezy Release.gpg
Get:1 file: wheezy Release [18.5 kB]
Get:2 file: wheezy Release [17.7 kB]
Get:3 file: wheezy Release [14.2 kB]
Reading package lists... Done
root@forkits:~#
```

Gambar 4.28 Konfigurasi repository lokal

Perhatikan bahwa pada file konfigurasi repository (*/etc/apt/sources.list*), hanya terdapat tiga baris konfigurasi repository. Hal ini karena kita hanya akan menggunakan repository dari harddisk. Jadi repository jaringan/internet dan dvd sudah tidak kita butuhkan lagi, sehingga kita bisa menonaktifkannya dengan memberikan tanda pagar (#) didepannya atau menghapusnya.

Setelah merubah konfigurasi repository, jangan lupa melakukan update dengan perintah apt-get update seperti perintah kedua diatas.

Sampai saat ini kita telah belajar konfigurasi repository menggunakan tiga metode, yaitu repository melalui jaringan/internet, dvd, dan harddisk. Dari ketiga metode tersebut yang terbaik tetap yang menggunakan jaringan/internet, karena file-file repository akan selalu up to date. Namun jika hanya untuk keperluan belajar, repository dvd ataupun harddisk tidak terlalu menjadi masalah.

---END OF CHAPTER---

Bab 5
Remote Access di Debian

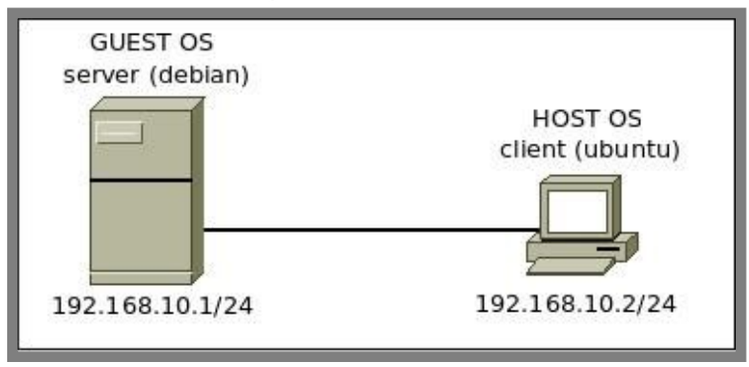
Pada bab ini akan dibahas sebuah fitur di linux debian yang memungkinkan untuk melakukan konfigurasi server debian tanpa harus berada didepan komputer server.

Fitur remote access akan sangat lazim digunakan jika kita bekerja di dunia nyata. Karena pada umumnya, komputer server diletakkan pada sebuah ruangan tertutup dengan suhu yang sangat dingin dan tidak bersahabat dengan manusia. Oleh karena itu kita tidak bisa terlalu lama berada didalam ruang server.

Untuk mengatasi masalah tersebut, kita bisa menggunakan fasilitas remote access untuk meremote server dari luar ruang server. Bukan hanya dari luar ruangan, kita bahkan bisa meremote server dari luar negeri asalkan terhubung dengan jaringan internet.

Remote Access dengan Telnet

Telnet adalah salah satu aplikasi yang bisa kita manfaatkan untuk remote access. Namun ada sebuah syarat paling dasar untuk melakukan remote access, yaitu antara komputer client dan server harus dapat saling berkomunkasi. Untuk itu, pelajaran konfigurasi ip address yang telah kita bahas di bab sebelumnya akan sangat bermanfaat pada bab ini. Berikut topologi yang akan kita praktikkan untuk memperlajari remote access dengan telnet



Gambar 5.1 Topologi jaringan untuk praktik remote access

Tidak akan dibahas tentang konfigurasi ip address pada bab ini, namun akan diberikan hasil ping yang membuktikan client dan server telah terhubung

```
admin@ubuntu:~$ ifconfig vboxnet0
vboxnet0   Link encap:Ethernet   HWaddr 0a:00:27:00:00:00
           inet addr:192.168.10.2 Bcast:192.168.10.255   Mask:255.255.255.0
           inet6 addr: fe80::800:27ff:fe00:0/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:120 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 (0.0 B)   TX bytes:17388 (17.3 KB)
admin@ubuntu:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.449 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.379 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.473 ms
```

Gambar 5.2 Pengujian koneksi jaringan dari client ke server

Jika client dan server telah terhubung, langkah selanjutnya kita harus install aplikasi telnet server di komputer server

```
root@forkits:~# apt-get install telnetd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libfile-copy-recursive-perl openbsd-inetd update-inetd
The following NEW packages will be installed:
  libfile-copy-recursive-perl openbsd-inetd telnetd update-inetd
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/122 kB of archives.
After this operation, 407 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
WARNING: The following packages cannot be authenticated!
  libfile-copy-recursive-perl update-inetd openbsd-inetd telnetd
Install these packages without verification [y/N]? y
```

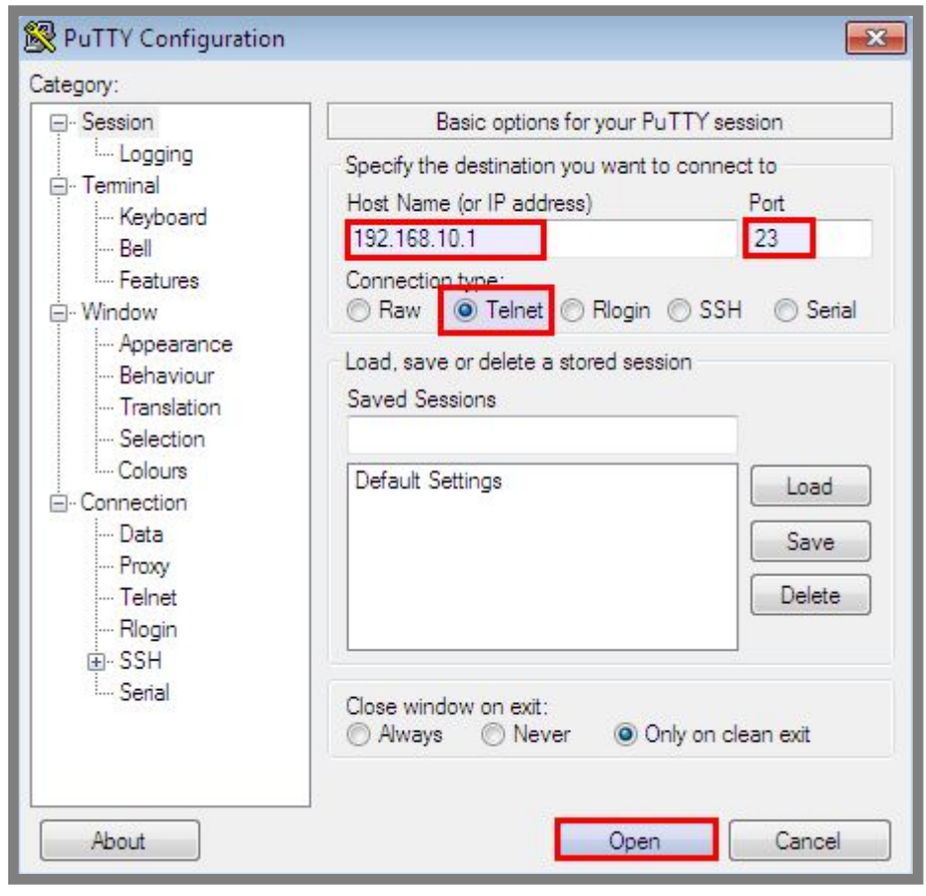
Gambar 5.3 Installasi aplikasi telnet server

Jika installasi sudah selesai, kita sudah bisa meremote server dari client. Untuk meremote server, kita bisa menggunakan terminal jika sistem operasi client yang digunakan adalah linux. Namun jika sistem operasi client yang digunakan adalah windows, kita bisa memanfaatkan aplikasi putty (silahkan download dari internet). Berikut pengujian saat menggunakan client linux ubuntu.

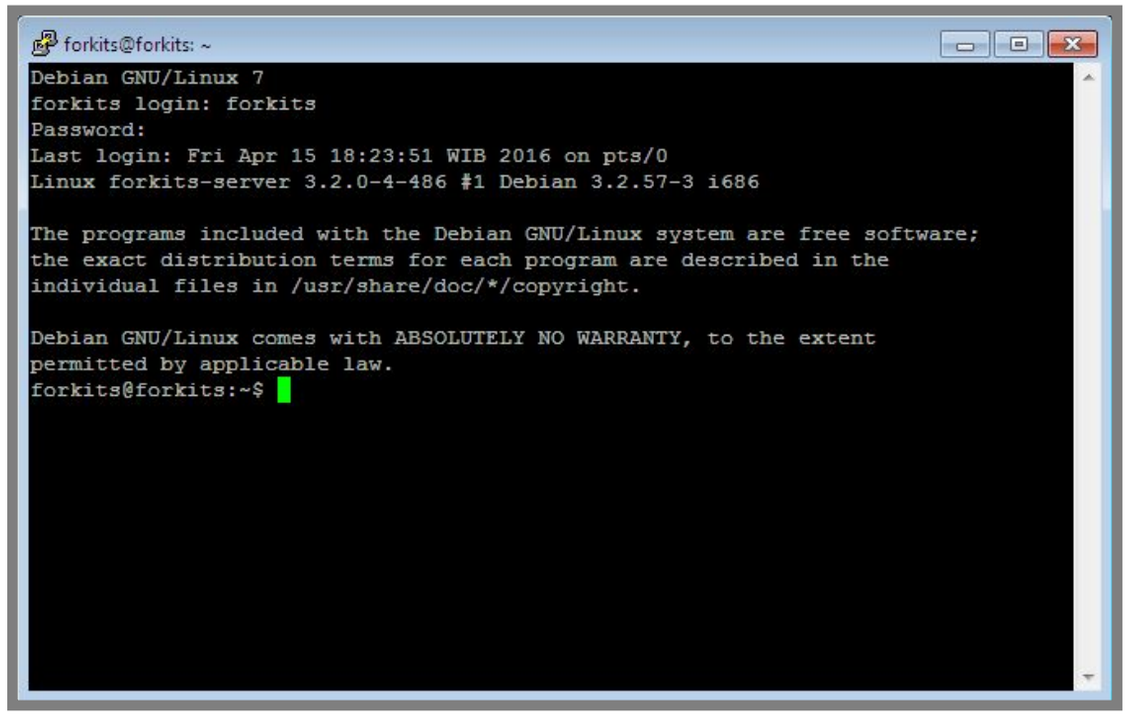
```
admin@ubuntu:~$ telnet 192.168.10.1
Trying 192.168.10.1...
Connected to 192.168.10.1.
Escape character is '^]'.
Debian GNU/Linux 7
forkits login: forkits
Password: (tidak terlihat)
Last login: Sat Apr  9 19:30:10 WIB 2016 from 192.168.56.1 on pts/0
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
forkits@forkits:~$
```

Gambar 5.4 Melakukan remote access dari client ubuntu ke server

Berikut pengujian saat client yang digunakan adalah windows 7



Gambar 5.5 Melakukan remote access dari client windows ke server



Gambar 5.6 Tampilan pertama saat melakukan remote access dari client windows

Perlu diketahui bahwa jika kita meremote server menggunakan telnet, maka kita tidak akan bisa login sebagai root. Jadi kita harus login sebagai user biasa, kemudian berpindah ke user root dengan perintah su jika memerlukan hak akses root.

Merubah Default Port Telnet

Secara default, telnet menggunakan port 23. Jika kita menggunakan port default tersebut, maka kemungkinan orang-orang yang tidak bertanggung jawab membobol server kita akan sangat besar, karena semua orang tahu pasti bahwa port default yang digunakan telnet adalah 23. Untuk itu, sangat disarankan merubah port default telnet dengan alasan keamanan.

Kita akan tetap menggunakan topologi yang sama dengan pembahasan sebelumnya, hanya saja kita akan mengganti port telnet menjadi 29. Namun sebelumnya, kita harus memastikan apakah port 29 digunakan oleh service lain atau. Jika port 29 ternyata telah digunakan oleh service lain, maka kita harus mencari port lain

```
root@forkits:~# netstat -tanp | grep 29
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
2298/inetd
tcp        0      0 192.168.10.1:22         192.168.10.2:37229      ESTABLISHED
2387/0
root@forkits:~#
```

Gambar 5.7 Mencari port yang tidak terpakai

Perintah ditas digunakan untuk melihat seluruh service yang berjalan di server (*netstat -tanp*), kemudian dilanjutkan dengan perintah untuk mencari angka 29 (*grep 29*). Nomor yang menunjukkan port adalah teks dengan warna merah, terlihat bahwa tidak ada service yang menggunakan port 29. Berarti kita bisa menggunakan port 29 untuk telnet. Berikut langkah yang perlu dilakukan untuk merubah port telnet

```
root@forkits:~# nano /etc/services
.....
.....
.....
ssh             22/tcp          # SSH Remote Login Protocol
ssh             22/udp
telnet          29/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
.....
.....
.....
root@forkits:~# reboot
```

Gambar 5.8 Merubah port telnet

Perintah pertama (nano /etc/services) digunakan untuk mengedit file /etc/services, pada file ini kita akan merubah port telnet dari 23 menjadi 29. Cari pada bagian yang ditunjukkan dengan teks warna hijau. Untuk mempercepat pencarian, gunakan fasilitas pencarian dengan menekan tombol kombinasi *ctrl+w* kemudian ketikkan kata kunci pencarian dengan *telnet.* Selanjutnya rubah angka 23 menjadi 29 seperti teks warna hijau. Simpan perubahan yang dilakukan dan restart komputer dengan perintah kedua (*reboot*). Berikut pengujian yang dilakukan dari client linux ubuntu

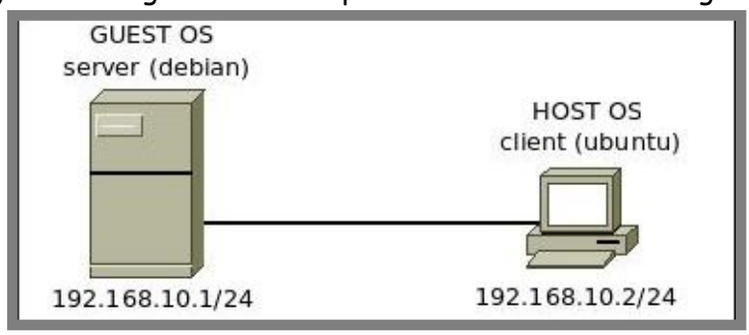
```
admin@ubuntu:~$ telnet 192.168.10.1 -l forkits -r 29
Trying 192.168.10.1...
Connected to 192.168.10.1.
Escape character is '^]'.
Password: (tidak terlihat)
Last login: Tue Apr 12 15:22:54 WIB 2016 on pts/1
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
forkits@forkits:~$
```

Gambar 5.9 Meremote server dari client menggunakan telnet

Perhatikan perintah yang digunakan diatas, terlihat bahwa ada sedikit perbedaan dengan perintah yang digunakan pada sub bab sebelumnya. Option (-l) menunjukkan user yang digunakan, sedangkan option (-r) menunjukkan port yang digunakan.

Remote Access dengan SSH

Selain menggunakan telnet, kita juga bisa menggunakan ssh untuk meremote server. Kenyataannya, ssh lebih sering digunakan daripada telnet, hal ini karena ssh memiliki tingkat keamanan yang jauh lebih tinggi daripada telnet. Berikut topologi yang akan kita gunakan untuk praktik remote access dengan ssh



Gambar 5.10 Topologi jaringan untuk praktik ssh

Diasumsikan bahwa komputer server dan client telah dikonfigurasi dengan ip address sesuai topologi diatas dan sudah bisa saling berkomunikasi. Selanjutnya install aplikasi ssh server di komputer server

```
root@forkits:~# apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  openssh-client
Suggested packages:
  ssh-askpass libpam-ssh keychain monkeysphere rssh molly-guard ufw
The following NEW packages will be installed:
  openssh-server
The following packages will be upgraded:
  openssh-client
1 upgraded, 1 newly installed, 0 to remove and 74 not upgraded.
Need to get 1,387 kB of archives.
After this operation, 669 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 5.11 Installasi aplikasi ssh server

Untuk pengujian, kita bisa menggunakan terminal pada sistem operasi linux, atau putty pada sistem operasi client sebagai ssh client.

```
admin@ubuntu:~$ ssh 192.168.10.1 -l root
root@192.168.10.1's password: (tidak terlihat)
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 12 16:21:56 2016
root@forkits:~#
```

Gambar 5.12 Melakukan remote access dari client menggunakan ssh

Perintah diatas digunakan untuk meremote server (192.168.10.1) menggunakan user root. Ingat, bahwa option (-l) menunjukkan user yang digunakan.

Merubah Default Port SSH

Ingat alasan kenapa kita harus merubah port pada telnet? Alasan tersebut juga berlaku untuk ssh. Secara default, ssh menggunakan port 22 dan saat ini kita akan merubahnya menjadi 242. Sebelumnya pastikan bahwa tidak ada service lain yang menggunakan port 242

```
root@forkits:~# netstat -tanp | grep 242
root@forkits:~#
```

Gambar 5.13 Mencari port yang tidak terpakai

Perhatikan bahwa tidak ada keluaran dari perintah diatas, hal ini menunjukkan bahwa tidak ada service lain yang menggunakan port 242. Lakukan langkah berikut untuk merubah port ssh dari 22 menjadi 242

```
root@forkits:~# nano /etc/ssh/sshd_config
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 242
# Use these options to restrict which interfaces/protocols sshd will bind to
.....
.....
.....
root@forkits:~# service ssh restart
[ ok ] Restarting OpenBSD Secure Shell server: sshd.
root@forkits:~#
```

Gambar 5.14 Mengganti port ssh

Perhatikan gambar diatas, perintah pertama (nano /etc/ssh/sshd_config) digunakan untuk merubah konfigurasi ssh, yaitu merubah port 22 menjadi 242 (rubah pada bagian teks warna hijau). Selanjutnya, restar service ssh dengan perintah kedua (*service ssh restart*). Berikut pengujian yang dilakukan dari client

```
admin@ubuntu:~$ ssh 192.168.10.1 -l root -p 242
root@192.168.10.1's password: (tidak terlihat)
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 12 16:28:17 2016 from 192.168.10.2
root@forkits:~#
```

Gambar 5.15 Pengujian ssh dari client

Perhatikan perintah diatas, option (-p) menunjukkan port yang digunakan oleh ssh server.

Disable Root Login SSH

User root merupakan user yang memiliki hak akses tertinggi pada server. Tentu akan sangat berbahaya jika password user root diketahui oleh orang lain yang tidak bertanggungjawab.

Saat kita melakukan remote ke server menggunakan ssh ataupun telnet, username, password, dan perintah-perintah yang kita gunakan, akan dikirimkan melalui jaringan. Namun tentu saja username, password, dan perintah-perintah itu akan dikirimkan dalam bentuk terenkripsi (semacam kata sandi). Meskipun demikian, bukan berarti tidak mungkin untuk mememecahkan enkripsi tersebut dan membacanya. Jika password root sampai bisa dibaca oleh orang lain, akan sangat fatal akibatnya. Oleh sebab itu, kita harus menghindari melakukan remote access menggunakan user root.

Berikut langkah yang perlu dilakukan untuk melarang user root untuk login ssh

```
root@forkits:~# nano /etc/ssh/sshd_config
.....
.....
.....
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
.....
.....
.....
root@forkits:~# service ssh restart
[ ok ] Restarting OpenBSD Secure Shell server: sshd.
root@forkits:~#
```

Gambar 5.16 Melarang user root login ssh

Cari dan ganti pada bagian teks warna hijau. Jangan lupa untuk merestart service ssh setelah melaukan perubahan konfigurasi. Berikut hasil uji coba dari client

```
admin@ubuntu:~$ ssh 192.168.10.1 -l root -p 242
root@192.168.10.1's password: (tidak terlihat)
Permission denied, please try again.
root@192.168.10.1's password:
```

Gambar 5.17 Pengujian login ssh dengan user root

Perhatikan teks warna merah, tertulis bahwa permisi ditolak, hal ini karena pada contoh diatas kita login sebagai user root. Berikut hasilnya saat kita login menggunakan user biasa

```
admin@ubuntu:~$ ssh 192.168.10.1 -l forkits -p 242
forkits@192.168.10.1's password: (tidak terlihat)
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 12 16:22:18 2016
forkits@forkits:~$
```

Gambar 5.18 Pengujian login ssh dengan user biasa

Perhatikan gambar diatas, terlihat bahwa kita berhasil login saat menggunakan user biasa.

Membuat User Setara Root

Tentu kita tidak akan nyaman jika tidak bisa menggunakan user root saat melakukan remote ke server melalui telnet ataupun ssh. Hal ini karena kita tidak bisa melakukan managemen server dengan maksimal tanpa user root. Namun tidak perlu hawatir, kita bisa membuat suatu user biasa yang memiliki hak akses setara root. Langkah pertama, buat user dengan cara seperti biasanya (jangan menggunakan password yang sama dengan root)

```
root@forkits:~# adduser administrator
.....
.....
Enter new UNIX password: (tak terlihat)
Retype new UNIX password: (tak terlihat)
passwd: password updated successfully
Changing the user information for administrator
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
root@forkits:~#
```

Gambar 5.19 Membuat user baru

Langkah selanjutnya kita perlu menginstall aplikasi sudo. Gunakan perintah seperti berikut

```
root@forkits:~# apt-get install sudo
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sudo
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/851 kB of archives.
After this operation, 1885 kB of additional disk space will be used.
Install these packages without verification [y/N]? y
```

Gambar 5.20 Menginstall aplikasi untuk managemen user (sudo)

Selanjutnya, konfigurasi sudo dengan perintah seperti ini

```
root@forkits:~# nano /etc/sudoers
.....
.....
# User privilege specification
root            ALL=(ALL:ALL) ALL
administrator   ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
.....
.....
```

Gambar 5.21 Konfigurasi sudo

Pada file diatas, tambahkan sebuah baris yang sama dengan teks warna hijau. Sesuaikan administrator dengan user yang dibuat. Untuk pengujian, coba login ssh menggunakan user tersebut

```
admin@ubuntu:~$ ssh 192.168.10.1 -l administrator -p 242
administrator@192.168.10.1's password: (tidak terlihat)
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
administrator@forkits:~$
```

Gambar 5.22 Login ssh dari komputer client

Berikut bukti bahwa user ini memiliki hak akses setara root

```
administrator@forkits:~$ touch /media/administrator.file
touch: cannot touch `/administrator.file': Permission denied
administrator@forkits:~$ sudo touch /media/administrator.file

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for administrator: (tak terlihat)
administrator@forkits:~$ ls /media/
administrator.file  bsd.txt  cd  cdrom  cdrom0  linux  unix
administrator@forkits:~$
```

Gambar 5.23 Pengujian hak akses setara root

Perhatikan gambar diatas, terlihat bahwa saat mengetikkan perintah pertama, ada sebuah peringatan bahwa hak akses ditolak. Hal ini karena direktori /media hanya bisa dimodifikasi oleh user root atau user biasa yang memiliki hak akses setara root. Namun perhatikan perintah kedua, terlihat bahwa kita diminta untuk melakukan authentikasi dan berhasil membuat file di direktori /media. Hal tersebut membuktikan bahwa user administrator memiliki hak akses setara root saat menambahkan syntak *sudo* didepan setiap perintahnya.

Membatasi Akses SSH pada User

Pada sub bab ini, akan dibahas cara untuk mengkonfigurasi ssh agar hanya mengizinkan user tertentu saja yang bisa melakukan remote access. Dalam hal ini, kita hanya akan mengizinkan user administrator saja, selain user tersebut, tidak akan diperbolehkan melakukan remote access. Berikut konfigurasi yang perlu dilakukan

```
root@forkits:~# nano /etc/ssh/sshd_config
# Package generated configuration file
# See the sshd_config(5) manpage for details
AllowUsers administrator
# What ports, IPs and protocols we listen for
.....
.....
root@forkits:~# service ssh restart
[ ok ] Restarting OpenBSD Secure Shell server: sshd.
root@forkits:~#
```

Gambar 5.24 Membatasi akses ssh untuk user tertentu

Perhatikan gambar diatas, terlihat bahwa kita menambahkan sebuah baris (ditunjukkan teks warna hijau) pada konfigurasi ssh. Selanjutnya jangan lupa untuk merestart service ssh. Berikut hasil pengujian yang dilakukan dari client

```
admin@ubuntu:~$ ssh 192.168.10.1 -l forkits -p 242
forkits@192.168.10.1's password: (tidak terlihat)
Permission denied, please try again.
admin@ubuntu:~$ ssh 192.168.10.1 -l administrator -p 242
administrator@192.168.10.1's password: (tidak terlihat)
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 13 05:08:13 2016 from 192.168.10.2
administrator@forkits:~$
```

Gambar 5.25 Pengujian ssh dari komputer client

Perhatikan gambar diatas, terlihat bahwa ada peringatan hak akses ditolak saat kita mencoba login ssh dengan user forkits, namun tidak demikian saat kita login ssh dengan user administrator.

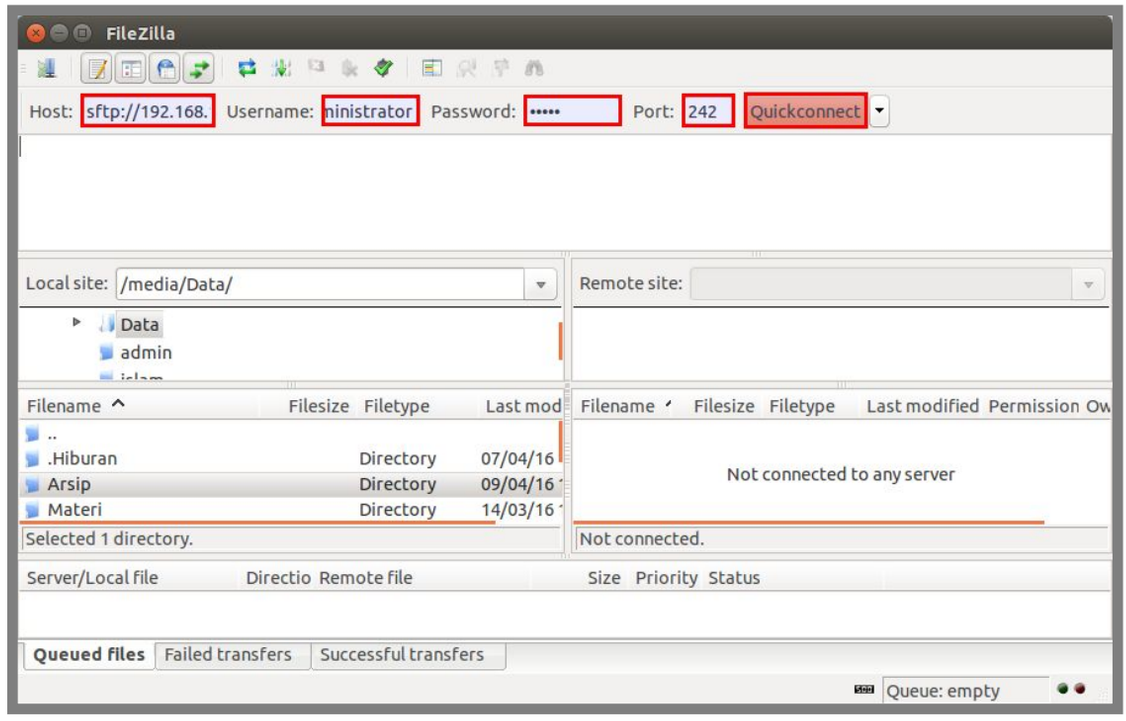
File Transfer dengan SFTP

Secure File Transfer Protocol (SFTP) adalah salah satu fitur dari ssh yang memungkinkan kita untuk upload ataupun download file/direktori pada server dengan menggunakan aplikasi GUI. Salah satu aplikasi yang bisa kita gunakan pada client adalah filezilla. Kita bisa menginstallnya pada ubuntu dengan perintah seperti ini.

```
admin@ubuntu:~$ sudo apt-get install filezilla
[sudo] password for admin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
.....
.....
```

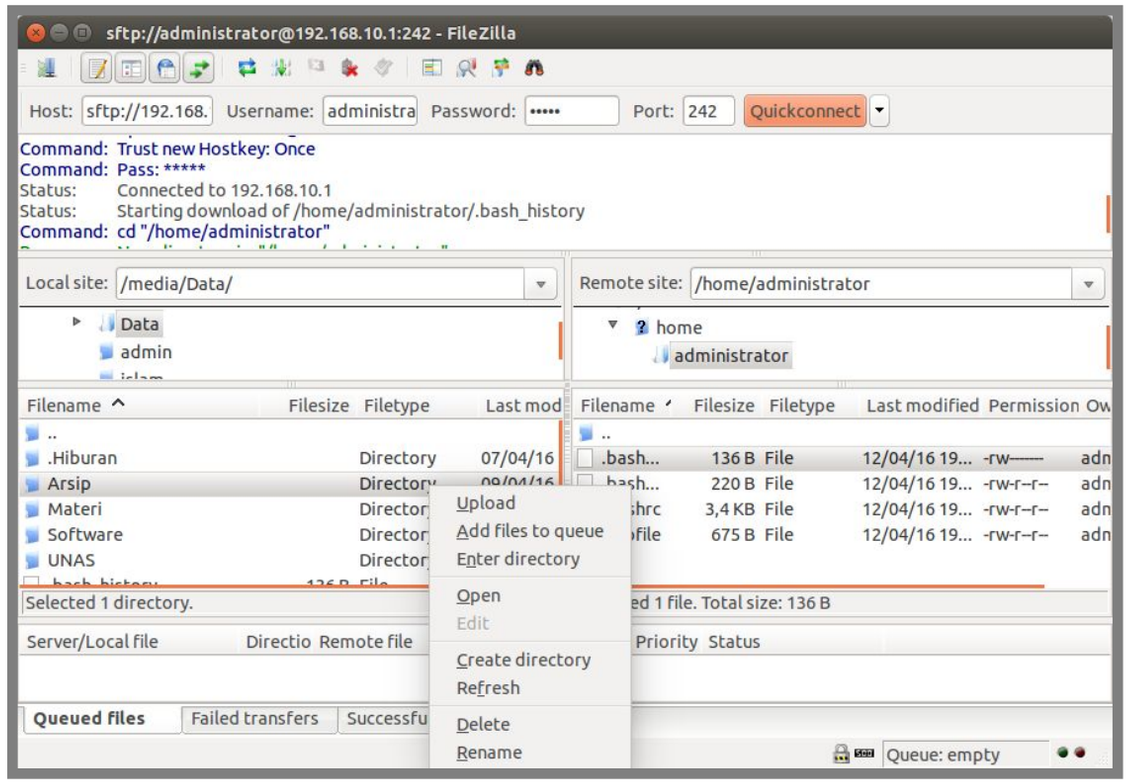
Gambar 5.26 Installasi filezilla di client

Namun jika menggunakan sistem operasi windows sebagai client, kita bisa download apikasi filezilla dari internet kemudian menginstallnya seperti biasa. Berikut langkah yang perlu dilakukan untuk melakukan file transfer dengan sftp



Gambar 5.27 Login sftp menggunakan filezilla

Pada kolom host, isikan *sftp://192.168.10.1* (192.168.10.1 ialah ip address server), Isikan pada kolom username, password, dan password sesuai dengan yang telah dikonfigurasikan pada ssh. Selanjutnya klik pada *quickconnect*, berikut tampilan saat kita telah berhasil connect



Gambar 5.28 Halaman utama sftp

Perhatikan gambar diatas, terlihat bahwa terdapat beberapa menu yang bisa kita gunakan untuk transfer file,

SSH dengan RSA Key Authentication

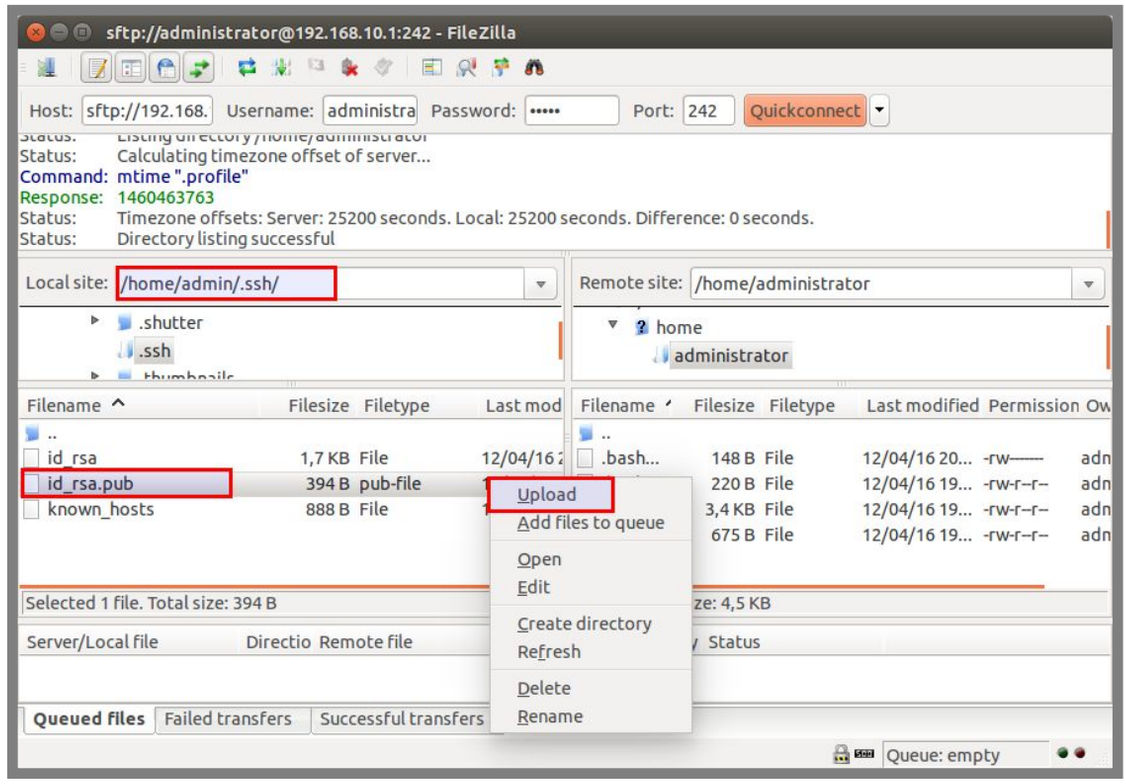
Dengan menggunakan metode authentikasi ini, kita bisa meremote server tanpa harus memasukkan password. Sistem authentikasi yang digunakan pada metode ini adalah, antara komputer client dan server mempunyai pasangan public key dan private key (semacam kunci dan gembok). Dimana client harus mempunyai kunci (private key) yang cocok dengan gembok (public key) pada server agar bisa melakukan remote access.

Tentu saja metode ini sangat aman, karena tidak mungkin orang lain bisa mempunyai private key yang cocok dengan public key pada server. Selain itu public key dan private key yang digunakan pada metode ini menggunakan enkripsi tingkat tinggi, jadi sangat sulit untuk memecahkannya. Berikut langkah yang perlu dilakukan

```
admin@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/admin/.ssh/id_rsa): (enter)
Enter passphrase (empty for no passphrase): (enter)
Enter same passphrase again: (enter)
Your identification has been saved in /home/admin/.ssh/id_rsa.
Your public key has been saved in /home/admin/.ssh/id_rsa.pub.
The key fingerprint is:
61:77:b0:3e:c0:28:40:75:80:67:10:9a:0b:78:ff:56 admin@ubuntu
The key's randomart image is:
+----[ RSA 2048]----+
| .==o..    .       |
|.o..o. o    o      |
|= .o. . = o .      |
|.o . . . = .       |
|.    .    E o      |
|       . .   .     |
|        o          |
|       .           |
+-------------------+
admin@ubuntu:~$ ls /home/admin/.ssh/
id_rsa  id_rsa.pub  known_hosts
admin@ubuntu:~$
```

Gambar 5.29 Membuat pasangan private key dan public key

Perintah pertama yang digunakan pada gambar diatas (*ssh-keygen*) otomatis akan membuat file *id_rsa* dan *id_rsa.pub* pada home direktori user yang digunakan, seperti yang ditunjukkan pada perintah kedua. Selanjutnya, upload file *id_rsa.pub* ke komputer server menggunakan sftp yang telah kita pelajari di sub bab sebelumnya



Gambar 5.30 Upload public key ke server

Langkah berikutnya, gunakan perintah seperti berikut pada komputer server

```
administrator@forkits:~$ pwd
/home/administrator
administrator@forkits:~$ mkdir .ssh
administrator@forkits:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .profile  .ssh  id_rsa.pub
administrator@forkits:~$ cp id_rsa.pub .ssh/authorized_keys
administrator@forkits:~$ ls .ssh/
authorized_keys
administrator@forkits:~$
```

Gambar 5.31 Konfigurasi public key pada server

Karena user yang ingin kita gunakan saat meremote server adalah administrator, maka kita mengupload file id_rsa.pub, membuat direktori .ssh, dan semua langkah diatas di home direktori administrator (/home/administrator). Namun jika kita ingin menggunakan user root untuk meremote server, kita harus melakukan langkah-langkah diatas di home direktori root (/root). Selanjutnya rubah konfigurasi ssh seperti berikut

```
root@forkits:~# nano /etc/ssh/sshd_config
.....
.....
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no
.....
.....
root@forkits:~# service ssh restart
[ ok ] Restarting OpenBSD Secure Shell server: sshd.
root@forkits:~#
```

Gambar 5.32 Konfigurasi ssh pada server

Cari, dan lakukan sedikit perubahan pada file konfigurasi ssh, tepatnya pada teks berwarna hijau. Untuk mempermudah pencarian, kita bisa menggunakan fasilitas search di nano dengan menekan tombol kombinasi *ctrl+w* kemudian ketikkan kata kuncinya *PasswordAuthentication.* Setelah itu jangan lupa untuk merestart service ssh. Berikut pengujian yang dilakukan dari komputer client

```
admin@ubuntu:~$ ssh 192.168.10.1 -l administrator -p 242
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 13 05:20:43 2016 from 192.168.10.2
administrator@forkits:~$
```

Gambar 5.33 Pengujian ssh dari client

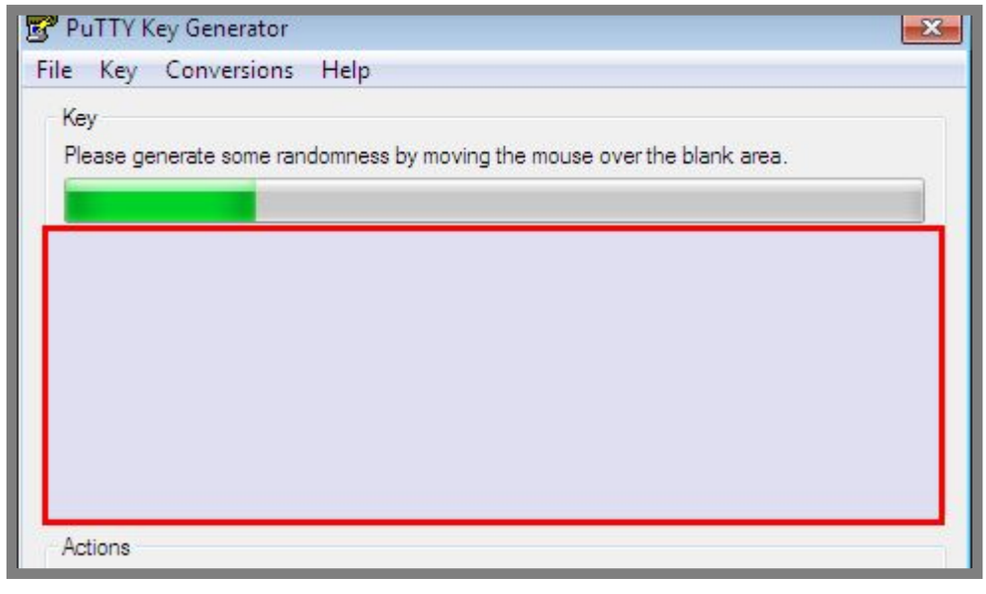
Perhatikan gambar diatas, terlihat bahwa kita tidak diminta untuk memasukkan password. Karena proses authentikasi telah dilakukan dengan menggunakan pasangan file private key dan public key yang telah dimiliki oleh client dan server.

Pembahasan diatas, kita hanya fokus menggunakan linux ubuntu sebagai sistem operasi client. Karena terdapat perbedaan langkah konfigurasi yang sangat besar dengan sistem operasi windows client, pada sub bab ini juga akan dibahas remote access ssh dengan rsa key authentication menggunakan sistem operasi windows sebagai clientnya.

Software yang diperlukan adalah puttygen (untuk membuat pasangan private key dan public key) dan putty (untuk melakukan remote access). Silahkan download kedua aplikasi yang dibutuhkan tersebut di internet. Selanjutnya buka puttygen dan lakukan langkah seperti berikut

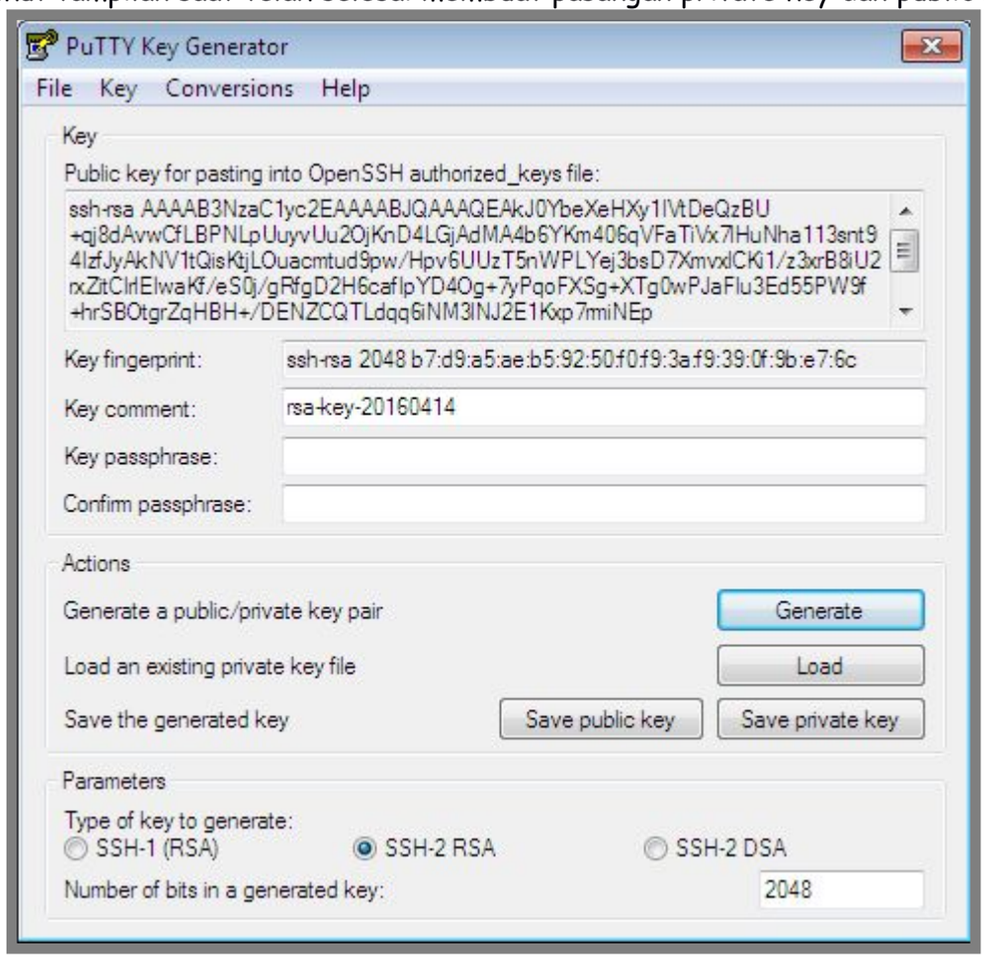
Gambar 5.34 Membuat pasangan private key dan public key dengan puttygen

Saat proses pembuatan private key dan public key, gerak-gerakkan kursor pada bagian yang diberi tanda kotak merah



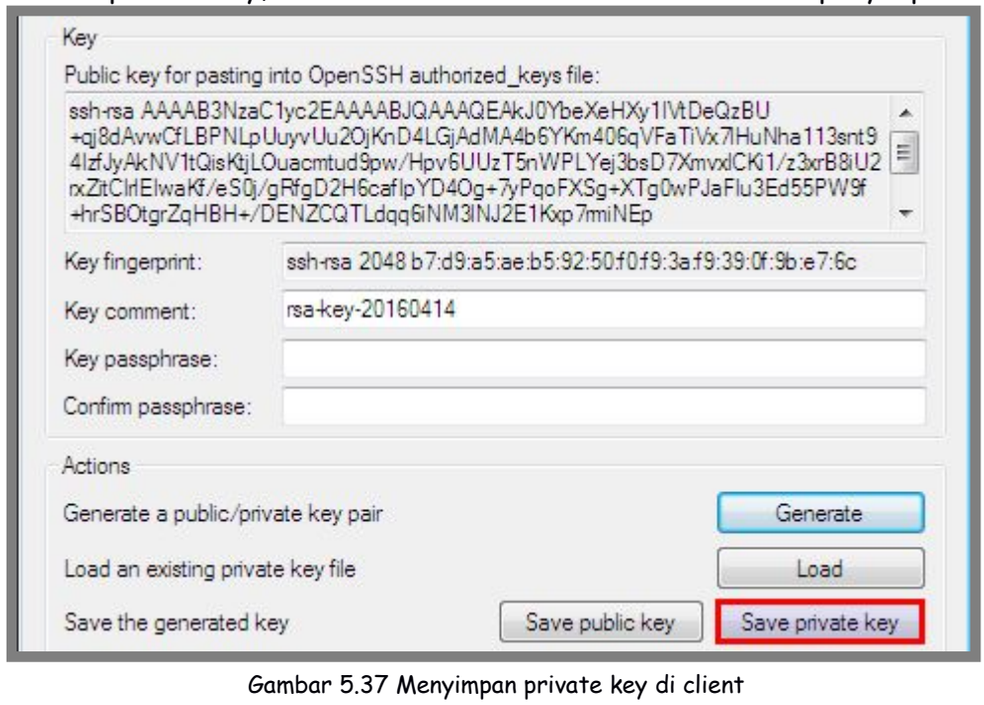
Gambar 5.35 Proses pembuatan private key dan public key

Berikut tampilan saat telah selesai membuat pasangan private key dan public key



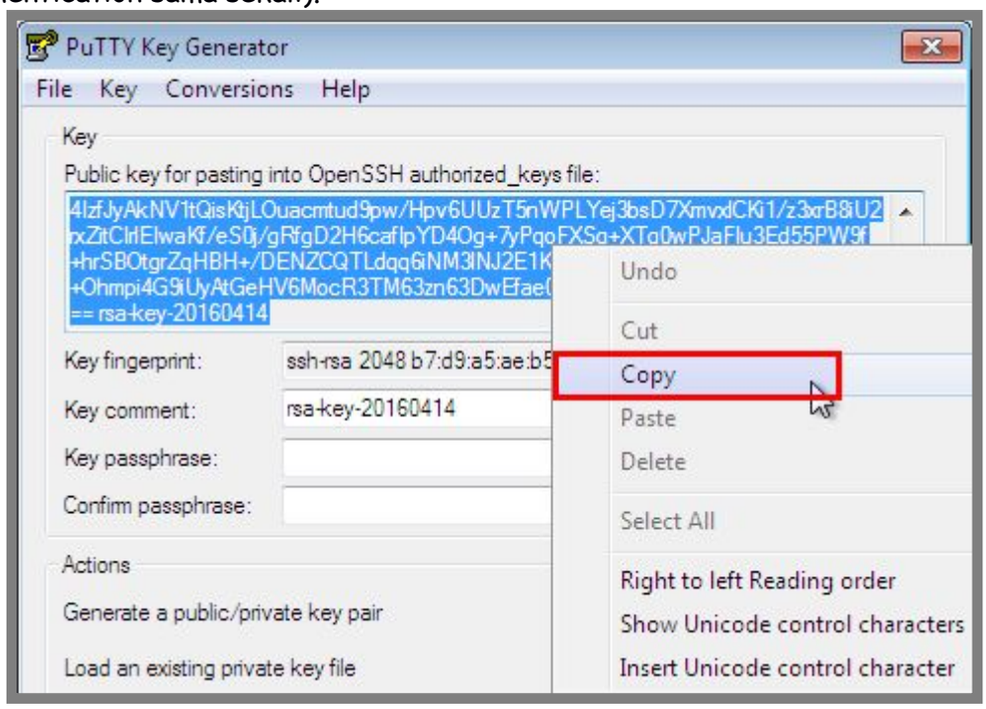
Gambar 5.36 Hasil setelah selesai membuat private key dan public key

Sampai saat ini, kita telah selesai membuat pasangan private key dan public key. Langkah berikutnya, kita harus menyimpan file private key di client. Klik pada bagian save private key, kemudian beri nama dan tentukan lokasi penyimpanan



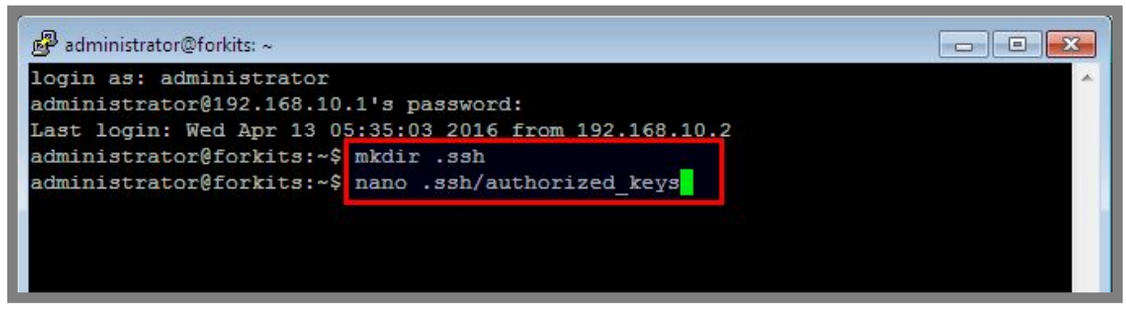
Gambar 5.37 Menyimpan private key di client

Setelah menyimpan file private key, langkah selanjutnya adalah memasukkan public key yang sudah digenerate tadi ke komputer server. Kita asumsikan bahwa komputer server masih bisa diremote seperti biasa (belum dikonfigurasi rsa key authentication sama sekali).



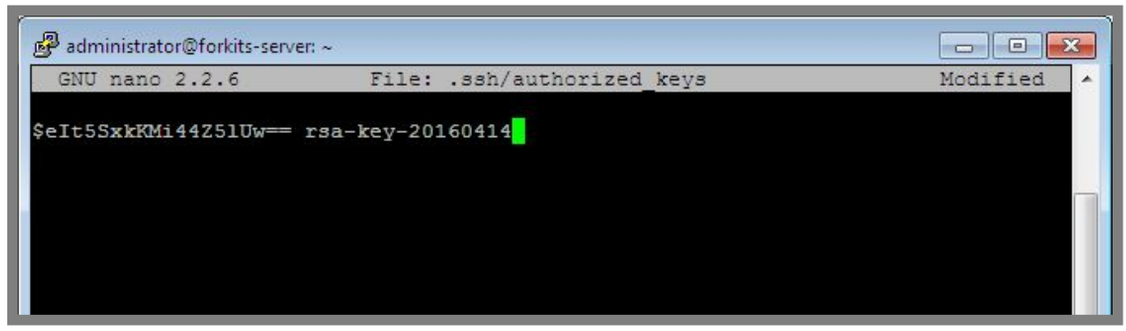
Gambar 5.37 Copy public key dari client

Remote server menggunakan putty, buat direktori .ssh di home direktori user, dan kemudian buat file authorized_keys didalamnya



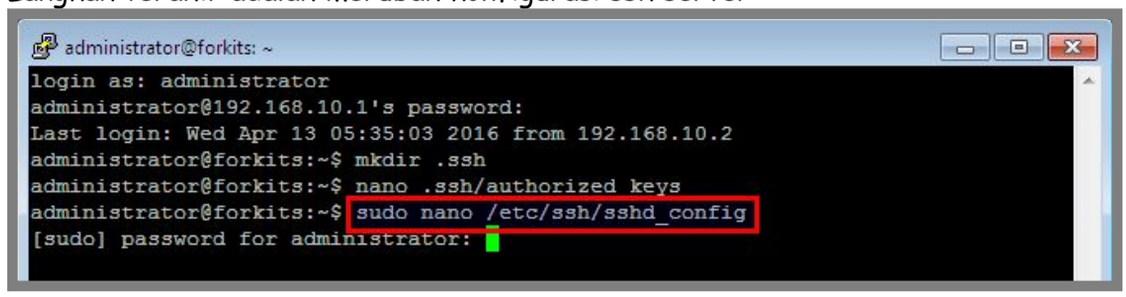
Gamba 5.38 Membuat konfigurasi public key di client

Pastekan file public key yang telah kita copy sebelumnya ke dalam file authorized_keys tersebut dengan cara klik kanan. Pastikan file tersebut tertulis dalam satu baris



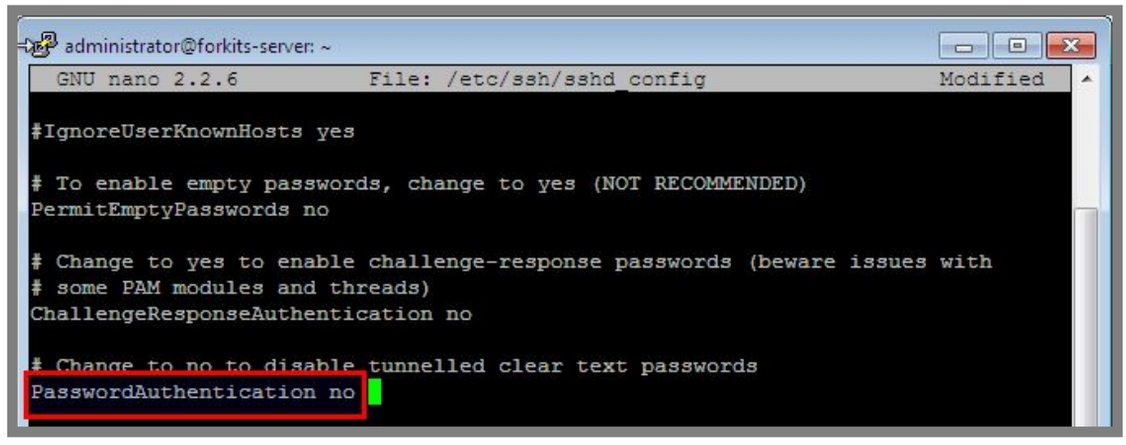
Gambar 5.39 Paste public key ke server

Langkah terahir adalah merubah konfigurasi ssh server



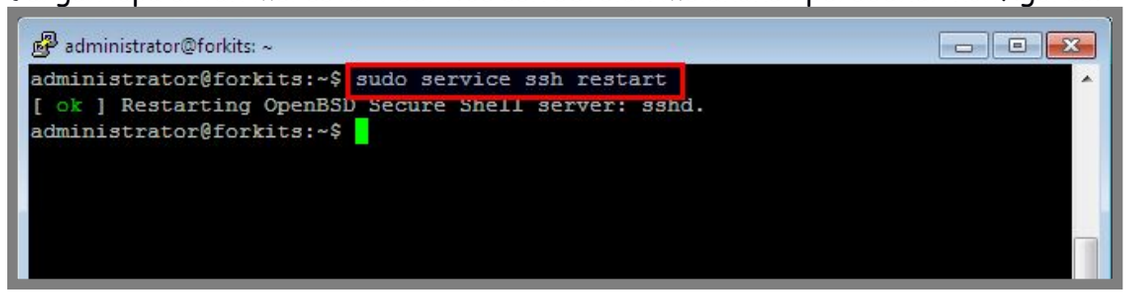
Gambar 5.40 Merubah konfigurasi ssh

Rubah pada bagian yang bertanda kotak warna merah



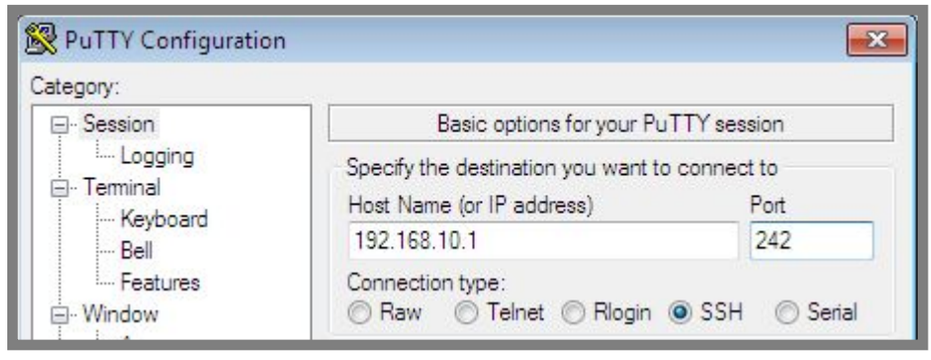
Gambar 5.41 Merbuah konfigurasi ssh

Jangan lupa untuk merestart service ssh setelah melakukan perubahan konfigurasi



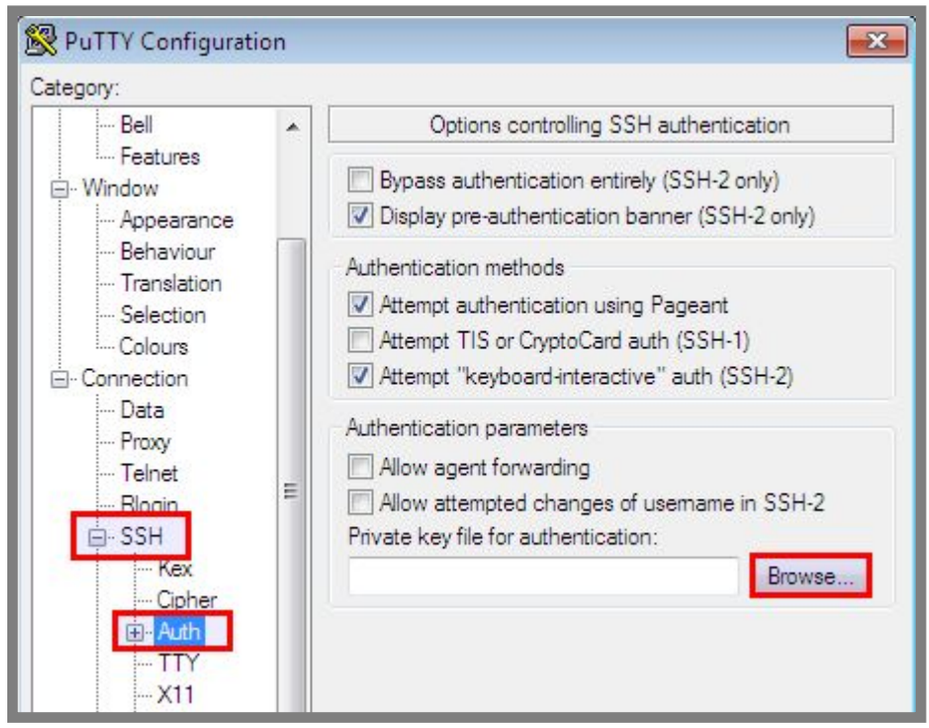
Gambar 5.42 Merestart service ssh

Sampai saat ini kita sudah selesai konfigurasi ssh dengan rsa key authentication di komputer server. Selanjutnya lakukan pengujian di komputer client, buka jendela baru aplikasi putty kemudian ikuti langkah berikut

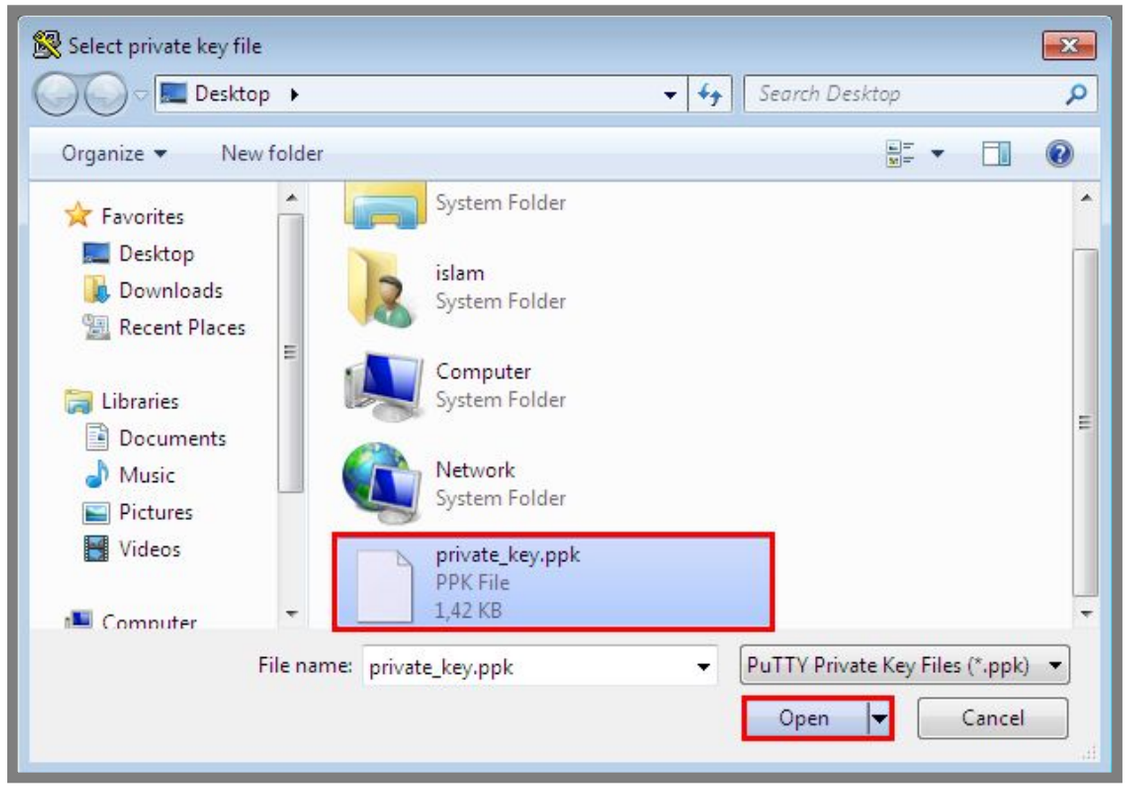


Gambar 5.43 Melakukan pengujian dari client

Setelah memasukkan ip address server dan port yang digunakan, buka pada bagian *ssh -> auth*, kemudian klik *browse* dan arahkan ke file private key yang telah kita simpan tadi

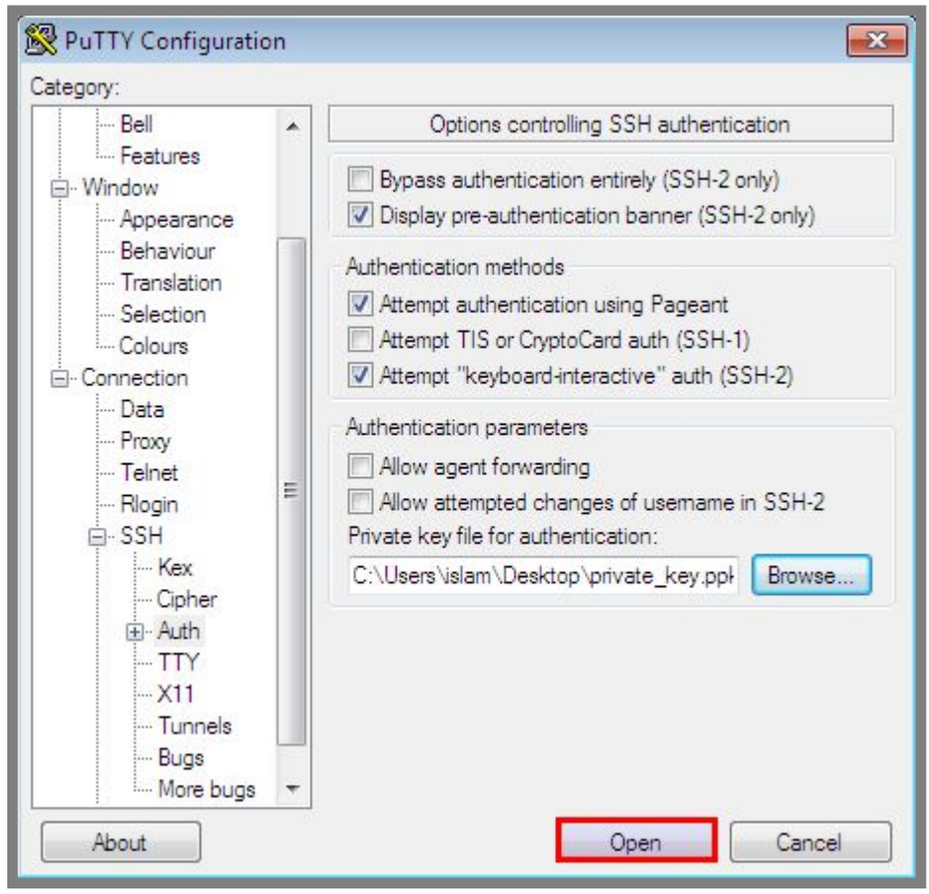


Gambar 5.44 Konfigurasi private key di putty

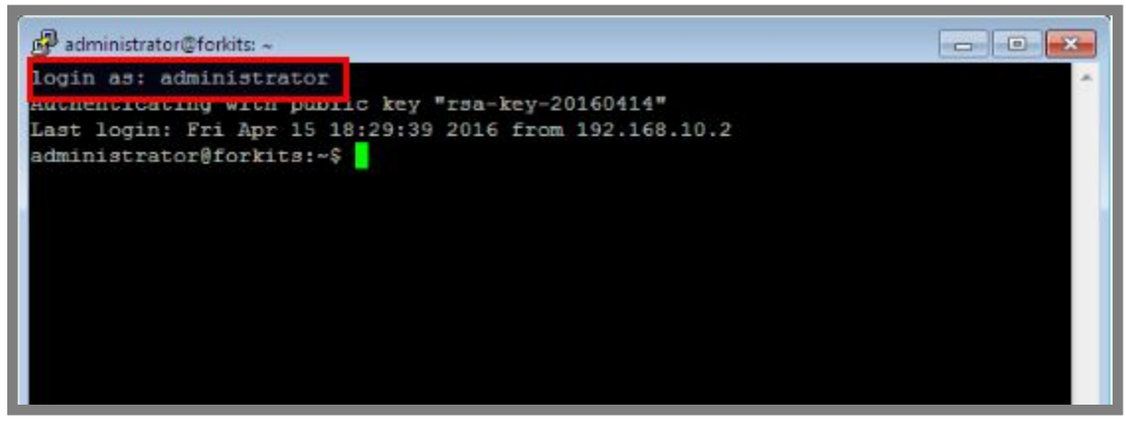


Gambar 5.45 Mencari lokasi private key

Selanjutnya, klik *open* pada putty dan perhatikan hasilnya



Gambar 5.46 Mencoba melakukan ssh ke server dengan private key



Gambar 5.47 Hasil pengujian ssh dari client menggunakan private key

Perhatikan gambar diatas, terlihat bahwa kita telah berhasil login ke server tanpa menggunakan password. Hal ini karene authentikasi telah dilakukan menggunakan pasangan file private key dan public key yang telah kita generate.

Merubah Welcome Message Linux

Jika kita perhatikan, setiap kita login ke komputer server, entah itu melalui telnet, ssh, ataupun dari komputer langsung, selalu muncul tampilan seperti ini (teks warna merah)

```
admin@ubuntu:~$ ssh 192.168.10.1 -l administrator -p 242
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
administrator@forkits:~$
```

Gambar 5.48 Welcome message default di linux

Jika kita bosan dengan tampilan tersebut, kita bisa menggantinya sesuai keinginan kita dengan melakukan langkah berikut

```
root@forkits:~# nano /etc/motd
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
                              WELCOME TO
                          SECURE FOR KITS SERVER

---------------------NO DREAM IN THE SIMPLE WAY--------------------------
--------------------------------------------------------------------------------
```

Gambar 5.49 Merubah welcome message di linux

Kita bebas mengisi file tersebut dengan tulisan yang kita inginan. Berikut pengujian yang dilakukan setelah memodifikasi file tersebut

```
admin@ubuntu:~$ ssh 192.168.10.1 -l administrator -p 242
Last login: Thu Apr 14 11:03:47 2016 from 192.168.10.2
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
                              WELCOME TO
                          SECURE FOR KITS SERVER

---------------------NO DREAM IN THE SIMPLE WAY--------------------------
--------------------------------------------------------------------------------
administrator@forkits:~$
```

Gambar 5.50 Pengujian perubahan welcome message yang telah dilakukan

---END OF CHAPTER---

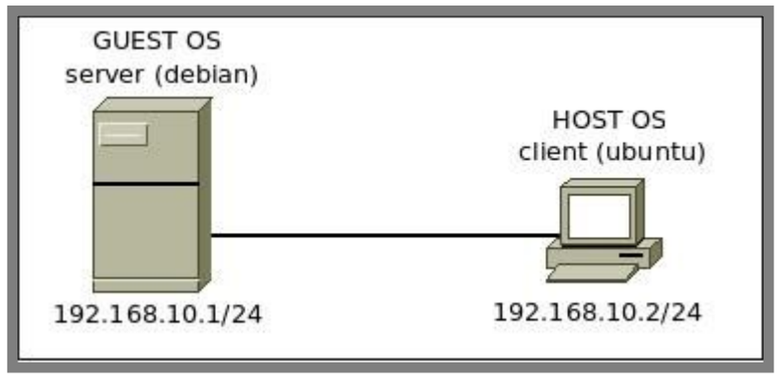
Bab 6
Domain Name System Server

Domain Name System (DNS) adalah sebuah protocol jaringan komputer yang bertugas untuk malekukan konversi ip address suatu komputer menjadi domain, atau sebaliknya.

Sebagai contoh, saat membuka sebuah situs di internet, kita tidak pernah membuka situs tersebut menggunakan ip address, melainkan langsung menggunakan domain name seperti www.google.com, www.youtube.com, dll. Hal ini dikarenakan manusia cenderung lebih mudah mengingat nama domaian (alfabet) daripada mengingat ip address (numerik).

Konfigurasi Primary DNS Server

Sebagai contoh kasus, kita diminta untuk membuat sebuah dns server dengan domain forkits.com dengan topologi jaringan sebagai berikut



Gambar 6.1 Topologi jaringa untuk praktik DNS Server

Diasumsikan bahwa komputer server dan client telah dikonfigurasi ip address sesuai dengan topologi diatas dan telah bisa saling berkomunikasi.

Untuk membuat sebuah dns server, kita memerlukan sebuah aplikasi yang bisa melakukan pekerjaan tersebut. Salah satu aplikasi yang bisa kita gunakan adalah bind9. Aplikasi ini terkenal dengan beberapa keunggulannya, terutama kemudahannya dalam melakukan konfigurasi. Berikut cara yang bisa digunakan untuk install aplikasi bind9.

```
root@forkits:~# apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bind9utils
Suggested packages:
  bind9-doc resolvconf ufw
The following NEW packages will be installed:
  bind9 bind9utils
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/490 kB of archives.
After this operation, 1257 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 6.2 Installasii bind9 untuk dns server

Setelah bind9 terinstall, selanjutnya kita harus melakukan konfigurasi. Konfigurasi pertama yang harus dilakukan adalah membuat domain zone. Telah disepakati sebelumnya bahwa domain yang akan kita buat adalah forkits.com.

```
root@forkits:~# cd /etc/bind
root@forkits:/etc/bind# nano named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

zone "forkits.com" {
     type master;
     file "/etc/bind/db.forkits";
};
zone "192.in-addr.arpa" {
     type master;
     file "/etc/bind/db.192";
};

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Gambar 6.3 Konfigurasi domain zone

Pada langkah diatas, kita hanya menambahkan baris dengan teks warna hijau. Konfigurasi bind9 mempunyai tingkat sensitifitas yang tinggi, artinya kurang satu titik atau kelebihan satu titik saja akan menyebabkan bind9 error, maka dari itu harus dipastikan bahwa kita tidak melakukan kesalahan dalam menulis syntak. Berikut keterangan dari masing-masing baris yang ditambahkan diatas

Syntak	Deskripsi
zone "forkits.com" {	Baris ini menunjukkan bahwa kita membuat sebuah domain dengan nama *forkits.com*
type master;	Baris ini menunjukkan bahwa domain yang kita buat (forkits.com) adalah primary dns server (dns utama). Selain primary dns, nantinya kita juga akan belajar tentang secondary dns server.
file "/etc/bind/db.forkits";	Menunjukkan lokasi penyimpanan file konfigurasi domain forward. Fungsi file forward adalah menerjemahkan domain name menjadi ip address.
};	Merupakan penutup/ahir dari pendeskripsian domain forward.
zone "192.in-addr.arpa" {	Baris ini mendeskripsikan domain reverse yang kita buat. Domain reverse bertugas untuk menerjemahkan ip address menjadi domain name (kebalikan dari domain forward).
File "/etc/bind/db.192";	Baris ini mendeskripsikan lokasi penyimpanan file konfigurasi domain reverse.

Selanjutnya, kita harus membuat file konfigurasi domain forward dan mengkonfigurasinya

```
root@forkits:/etc/bind# cp db.local db.forkits
root@forkits:/etc/bind# nano db.forkits
; BIND data file for local loopback interface
$TTL    604800
@       IN      SOA     forkits.forkits.com. admin.forkits.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
@       IN      NS              forkits.forkits.com.
@       IN      A               192.168.10.1
forkits IN      A               192.168.10.1
www     IN      A               192.168.10.1
mail    IN      A               192.168.10.1
web     IN      CNAME           www
ftp     IN      A               192.168.10.1
@       IN      MX      1       mail.forkits.com.
```

Gambar 6.4 Konfigurasi file forward

Berikut keterangan dari masing-masing syntax diatasMisalnya s

Syntak	Deskripsi
forkits.forkits.com.	Mendeklarasikan hostname yang menjadi dns server
admin.forkits.com.	Mendeklarasikan email administrator dns server. Namun penulisa '@' diganti dengan titik (.), seperti penulisan tersebut menandakan email adminstrator adalah *admin@forkits.com*
2 ; Serial	Menunjukkan nomor serial yang dimiliki dns server
604800 ; Refresh (satu minggu)	Mendeklarasikan selang waktu (dalam detik) yang digunakan oleh secondary dns server untuk melakukan pengecekan terhadap perubahan file zone primary dns server. Jika ada perubahan, maka secondary dns server juga akan menyesuaikan perubahan tersebut.
86400 ; Retry (satu hari)	Mendeklarasikan berapa lama (dalam detik) secondary dns server menunggu untuk mengulangi pengecekan terhadap file zona primary dns server jika primary dns server tidak merespon saat proses refresh.
2419200 ; Expire (satu bulan)	Mendeklarasikan berapa lama (dalam detik) file zona dipertahankan oleh secondary dns server jika primary dns server tetap tidak melakukan respon saat proses refresh. Apabila setelah masa expire primary dns server tetap tidak merespon, maka secondary dns server akan menghapus file zona miliknya.
IN	Singkatan dari Internet Name, ini digunakan saat kita menggunakan protocal tcp/ip
NS	Mendeklarasikan hostname komputer yang akan menjawab atau melayani request informasi seputar domain name system.
A	A record digunakan untuk membuat sebuah sub domain, sekaligus bertugas mapping dari domain ke ip address
CNAME	Digunakan untuk mendeklarasikan domain alias. Dari contoh diatas terlihat bahwa web merupakan alias dari www
MX	Menentukan domain/hostname mana yang akan menjadi mail server. MX diikuti dengan priority, terlihat pada contoh diatas prioritynya adalah 1. Priority yang bisa dipakai adalah 0-65536, semakin kecil nilai prioritynya, maka domain tersebut akan lebih diprioritaskan menjadi mail server.

Selanjutnya buat dan konfigurasi file reverse dengan cara berikut

```
root@forkits:/etc/bind# cp db.127 db.192
root@forkits:/etc/bind# nano db.192
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     forkits.forkits.com. admin.forkits.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@               IN      NS      forkits.forkits.com.
1.10.168        IN      PTR     forkits.forkits.com.
```

Gambar 6.5 Konfigurasi file reverse

Seluruh isi file reverse mempunyai arti yang sama dengan file forward yang telah dijelaskan sebelumnya. Namun ada satu istilah yang belum dijelaskan, yaitu PTR. PTR adalah kebalikan dari A record, jika A record bertugas mapping domain name menjadi ip address, maka PTR bertugas mapping ip address menjadi domain name.

Perhatikan baris terahir pada gambar diatas, (1.10.168) adalah kebalikan tiga oktet terahir dari ip address server. Yang dimasukkan di konfigurasi reverse ini hanya tiga oktet terahir karena oktet pertama (192) sudah dimasukkan ke dalam konfigurasi zone sebelumnya (pada file */etc/bind/named.conf*).

Selanjutnya restart service bind9

```
root@forkits:/etc/bind# service bind9 restart
[....] Stopping domain name service...: bind9waiting for pid 3135 to die
. ok
[ ok ] Starting domain name service...: bind9.
root@forkits:/etc/bind#
```

Gambar 6.6 Proses restart dns server

Sebelum melakukan pengujian, pastikan bahwa resolver sudah diarahka ke ip address dns server

```
root@forkits:/etc/bind# cat /etc/resolv.conf
nameserver 192.168.10.1
root@forkits:/etc/bind#
```

Gambar 6.7 Konfigurasi dns resolver

Berikut hasil pengujian yang dilakukan dari komputer server

```
root@forkits:/etc/bind# nslookup forkits.com
Server:     192.168.10.1
Address:    192.168.10.1#53

Name:   forkits.com
Address: 192.168.10.1

root@forkits:/etc/bind# nslookup 192.168.10.1
Server:     192.168.10.1
Address:    192.168.10.1#53

1.10.168.192.in-addr.arpa     name = forkits.forkits.com.

root@forkits:/etc/bind#
```

Gambar 6.8 Pengujian dns server dengan *nslookup*

Berikut hasil pengujian yang dilakukan dari komputer client. Sebelumnya pastikan bahwa client sudah bisa berkomunikasi dengan server, selain itu pastikan juga bahwa resolver client telah diarahkan ke server.forkits.com

```
admin@ubuntu:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.425 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.364 ms
--- 192.168.10.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.364/0.394/0.425/0.036 ms
admin@ubuntu:~$ cat /etc/resolv.conf
nameserver 192.168.10.1
admin@ubuntu:~$ nslookup forkits.com
Server:     192.168.10.1
Address:    192.168.10.1#53

Name:   forkits.com
Address: 192.168.10.1

admin@ubuntu:~$ nslookup 192.168.10.1
Server:     192.168.10.1
Address:    192.168.10.1#53

1.10.168.192.in-addr.arpa     name = forkits.forkits.com.

admin@ubuntu:~$
```

Gambar 6.9 Pengujian dns server dari client

Membuat Virtual Domain

Sebelumnya telah dibahas tentang cara membuat sebuah domain. Pada sub bab ini, kita akan mempelajari tentang cara membuat virtual domain. Maksud dari virtual domain itu sendiri adalah kita bisa mempunyai lebih dari satu domain dalam sebuah dns server.

Kita telah membuat domain dengan nama forkits.com pada sub bab sebelumnya. Di sub bab ini, kita akan menambahkan sebuah domain dengan nama debian.id. Diasumsikan bahwa kita telah melakukan konfigurasi dns server pada sub bab sebelumnya, jadi pada sub bab ini kita hanya akan menambahkan beberapa konfigurasi saja.

Berikut konfigurasi file zone

```
root@forkits:~# cd /etc/bind
root@forkits:/etc/bind# nano named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in

zone "forkits.com" {
     type master;
     file "/etc/bind/db.forkits";
};
zone "debian.id" {
     type master;
     file "/etc/bind/db.debian";
};
zone "192.in-addr.arpa" {
     type master;
     file "/etc/bind/db.192";
};

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Gambar 6.10 Konfigurasi domain zone

Pada gambar diatas, terlihat bahwa kita hanya menambahkan beberapa baris konfigurasi (teks warna hijau) dari konfigurasi pada sub bab sebelumnya.

Selanjutnya buat dan konfigurasi file forward dari domain debian.id tersebut.

```
root@forkits:/etc/bind# cp db.local db.debian
root@forkits:/etc/bind# nano db.debian
; BIND data file for local loopback interface
$TTL    604800
@       IN      SOA     forkits.debian.id. admin.debian.id. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@           IN      NS      forkits.debian.id.
@           IN      A       192.168.10.1
forkits     IN      A       192.168.10.1
www         IN      A       192.168.10.1
```

Gambar 6.11 Konfigurasi file forward untuk debian.id

Sedangkan untuk file reverse, kita tidak perlu membuat lagi, melainkan cukup menambahkan konfigurasi dari file reverse yang dibuat pada sub bab sebelumnya

```
root@forkits:/etc/bind# nano db.192
; BIND reverse data file for local loopback interface
$TTL    604800
@       IN      SOA     forkits.forkits.com. admin.forkits.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@           IN      NS      forkits.forkits.com.
1.10.168    IN      PTR     forkits.com.
1.10.168    IN      PTR     debian.id.
```

Gambar 6.12 Konfigurasi file reverse untuk debian.id

Sebelum melakukan pengujian, jangan lupa untuk merestart service bind9

```
root@forkits:/etc/bind# service bind9 restart
[....] Stopping domain name service...: bind9waiting for pid 2042 to die
. ok
[ ok ] Starting domain name service...: bind9.
root@forkits:/etc/bind#
```

Gambar 6.13 Prses restart service dns server

Berikut pengujian yang dilakukan dari komputer server. Jangan lupa untuk mengkonfigurasi dns resolver agar menggunaakan ip address server

```
root@forkits:/etc/bind# cat /etc/resolv.conf
nameserver 192.168.10.1
root@forkits:/etc/bind# nslookup forkits.com
Server:     192.168.10.1
Address:    192.168.10.1#53

Name:   forkits.com
Address: 192.168.10.1

root@forkits:/etc/bind# nslookup debian.id
Server:     192.168.10.1
Address:    192.168.10.1#53

Name:   debian.id
Address: 192.168.10.1

root@forkits:/etc/bind#
```

Gambar 6.14 Pengujian dari komputer server

Berikut pengujian yang dilakukan dari komputer client

```
admin@ubuntu:~$ cat /etc/resolv.conf
nameserver 192.168.10.1
admin@ubuntu:~$ nslookup forkits.com
Server:     192.168.10.1
Address:    192.168.10.1#53

Name:   forkits.com
Address: 192.168.10.1

admin@ubuntu:~$ nslookup debian.id
Server:     192.168.10.1
Address:    192.168.10.1#53

Name:   debian.id
Address: 192.168.10.1

admin@ubuntu:~$
```

Gambar 6.15 Pengujian dari komputer client

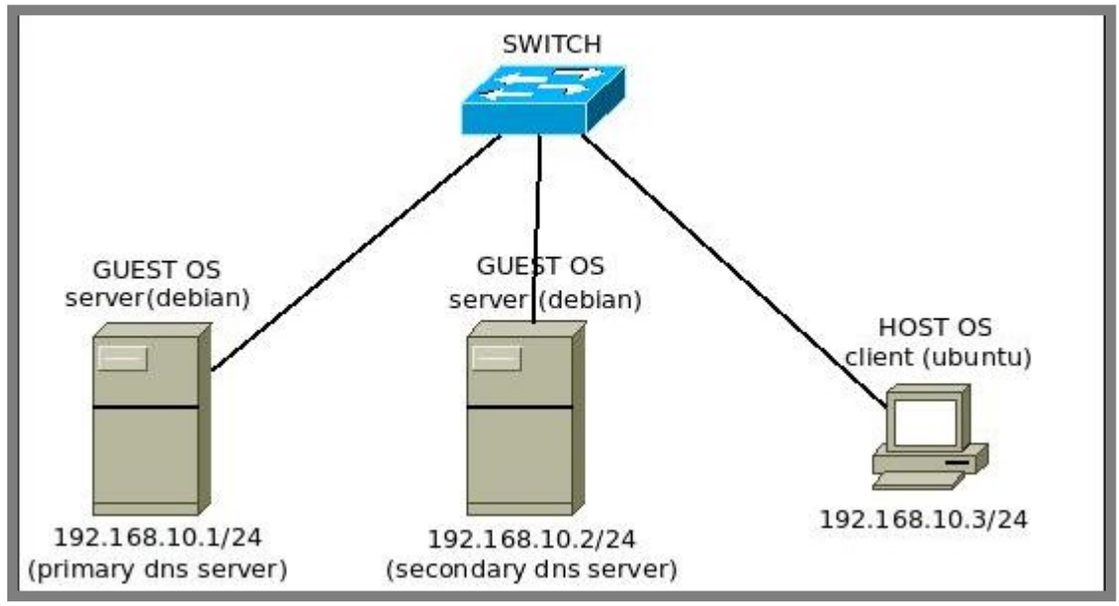
Konfigurasi Primary & Secondary DNS Server

Sebelumnya kita telah membuat primary dns server. Jika suatu saat primary dns server kita mengalami gangguan, entah itu gangguan pada komputer ataupun jaringan, maka seluruh client yang menggunakan dns server kita akan bermasalah. Disaat seperti itulah secondary dns server dibutuhkan untuk menjadi backup. Dengan kata lain, secondary dns server mempunyai tugas untuk menjadi backup jika suatu saat primary dns server tidak bisa menjalankan tugasnya dengan baik.

Pada dunia nyata (penerapan kerja), sangat disarankan untuk meletakkan secondary dns server di daerah yang berbeda dengan primary dns server. Hal ini dimaksudkan untuk menghindari terjadinya masalah yang sama antara primary dns server dan secondary dns server.

Misalkan primary dns server dan secondary dns server diletakkan di daerah geografis yang sama, kemudian didaerah tersebut terjadi gangguan listrik, maka kedua komputer tidak akan berjalan normal. Jika hal semacam ini terjadi, maka adanya secondary dns server tidak akan ada gunanya sama sekali. Namun jika hanya untuk pembelajaran, hal seperti ini bukanlah menjadi suatu masalah serius.

Berikut topologi yang akan kita praktikkan



Gambar 6.16 Topologi jaringan untuk praktik secondary dns server

Untuk mewujudkan topologi diatas di virtualbox, gunakan host only adapter pada kedua guest os (telah dibahas di bab 4). Switch ditopolgi diatas hanya sebuah switch virtual, kita tidak membutuhkan switch!

Diasumsikan bahwa ketiga komputer pada topologi diatas telah dikonfigurasi ip address sesuai topologi dan sudah bisa saling berkomunikasi. Selain itu, diasumsikan juga bahwa di komputer server (baik itu primary ataupun secondary dns server) telah diinstall aplikasi bind9.

Berikut konfigurasi hostname di masing-masing server

```
root@forkits:/etc/bind# hostname -f
forkits.forkits.com
root@forkits:/etc/bind#
```

Gambar 6.17 Konfigurasi hostname di primary dns server

```
root@forkits2:/etc/bind# hostname -f
forkits2.forkits.com
root@forkits2:/etc/bind#
```

Gambar 6.18 Konfigurasi hostname di secondary dns server

Langkah-langkah untuk konfigurasi hostname seperti diatas telah dibahas di bab 4. Kita lanjutkan dengan konfigurasi zone di primary dns server

```
root@forkits2:/etc/bind# nano named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

zone "forkits.com" {
     type master;
     file "/etc/bind/db.forkits";
     allow-transfer { 192.168.10.2; };
};
zone "192.in-addr.arpa" {
     type master;
     file "/etc/bind/db.192";
     allow-transfer { 192.168.10.2; };
};

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Gambar 6.19 Konfigurasi domain zone pada primary dns server

Perhatikan gambar diatas, terlihat bahwa kita hanya menambahkan baris warna hijau pada konfigurasi yang telah dibahas di sub bab konfigurasi primary dns server sebelumnya. Selanjutnya konfigurasi file forward seperti ini

```
root@forkits2:/etc/bind# nano db.forkits
; BIND data file for local loopback interface
$TTL    604800
@       IN      SOA     forkits.forkits.com. admin.forkits.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@           IN      NS          forkits.forkits.com.
@           IN      NS          forkits2.forkits.com.
@           IN      A           192.168.10.1
forkits     IN      A           192.168.10.1
forkits2    IN      A           192.168.10.2
www         IN      A           192.168.10.1
mail        IN      A           192.168.10.1
web         IN      CNAME       www
@           IN      MX      1   mail.forkits.com.
```

Gambar 6.20 Konfigurasi file forward pada primary dns server

Perhatikan gambar diatas, terlihat bahwa saat ini kita mempunya dua NS, yaitu primary dns server (forkits.forkits.com) dan secondary dns server (forkits2.forkits.com). Perhatikan pula bahwa ip address primary dns server (forkits) adalah 192.168.10.1, sedangkan ip address secondary dns server (forkits2) adalah 192.168.10.2

Berikutnya konfigurasi file reverse pada primary dns server seperti ini

```
root@forkits:/etc/bind# nano db.192
; BIND reverse data file for local loopback interface
$TTL    604800
@       IN      SOA     forkits.forkits.com. admin.forkits.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@           IN      NS          forkits.forkits.com.
@           IN      NS          forkits2.forkits.com.
1.10.168    IN      PTR         forkits.forkits.com.
2.10.168    IN      PTR         forkits2.forkits.com.
```

Gambar 6.21 Konfigurasi file reverse pada primary dns server

Terahir, restart service bind9 di primary dns server

```
root@forkits:/etc/bind# service bind9 restart
[....] Stopping domain name service...: bind9waiting for pid 3135 to die
. ok
[ ok ] Starting domain name service...: bind9.
root@forkits:/etc/bind#
```

Gambar 6.22 Proses restart service dns server di primary dns server

Sampai saat ini kita sudah selesai melakukan konfigurasi primary dns server, selanjutnya kita akan mengkonfigurasi secondary dns server. Berikut konfigurasi zone pada secondary dns server

```
root@forkits2:/etc/bind# nano named.conf
.....
.....
.....

zone "forkits.com" {
      type slave;
      masters { 192.168.10.1; };
      file "db.forkits";
};
zone "192.in-addr.arpa" {
      type slave;
      masters { 192.168.10.1; };
      file "db.192";
};

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Gambar 6.23 Konfigurasi domain zone pada secondary dns server

Setelah mengkonfigurasi zone, kita tidak perlu mengkonfigurasi file forward maupun reverse di secondary dns server. Hal ini karena secondary dns server akan otomatis mengcopy file forward dan reverse dari primary dns server. Restart service bind9 di secondary dns server

```
root@forkits:/etc/bind# service bind9 restart
[....] Stopping domain name service...: bind9waiting for pid 3135 to die
. ok
[ ok ] Starting domain name service...: bind9.
root@forkits:/etc/bind#
```

Gambar 6.24 Proses restart dns server pada secondary dns server

Setelah service bind9 direstart, maka otomatis akan ada file forward dan file reverse di direktori /var/cache/bind

```
root@forkits2:/etc/bind# ls /var/cache/bind/
db.192  db.forkits  managed-keys.bind  managed-keys.bind.jnl
root@forkits2:/etc/bind#
```

Gambar 6.25 Pengecekan file forward dan reverse di secondary dns server

Sampai saat ini kita sudah selesai mengkonfigurasi primary dan secondary dns server. Selanjutnya kita akan melakukan pengujian dari komputer client. Namun sebelum itu, pastikan bahwa resolver sudah diarahkan ke primary dan secondary dns server seperti ini

```
admin@ubuntu:~$ cat /etc/resolv.conf
nameserver 192.168.10.1
nameserver 192.168.10.2
```

Gambar 6.26 Konfigurasi dns resolver di client

Untuk membuktikan kinerja dari secondary dns server, kita akan melakukan pengujian dengan dua kondisi. Kondisi pertama adalah saat kedua server dalam keadaan baik, sedangkan kondisi kedua adalah saat primary dns server dalam keadaan tidak baik (kita akan shutdowh primary dns server, sehingga primary dns server tidak bisa dihubungi). Berikut hasil pengujian pada kondisi pertama

```
admin@ubuntu:~$ nslookup forkits.com
Server:     192.168.10.1
Address:    192.168.10.1#53

Name:   forkits.com
Address: 192.168.10.1

admin@ubuntu:~$
```

Gambar 6.27 Pengujian saat primary dns server dalam keadaan *up*

Berikut pengujian pada kondisi kedua, jangan lupa untuk shutdown primary dns server terlebih dahulu

```
admin@ubuntu:~$ nslookup forkits.com
Server:     192.168.10.2
Address:    192.168.10.2#53

Name:   forkits.com
Address: 192.168.10.1

admin@ubuntu:~$
```

Gambar 6.28 Pengujian saat primary dns server dalam keadaan *down*

Perbedaan dari kedua pengujian pada dua kondisi yang berbeda diatas adalah terletak pada server yang digunakan. Perhatikan pada kondisi pertama, terlihat bahwa server yang digunakan adalah primary dns server. Sedangkan pada kondisi kedua, terlihat bahwa server yang digunakan adalah secondary dns server.

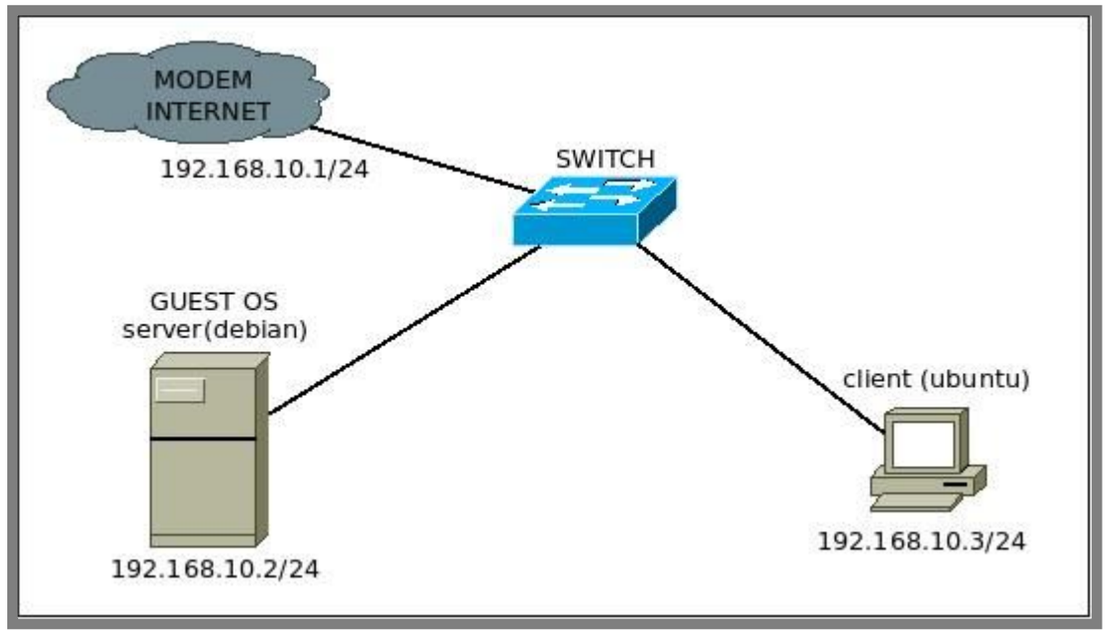
Cache Hint DNS Server

Tujuan utama penggunaan fitur ini adalah, client dapat meresolve domain di internet menggunakan dns server lokal. Perhatikan contoh berikut,

```
admin@ubuntu:~$ nslookup google.com
Server:         192.168.10.1
Address:        192.168.10.1#53

** server can't find google.com: SERVFAIL

admin@ubuntu:~$
```

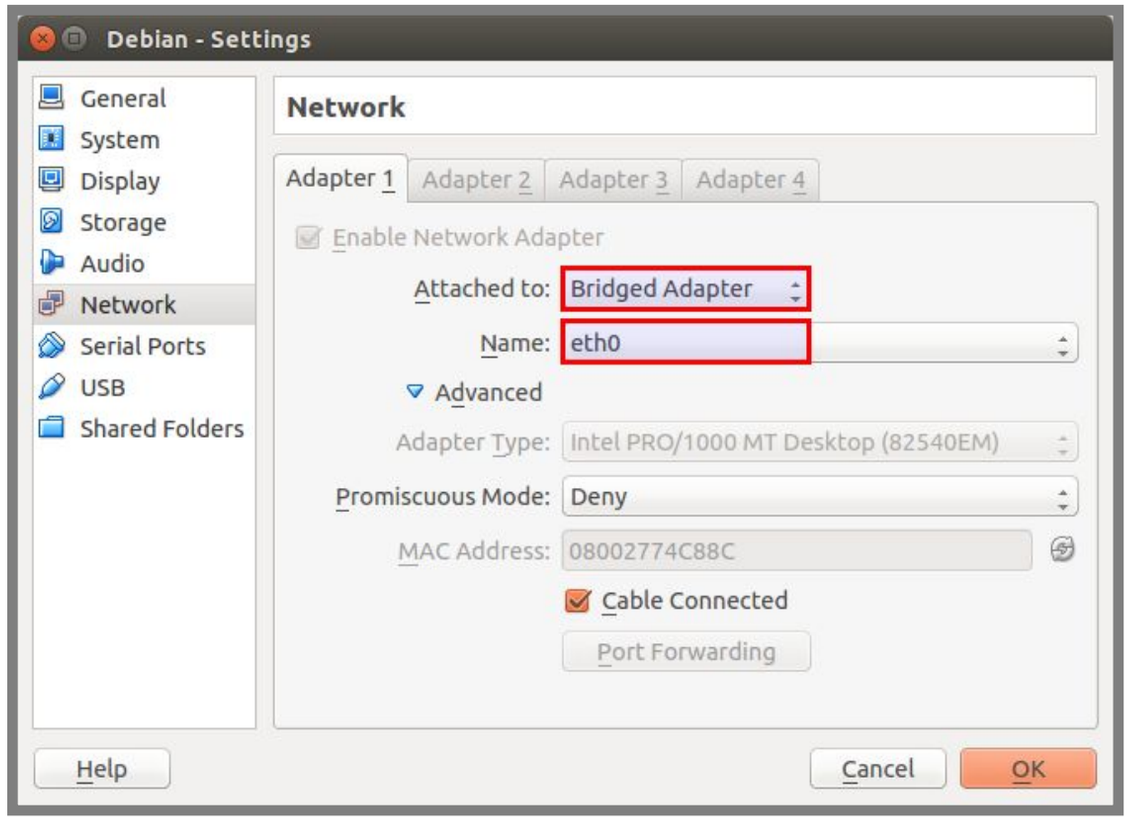
Gambar 6.29 Percobaan untuk meresolve domain di internet dengan dns server lokal

Pada contoh diatas terlihat bahwa dns server tidak bisa melayani permintaan untuk meresolve domain di internet. Untuk itu kita perlu melakukan beberapa konfigurasi. Berikut topologi yang akan kita praktikkan



Gambar 6.30 Topologi jaringa untuk prakti cache hint dns server

Perhatikan topologi diatas, terlihat bahwa kita membutuhkan akses internet dan dua komputer, komputer pertama diinstall virtualbox dengan sistem operasi debian didalamnya, dan komputer kedua digunakan sebagai client. Untuk mewujudkan topogi diatas, berikut konfigurasi network adapter yang perlu dilakukan untuk server



Gambar 6.31 Konfigurasi network adapter pada komputer server

Berikut konfigurasi ip address pada server, setelah itu pastikan bahwa server bisa ping ke internet

```
root@forkits:/etc/bind# nano /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.10.2
netmask 255.255.255.0
gateway 192.168.10.1
root@forkits:/etc/bind# service networking restart
[....] Running /etc/init.d/networking restart is deprecated because it may not
r[warnble some interfaces ... (warning).
[ ok ] Reconfiguring network interfaces...done.
root@forkits:/etc/bind# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=51 time=55.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=51 time=83.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=51 time=45.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=51 time=48.9 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=51 time=48.9 ms
```

Gambar 6.32 Konfigurasi ip address pada komputer server

Diasumsikan bahwa server telah dikonfigurasi sebagai primary dns server (silahkan merujuk ke sub bab sebelumnya, *konfigurasi primary dns server)*. Setelah dipastikan dns server bisa berjalan dengan baik, selanjutnya lakukan konfigurasi berikut untuk membuat cache hint dns server

```
root@forkits:/etc/bind# nano named.conf.options
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.   See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

         forwarders {
            8.8.8.8;
            8.8.4.4;
         };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.   See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation no;
        allow-recursion { any; };
        auth-nxdomain no;       # conform to RFC1035
        listen-on-v6 { any; };
};
```

Gambar 6.33 Konfigurasi cache hint dns server

Pada langkah diatas, kita menambahkan beberapa baris konfigurasi (teks warna hijau). Perhatikan pada bagian *forwarders,* terlihat bahwa kita menggunakan open dns yang ada diinternet, yaitu open dns google. Kita bisa saja menggunakan open dns selain milik google. Selanjutnya restart service bind9

```
root@forkits:/etc/bind# service bind9 restart
[....] Stopping domain name service...: bind9waiting for pid 2025 to die
. ok
[ ok ] Starting domain name service...: bind9.
root@forkits:/etc/bind#
```

Gambar 6.34 Proses restart service dns server

Berikut pengujian yang dilakukan dari komputer client, namun sebelumnya pastikan konfigurasi ip address sudah sesuai dengan topologi dan sudah bisa ping ke internet. Pastikan juga bahwa resolver sudah diarahkan ke ip address server

```
client@ubuntu:~$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:1e:ec:5e:2f:d7
          inet addr:192.168.10.3 Bcast:192.168.10.255 Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
client@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=51 time=55.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=51 time=83.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=51 time=45.8 ms
client@ubuntu:~$ cat /etc/resolv.conf
nameserver 192.168.10.2
client@ubuntu:~$ nslookup google.com
Server:     192.168.10.2
Address:    192.168.10.2#53

Non-authoritative answer:
Name:  google.com
Address: 74.125.200.139
Name:  google.com
Address: 74.125.200.113
Name:  google.com
Address: 74.125.200.102
Name:  google.com
Address: 74.125.200.100
Name:  google.com
Address: 74.125.200.138
Name:  google.com
Address: 74.125.200.101

client@ubuntu:~$ nslookup forkits.com
Server:     192.168.10.2
Address:    192.168.10.2#53

Name:  forkits.com
Address: 192.168.10.2

client@ubuntu:~$
```

Gambar 6.35 Pengujian dari komputer client

Perhatikan gambar diatas, terlihat bahwa komputer client telah bisa meresolve domain dinternet menggunakan dns server lokal. Selain itu, client juga bisa meresolve domain lokal (forkits.com).

DNS Filtering dengan Bind RPZ

Selain fungsi-fungsi yang telah disampaikan di sub bab sSetelah menyelesaikan proses import tersebut, cari dan pastikan bahwa CA kita telah terdaftar sebagai CA terpercaya
ebelumnya, dns server bisa juga digunakan untuk melakukan filtering.

Saat ini sedang gencar-gencarnya dilakukan filtering untuk situs-situs yang mengandung konten pornografi. Untuk melakukan hal tersebut, metode yang paling mudah untuk dilakukan adalah menggunakan dns filtering yang cara kerjanya adalah meredirect request dns dari client ke halaman web tertentu yang berisi sebuah peringatan.

Untuk membuat dns filtering, kita akan praktik menggunakan topologi pada gambar 6.30. Diasumsikan bahwa dns server sudah bisa berjalan dengan baik, termasuk fitur cache hint. Sehingga client sudah bisa meresolve domain lokal dan domain diinternet. Berikut konfigurasi yang perlu ditambahkan

```
root@forkits:/etc/bind# nano named.conf.options
options {
    directory "/var/cache/bind";
    ..........................................................................................
    ..........................................................................
    ..................................................................
     forwarders {
        8.8.8.8;
        8.8.4.4;
     };
    //========================================================================
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.   See https://www.isc.org/bind-keys
    //========================================================================
    dnssec-validation no;
    allow-recursion { any; };
    auth-nxdomain no;      # conform to RFC1035
    listen-on-v6 { any; };
    response-policy { zone "rpz.zone"; };
};
zone "rpz.zone" {
    type master;
    file "/etc/bind/db.rpz";
};
```

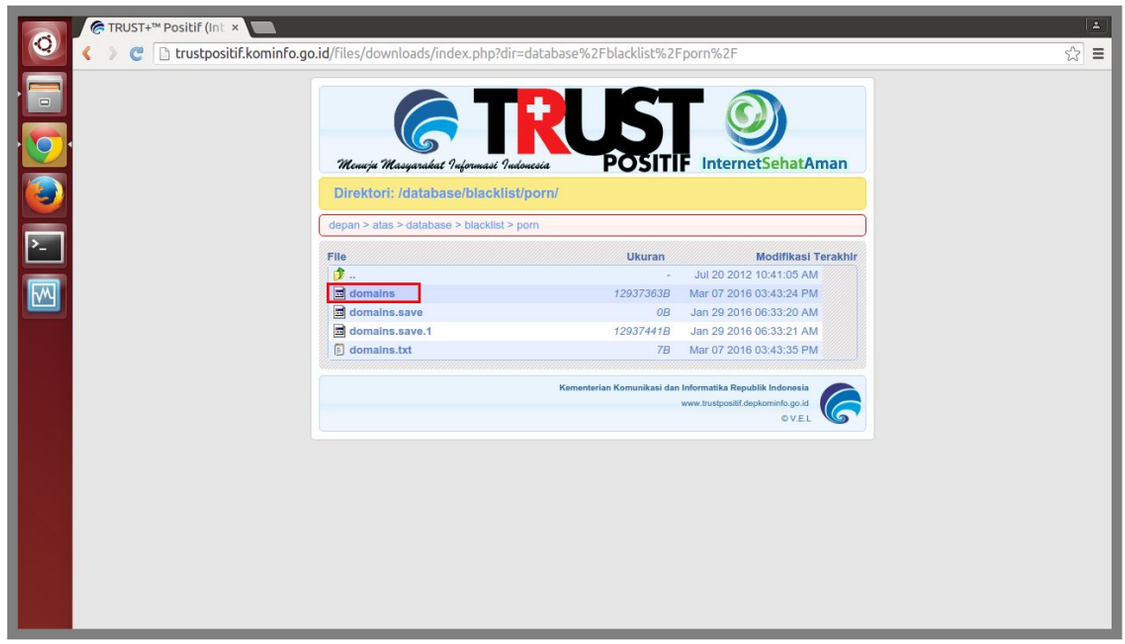
Gambar 6.36 Konfigurasi dns filtering dengan rpz

Perhatikan gambar diatas, terlihat bahwa kita menambahkan beberapa baris konfigurasi (teks warna hijau) untuk membuat sebuah domain dengan nama *rpz.zone.* Selanjutnya buat file forward untuk domain tersebut

```
root@forkits:/etc/bind# cp db.local db.rpz
root@forkits:/etc/bind#
```

Gambar 6.37 Membuat file forward untuk domain rpz.zone

Kita tidak perlu melakukan konfigurasi sama sekali terhadap file forward tersebut. Namun nantinya kita akan menambahkan domain-domain yang akan difilter ke file tersebut. Untuk domain-domain yang akan difilter, kita bisa menentukannya sendiri atau bisa download database domain-domain yang mengandung konten purnografi dari kominfo (http://trustpositif.kominfo.go.id/files/).

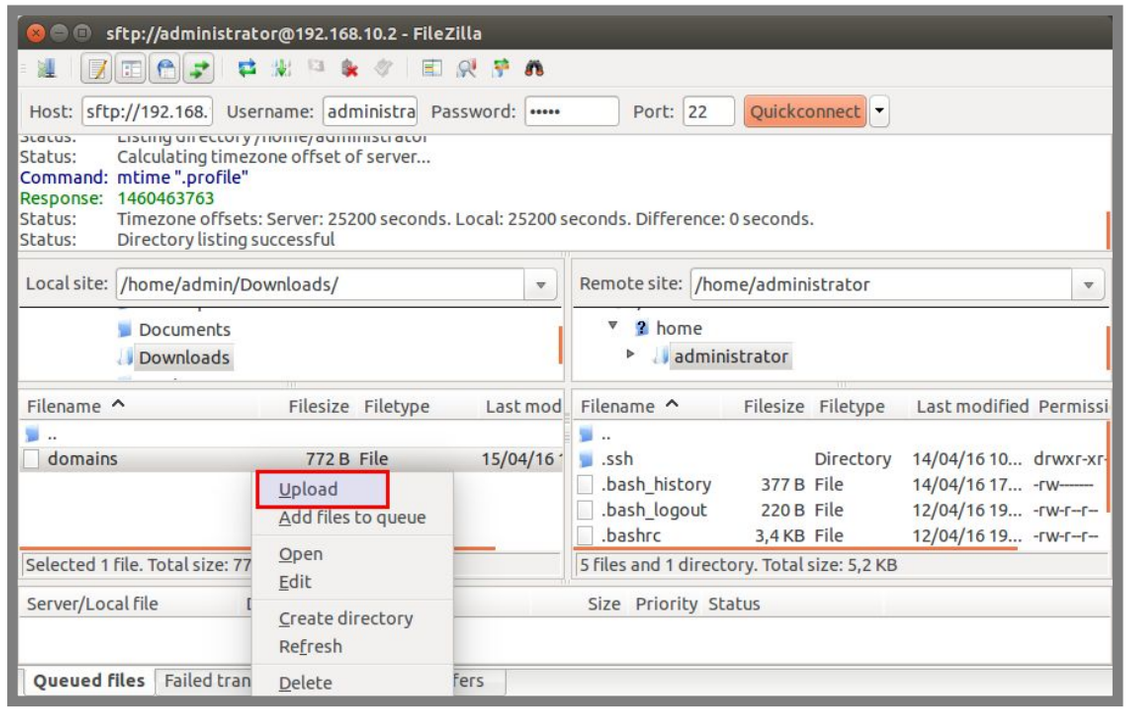


Gambar 6.37 Download database domain pornografi dari kominfo

Dalam database tersebut, terdapat sekitar 746.933 domain yang mengandung konten purnografi. Dengan jumlah sekian, kita membutuhkan spesifikasi komputer server yang lumayan tinggi, jika spsesifikasi server tidak memadai, maka server akan hang dan dns server tidak akan berjalan dengan baik.

Kita hanya praktik menggunakan server di virtualbox, oleh karena itu tidak memungkinkan untuk memfilter seluruh domain yang ada di database tersebut. Kita hanya akan memfilter 50 domain pertama yang ada didalam database tersebut, kita bisa menggunakan gedit/notepad untuk menghapus domain didalam database tersebut dan menyisakan hanya 50 domain saja.

Langkah selanjutnya yang harus kita lakukan adalah menambahkan database domain yang telah kita download dan edit tadi ke file forward yang telah kita buat sebelumnya. Untuk itu, kita harus upload database tersebut ke komputer server. Kita bisa menggunakan fitur sftp yang telah kita bahas di bab 5.



Gambar 6.38 Upload database domain porno ke dns server

Berikut perintah yang dapat kita gunakan untuk menambahkan database domain yang akan difilter ke file forward

```
root@forkits:/etc/bind# cp /home/administrator/domains /etc/bind/
root@forkits:/etc/bind# ls
bind.keys   db.255      db.local   named.conf                rndc.key
db.0        db.debian   db.root    named.conf.default-zones  zones.rfc1918
db.127      db.empty    db.rpz     named.conf.local
db.192      db.forkits  domains    named.conf.options
root@forkits:/etc/bind# awk '{print $1" IN A 192.168.10.2"}' domains>>db.rpz
root@forkits:/etc/bind# awk '{print "*."$1" IN A 192.168.10.2"}' domains>>db.rpz
root@forkits:/etc/bind#
```

Gambar 6.39 Menambahkan database domain porno ke file forward rpz.zone

Perhatikan gambar diatas, perintah pertama digunakan untuk copy database domain yang telah di upload ke direktori /etc/bind/, diikuti perintah kedua untuk memastikan apakah database telah berhasil dicopy.

Perintah ketiga digunakan untuk menambahkan domain yang ada didatabase ke file forward dengan format *domain IN A ip_address.* Sedangkan perintah keempat digunakan untuk menambhkan domain yang ada didatabase ke file forward dengan format **.domain IN A ip_address.* Contohnya sebagai berikut

```
17bb.info IN A 192.168.10.2
*.17bb.info IN A 192.168.10.2
```

Gambar 6.40 Gambaran umum isi file forward rpz.zone

Jika hanya ada entry baris pertama (17bb.info) saja, maka sub domain dari domain tersebut (misalnya www.17bb.info, web.17bb.info) tidak akan ikut terfilter. Tanda bintang (*) sebelum domain tersebut artinya semua subdomain.

Langkah terahir yang perlu kita lakukan adalah restart service bind9

```
root@forkits:/etc/bind# service bind9 restart
[....] Stopping domain name service...: bind9waiting for pid 1999 to die
. ok
[ ok ] Starting domain name service...: bind9.
root@forkits:/etc/bind#
```

Gambar 6.41 Proses restart service dns server

Berikut hasil pengujian saat client mencoba untuk meresolve domain yang difilter

```
client@ubuntu:~$ nslookup 17bb.info
Server:         192.168.10.2
Address:        192.168.10.2#53

Non-authoritative answer:
Name:   17bb.info
Address: 192.168.10.2

client@ubuntu:~$
```

Gambar 6.42 Pengujian dari komputer client

Perhatikan hasil pengujian diatas, terlihat bahwa ip address dari 17bb.info adalah 192.168.10.2 (ip address server). Hal ini membuktikan bahwa kita telah berhasil melakukan filtering domain yang mengandung konten pornografi.

Untuk memberikan kesan yang lebih hidup, kita bisa membuat sebuah halaman web pada komputer server yang berisi sebuah peringatan. Sehingga saat ada client yang mencoba mengakses situs yang berbau purnografi, akan diredirect ke halaman web yang berisi peringatan tersebut. Materi tentang webserver akan dibahas di bab berikutnya.

---END OF CHAPTER---

Bab 7
Certificate Authority

Certificate Authority (CA) adalah perusahaan yang bertugas memberikan sertifikat pada ssl/tls. Secure Socket Layer (SSL) atau dikenal juga dengan istilah Transport Secure Layer (TSL) adalah sebuah protocol yang bertugas memastikan keaslian suatu layanan, entah itu layanan web, mail, ataupun ftp.

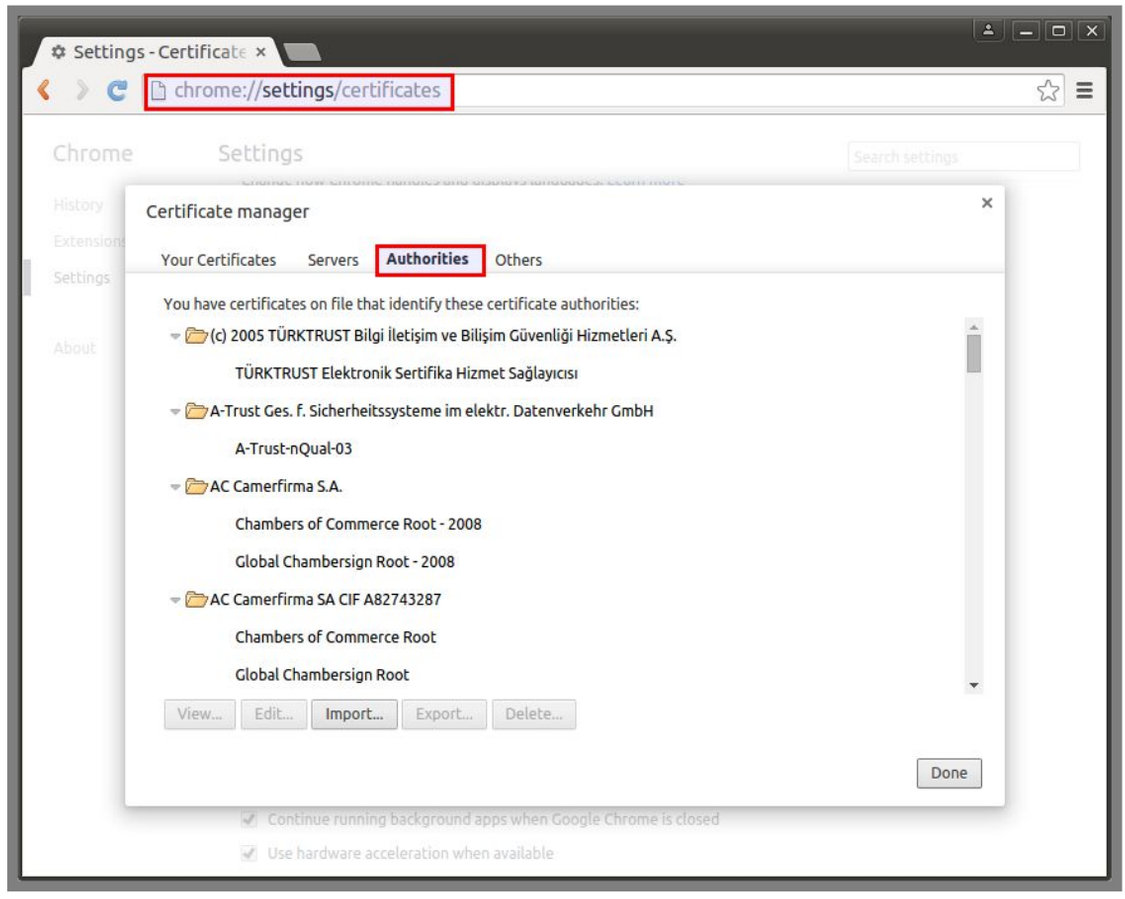
Analogi dari penerapan ssl/tsl adalah sebagai berikut. Saat kita bertemu dengan calon mitra kerja yang sebelumnya telah berkomunikasi via telephone. Bagaimana kita yakin bahwa orang yang ada didepan kita itu adalah orang yang sebelumnya berkomunikasi via telephone dengan kita? Bagaimana kita yakin bahwa orang tersebut bukan orang yang menyamar menjadi calon mitra kerja kita? Hal yang paling lazim dilakukan adalah menyuruh orang tersebut menunjukkan bukti identitas, entah itu KTP, SIM, atau yang lain.

Saat kita mengakses sebuah website toko online, bagaimana kita yakin bahwa website tersebut merupakan website asli yang dikelola oleh pemilik toko online tersebut? Bagaimana kita yakin bahwa website tersebut bukan website palsu yang dikelola oleh orang yang tidak bertanggung jawab? Jika kita menyamakan masalah ini dengan analogi diatas, maka kita akan meminta website tersebut menunjukkan bukti identitas seperti KTP ataupun SIM. Bukti identitas yang dimiliki oleh sebuah website disebut *private key.*

Private key merupakan sebuah file yang menjadi bukti identitas suatu server. Oleh karena itu, file ini harus benar-benar kita lindungi dengan baik, karena jika file ini sampai dimiliki oleh orang lain, sama saja kita memberikan KTP kita ke orang lain.

Lembaga yang berhak mengeluarkan KTP yang falid hanyalah pemerintah pusat. Jika ada lembaga lain mengeluarkan KTP, maka KTP tersebut tidak akan falid (palsu). Begitu juga dengan private key, lembaga yang berhak mengeluarkan private key yang falid disebut *certificate authority.* jika ada lembaga/orang lain yang mengeluarkan private key, maka private key tersebut tidak akan failid (palsu).

Untuk mendapatkan private key yang falid kita harus membeli sertifikat dari salah satu certificate authority terpercaya yang ada didunia. Terdapat banyak sekali certificate authority terpercaya yang berhak mengeluarkan private key yang falid di dunia ini, seperti comodo, verisgn, cybertrust/verizon, startssl, dll. Berikut daftar certificate authority terpercaya yang ada di web browser chrome.



Gambar 7.1 Daftar CA terpercaya

Terpercaya atau tidaknya suatu certificate authority bisa kita lihat di web browser kita masing-masing. Jika suatu certificate authority terdapat pada web browser, berarti certificate atuhority tersebut terpercaya, begitu juga sebaliknya.

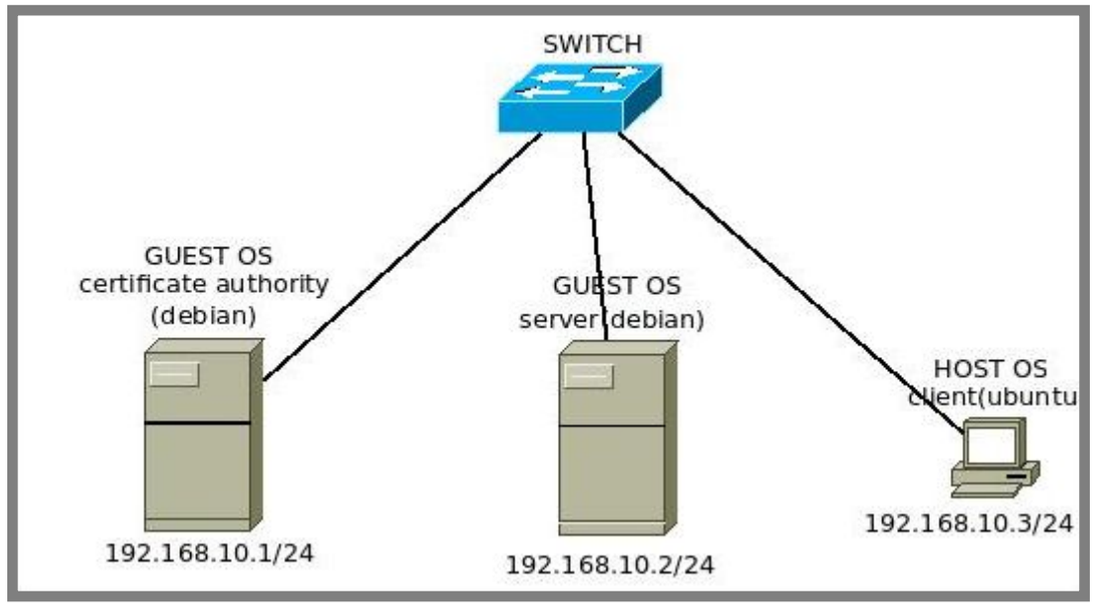
Perbedaan mendasar dari certificate atuthority terpercaya dan yang tidak terpercaya terletak pada private key yang dihasilkan. Jika certificate authority terpercaya, maka private key yang dihasilkan akan falid dan sertifikat ssl juga akan terpercaya. Berikut contoh penggunakan sertifikat ssl yang terpercaya.

Gambar 7.2 Contoh CA terpercaya

Sedangkan jika ssl tidak terpercaya, maka hasilnya akan seperti ini

Gambar 7.3 Contoh CA tidak terpercaya

Berikut topologi yang akan kita praktikkan pada bab ini



Gambar 7.4 Topologi jaringan untuk praktik CA Server

Nantinya komputer server akan mengajukan permohonan kepada certificate authority untuk menyetujui sertifikat miliknya. Kemudian client akan mendaftarkan certificate authority yang kita buat menjadi salah satu certificate authority terpercaya di browser nya.

Pada bab ini, kita akan praktik seolah-olah menjadi tiga pihak yang berbeda, yaitu sebagai CA, sebagai pemilik server, dan juga sebagai client.

Mengajukan Permohonan ke CA

Note : Praktik pada sub bab ini dilakukan pada komputer server

Ada dua file yang perlu disiapkan untuk mengajukan permohonan persetujuan sertifikat kepada CA, yaitu private key dan dokumen persetujuan sertifikat yang disebut Certificate Signing Request (CSR). Berikut langkah-langkah untuk membuat private key

```
root@forkits:~# openssl genrsa -aes256 2048 > forkits.com.key
Generating RSA private key, 2048 bit long modulus
.....................................+++
.....................+++
e is 65537 (0x10001)
Enter pass phrase: (tidak terlihat)
Verifying - Enter pass phrase: (tidak terlihat)
root@forkits:~#
```

Gambar 7.5 Membuat private key pada server

Perintah diatas digunakan untuk membuat sebuah private key dan disimpan dengan nama forkits.com.key. Kita bisa menggunakan nama apa saja untuk file private key tersebut, namaun disarankan untuk menggunakan nama domain yang kita gunakan. Selanjutnya berikut perintah yang digunakan untuk membuat CSR

```
root@forkits:~# openssl req -new -key forkits.com.key -out forkits.com.csr
Enter pass phrase for forkits.com.key: (tidak terlihat)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ID
State or Province Name (full name) [Some-State]:Jawa Timur
Locality Name (eg, city) []:Blitar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Komunitas IT
Organizational Unit Name (eg, section) []:SysAdmin
Common Name (e.g. server FQDN or YOUR name) []:forkits.com
Email Address []:admin@forkits.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: (enter)
An optional company name []: (enter)
root@forkits:~#
```

Gambar 7.6 Membuat CSR pada komputer server

Berikut keterangan dari masing-masing langkah yang dilakukan diatas

Syntak	Deskripsi
Perintah pertama	Digunakan untuk membuat CSR dan disimpan dengan nama forkits.com.csr
(tidak terlihat)	Masukkan password yang kita gunakan saat membuat file private key sebelumnya
ID	Kode dari negara kita
Jawa Timur	Provinsi tempat kita tinggal
Blitar	Kota tempat kita tinggal
Komunitas IT	Nama organisasi tempat kita bekerja
SysAdmin	Divisi tempat kita bekerja
forkits.com	Domain dari organisasi kita
admin@forkits.com	Email administrator ceritificate authority (CA)
(enter) 2 kali	Kita diminta untuk memasukkan extra password. Bagian ini opsional, bisa diisi, bisa juga tidak

Kita telah membuat dua file yang diperlukan, yaitu private key dan CSR. Langkah selanjutnya adalah mengirimkan CSR kepada certificate authority. Untuk mengirimkan CSR kepada server, kita bisa menggunakan emai atau media transfer file yang lain. Setelah itu, sebagai pemilik server kita tinggal menunggu CSR kita disetujui oleh CA.

Pekerjaan sebagai pemilik server telah selesai, selanjutnya kita akan bekerja seolah-olah menjadi certificate authority. Pekerjaan yang perlu dilakaukan oleh ceritificate authority akan dibahas di sub bab berikut.

Membuat Certificate Authority

Note : Praktik pada sub bab ini dilakukan pada komputer Certificate Authority

Telah dijelaskan sebelumnya, bahwa certificate authority (CA) adalah suatu lembaga yang bertugas memberikan sertifikat ssl kepada suatu server. Pada sub bab ini kita akan membuat sebuah certificate authority untuk digunakan pada jaringan lokal. Karena untuk membuat suatu certificate authority yang diakui di dunia, tidaklah mudah untuk mengurus perijinan dan juga tidak kecil dalam hal biaya.

Langkah-langkah yang perlu dilakukan untuk membuat CA adalah sebagai berikut

```
root@ca:~# mkdir /ca
root@ca:~# mkdir /ca/ca_dir
root@ca:~# cd /ca/ca_dir/
root@ca:/ca/ca_dir# openssl genrsa -aes256 2048 > ca.key
Generating RSA private key, 2048 bit long modulus
.............+++
...........+++
e is 65537 (0x10001)
Enter pass phrase: (tidak terlihat)
Verifying - Enter pass phrase: (tidak terlihat)
root@ca:/ca/ca_dir#
```

Gambar 7.7 Proses pembuatan private key untuk CA

Perintah-perintah diatas digunakan untuk membuat direktori yang akan menjadi lokasi penyimpanan file-file yang diperlukan oleh certificate authority dan membuat private key yang diperlukan oleh certificate authority. Selanjutnya kita harus membuat sertifikat dengan memanfaatkan private key yang telah kita buat. Sertifikat ini nantinya yang akan digunakan oleh certificate authority untuk menyetujui CSR yang diajukan oleh pemilik server.

```
root@ca:/ca/ca_dir# openssl req -new -x509 -days 3650 -key ca.key>ca.crt
Enter pass phrase for ca.key: (tidak terlihat)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ID
State or Province Name (full name) [Some-State]:Jawa Timur
Locality Name (eg, city) []:Blitar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CA
Organizational Unit Name (eg, section) []:Certificate Authority
Common Name (e.g. server FQDN or YOUR name) []:ca.com
Email Address []:admin@ca.com
root@ca:/ca/ca_dir#
```

Gambar 7.8 Membuat file sertifikat untuk keperluan CA

Berikut keterangan hal-hal yang dilakukan diatas

Syntak	Deskripsi
Perintah pertama	Digunakan untuk membuat sertifikat dengan nama file ca.crt
(tidak terlihat)	Masukkan password yang kita gunakan saat membuat file private key sebelumnya
ID	Kode dari negara kita
Jawa Timur	Provinsi tempat kita tinggal
Blitar	Kota tempat kita tinggal
CA	Nama organisasi kita, untuk memudahkan mengingat, kita menggunakan CA saja
Certificate Authority	Divisi tempat kita bekerja
ca.com	Domain dari organisasi kita
admin@ca.com	Email administrator ceritificate authority (CA)

Selanjutnya kita akan membuat sebuah file konfigurasi yang diperlukan untuk proses penyetujuan CSR yang diajukan oleh server.

```
root@ca:/ca/ca_dir# cd ..
root@ca:/ca# nano config_ca
[ca]
default_ca              = ca

[ca]
dir                     = ca_dir

# sertifikat dan private key CA
certificate             = $dir/ca.crt
private_key             = $dir/ca.key

# Folder penyimpanan
cert                    = $dir/sertifikat-customer
new_certs_dir           = $dir/sertifikat-customer
crl                     = $dir/sertifikat-batal

# Database sertifikat yang sudah dikeluarkan
database                = $dir/database.txt
serial                  = $dir/serial.txt

# Nilai default untuk sertifikat baru
default_days            = 365      # masa berlaku sertifikat customer
default_crl_days        = 30       # masa berlaku daftar pembatalan
sertifikat
default_md                  = sha1
x509_extensions         = usr_cert

policy = policy-saya
x509_extensions         = certificate_extensions

[ policy-saya ]
commonName              = supplied
stateOrProvinceName     = supplied
countryName             = supplied
emailAddress            = supplied
organizationName        = supplied
organizationalUnitName = optional

[ certificate_extensions ]
basicConstraints        = CA:false

[ req ]
default_keyfile         = ca_dir/ca.key
```

Gambar 7.9 File konfigurasi untuk proses penyetujuan CSR yang diajukan komputer server

Langkah selanjutnya, kita harus membuat file dan direktori tambahan yang diperlukan oleh certificate authority

```
root@ca:/ca# cd ca_dir/
root@ca:/ca/ca_dir# touch database.txt
root@ca:/ca/ca_dir# touch database.txt.attr
root@ca:/ca/ca_dir# echo 0000 > serial.txt
root@ca:/ca/ca_dir# mkdir sertifikat-request
root@ca:/ca/ca_dir# mkdir sertifikat-customer
root@ca:/ca/ca_dir# ls
ca.crt   database.txt       sertifikat-customer
ca.key   database.txt.attr  sertifikat-request
root@ca:/ca/ca_dir#
```

Gambar 7.10 Proses pembuatan file dan direktori yang diperlukan CA

Telah dijelaskan sebelumnya, bahwa pekerjaan terahir yang harus dilakukan oleh pemilik server adalah mengirimkan file CSR kepada ceritificate authoriy. Umumnya, kita bisa mengirimkan file CSR tersebut via email. Namun jika hanya untuk keperlua belajar seperti saat ini, kita bisa menggunakan file transfer seperti biasa. Kita bisa memanfaatkan SFTP untuk mengirimkan file CSR kepada certificate authority. Pembahasan selanjutnya akan diasumsikan bahwa file CSR telah dikirimkan ke certificate authority.

Penyetujuan CSR oleh Certificate Authority

Note : Praktik pada sub bab ini kita akan bekerja pada komputer Certificate Authority

Sampai saat ini kita telah bekerja sebagai pemilik server dan telah mengirimkan CSR kepada CA. Kita juga telah selesai melakukan konfigurasi pada certificate authority.

Pada sub bab ini, kita akan bekerja sebaga CA dan menyetujui CSR yang telah diajukan oleh pemilik server. Sebelum melakukan hal tersebut, kita harus memastikan bahwa CSR yang telah dikirimkan oleh pemilik server telah kita masukkan ke direktori *sertifikat-request*

```
root@ca:/ca/ca_dir# cd ..
root@ca:/ca# ls ca_dir/sertifikat-request/
forkits.com.csr
root@ca:/ca#
```

Gambar 7.11 Melihat file CSR yang diajukan oleh server

Terlihat bahwa CSR telah berada didreiktori *sertifikat-request,* selanjutnya untuk menyetujui CSR tersebut kita bisa menggunakan perintah berikut

```
root@ca:/ca# openssl ca -config config_ca -in ca_dir/sertifikat-request/forkits.com.csr
Using configuration from config_ca
Enter pass phrase for ca_dir/ca.key: (tak terlihat)
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName               :PRINTABLE:'ID'
stateOrProvinceName       :ASN.1 12:'Jawa Timur'
localityName              :ASN.1 12:'Blitar'
organizationName          :ASN.1 12:'Komunitas IT'
organizationalUnitName    :ASN.1 12:'SysAdmin'
commonName                :ASN.1 12:'forkits.com'
emailAddress              :IA5STRING:'admin@forkits.com'
Certificate is to be certified until Apr 17 02:49:21 2017 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=ID, ST=Jawa Timur, L=Blitar, O=CA, OU=Certificate Authority,
CN=ca.com/emailAddress=admin@ca.com
        Validity
            Not Before: Apr 17 02:49:21 2016 GMT
            Not After : Apr 17 02:49:21 2017 GMT
        Subject: CN=forkits.com, ST=Jawa Timur, C=ID/emailAddress=admin@forkits.com,
O=Komunitas IT, OU=SysAdmin
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:bf:95:3a:70:7f:9c:7b:26:fb:23:b0:8f:86:a2:
                    2a:f9:df:ff:af:84:b4:a0:db:ea:ff:dd:59:0e:df:
                    67:e4:b1:11:e5:eb:ca:9f:7e:9d:89:4a:1f:6c:e9:
                    ..........................................................................
                    ..........................................................................
..................................................................................................
..........................................................................
root@ca:/ca#
```

Gambar 7.12 Menyetujui CSR yang diajukan oleh server

Setelah perintah diatas dijalankan, maka akan dibuatkan sebuah file secara otomatis didalam direktori sertifikat-customer

```
root@ca:/ca# ls ca_dir/sertifikat-customer/
00.pem
root@ca:/ca#
```

Gambar 7.13 Hasil sertifikat yang telah disetuji oleh CA

Selanjutnya hal yang harus kita lakukan sebagai CA adalah mengirimkan file tersebut kepada pemilik server yang telah mengirimkan CSR. Namun sebelum mengirimkannya, ada baiknya jika kita sebagai CA merubah nama file tersebut menjadi nama yang sesuai dengan domain pengirim CSR (pemilik server)

```
root@ca:/ca# cd ca_dir/sertifikat-customer/
root@ca:/ca/ca_dir/sertifikat-customer# mv 00.pem forkits.com.crt
root@ca:/ca/ca_dir/sertifikat-customer# ls
forkits.com.crt
root@ca:/ca/ca_dir/sertifikat-customer#
```

Gambar 7.14 Merubah nama sertifikat yang telah disetujui

Setelah direname, kita harus segera mengirimkan file sertifikat yang telah kita setujui (sebagai ca) tersebut kepada pemilik server. Kita bisa mengirimnya menggunakan email. Tapi karena kita hanya melakukan ini untuk pembelajaran, kita bisa mengirimkannya ke server menggunakan file transfer SFTP seperti biasa.

Untuk saat ini, pemilik server akan memiliki tiga file, yaitu private key, CSR, dan sertifikat yang telah disetujui oleh CA. Nantinya yang dibutuhkan oleh pemilik server hanya private key dan sertifikat dari CA, sehingga file CSR boleh dihapus setelah mendapat sertifikat dari CA.

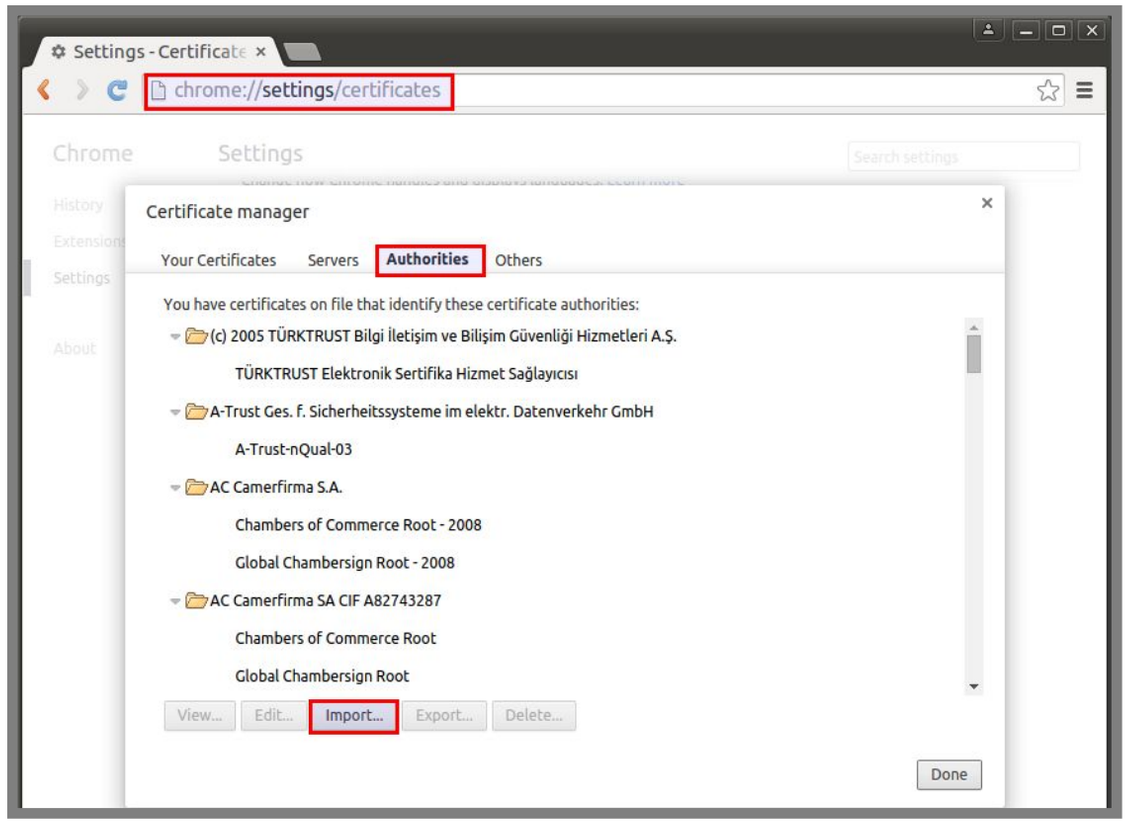
Mendaftarkan CA di Client

Pada awal bab kita telah membahas bahwa hanya ca terpercayalah yang bisa mengeluarkan sertifikat ssl. Jadi sertifikat ssl yang dikeluarkan oleh ca yang kita buat tadi tidaklah falid, hal ini karena ca yang kita buat tidak terpercaya.

Untuk membuat agar ca yang kita buat tadi terpercaya, kita harus import sertifikat milik ca di komputer client. Diasumsikan bahwa kita telah download file sertifikat dari komputer CA (file yang bernama ca.crt).

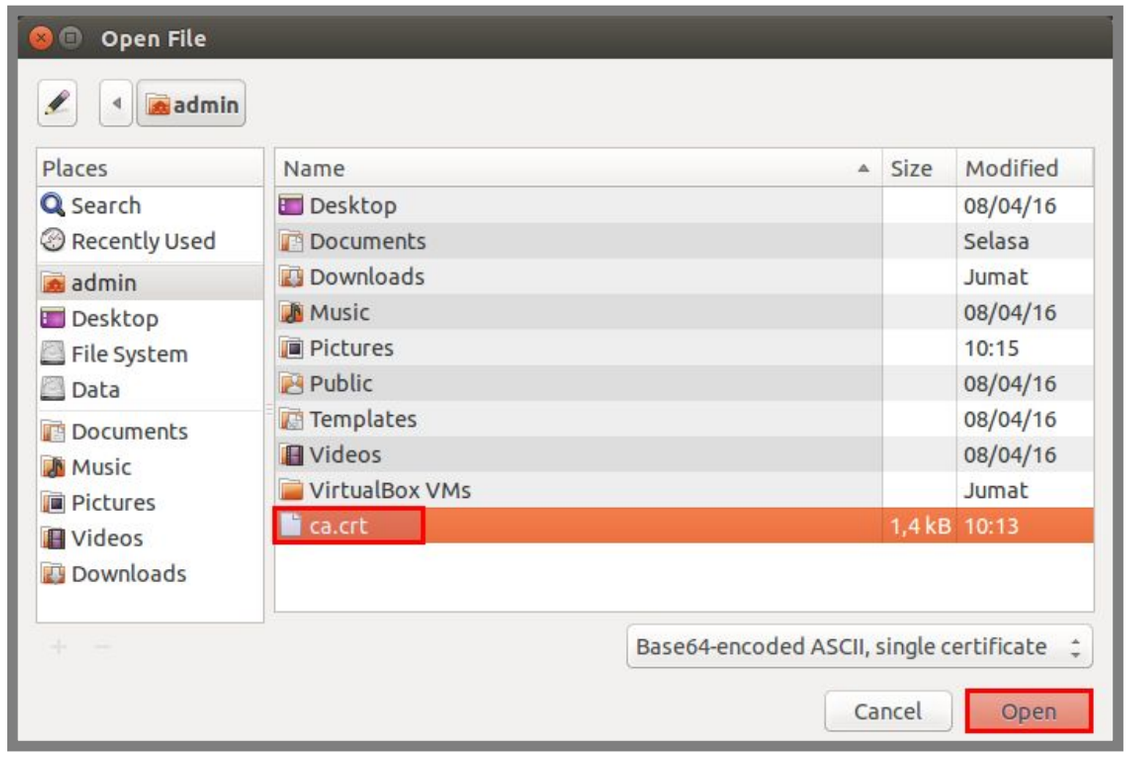
Sedangkan browser yang akan kita gunakan adalah google chrome, jika teman-teman menggunakan browser lain, bisa menyesuaikan langkah-langkah yang dijelaskan pada buku ini

Masuk pada setting -> certificates, kemudian pilih bagian authorities dan klik import.

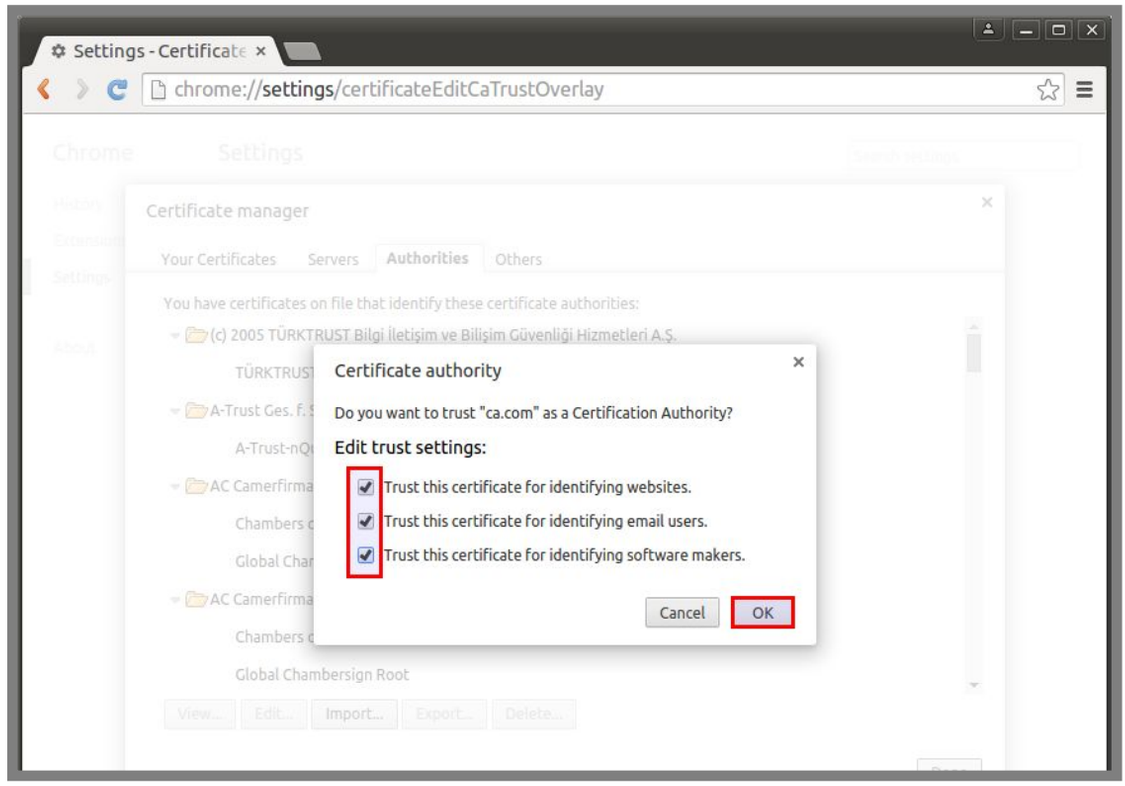


Gambar 7.15 Menambahkan CA terpercaya di Google Chrome

Selanjutnya cari lokasi penyimpanan file sertifikat CA yang telah kita download sebelumnya

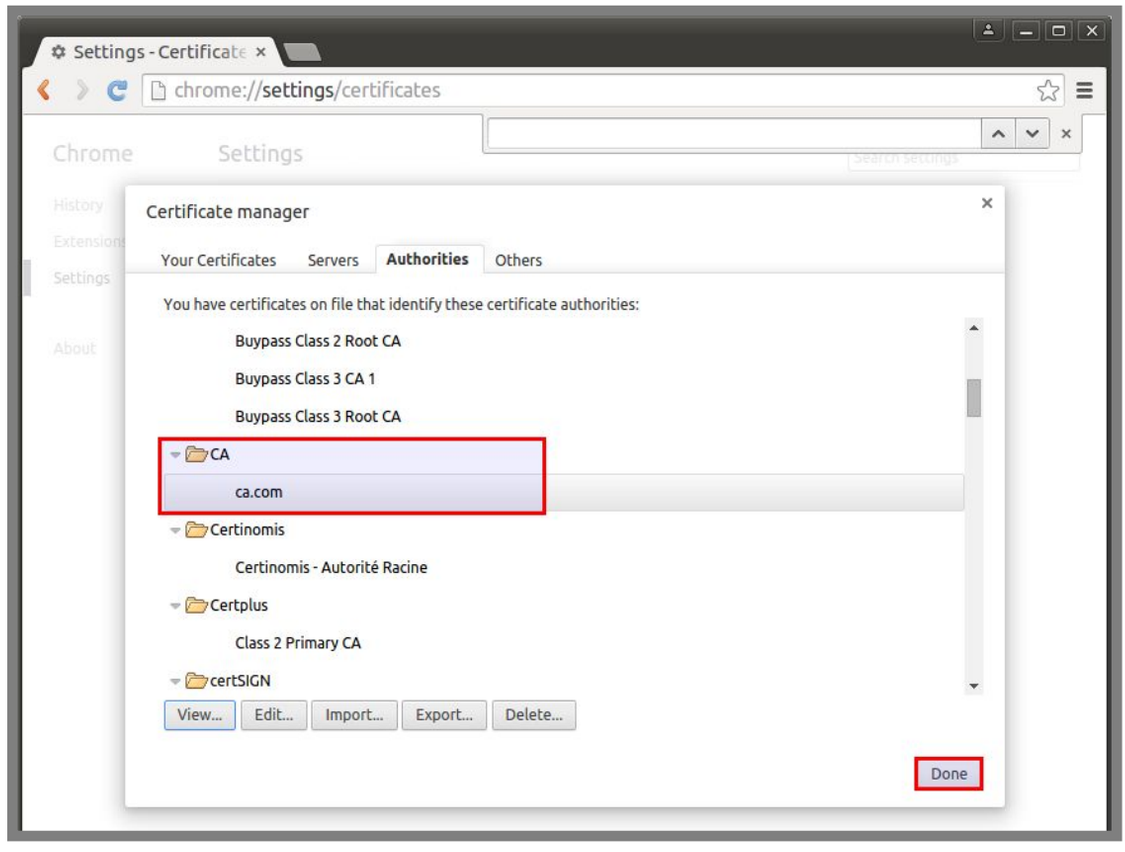


Gambar 7.16 Mencari file sertifikat milik CA



Gambar 7.17 Proses import sertifikat CA

Setelah menyelesaikan proses import tersebut, cari dan pastikan bahwa CA kita telah terdaftar sebagai CA terpercaya



Gambar 7.18 Hasil import file sertifikat CA

Terlihat bahwa nama CA kita telah ada didaftar CA terpercaya. Itu artinya CA yang kita buat tadi telah berhak mengeluarkan sertifikat ssl yang falid. Namun perlu diketahui bahwa CA kita hanya falid bagi client yang telah menambahkan sertifikat CA ke browsernya.

Pembahasan materi pada bab ini akan bermanfaat untuk pembahasan materi pada bab-bab selanjutnya, seperti pembahasan pada bab web server, ftp server, dll.

---END OF CHAPTER---

Bab 8
Web & Database Server

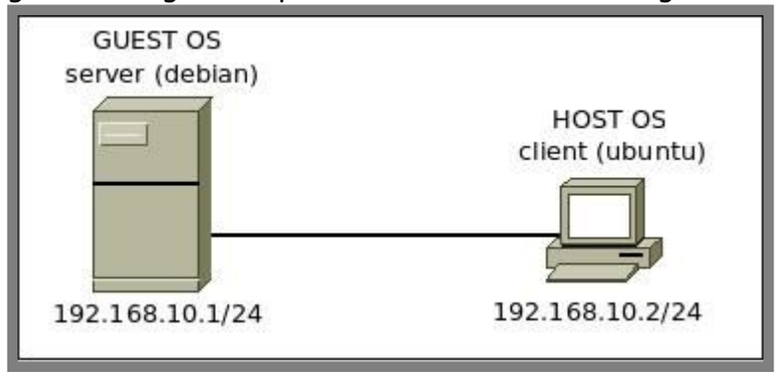
Web server merupakan sebuah layanan yang memungkinkan untuk menampilkan sebuah halaman web kepada client di dunia internet. Sadar atau tidak, kita telah melihat contoh penerapan web server setiap hari. Seperti saat kita melihat berita-berita terbaru di www.detik.com atau saat kita berbelanja online di www.lazada.com ataupun saat kita mencari hiburan di www.komikid.com. Semuanya adalah penerapan dari web server.

Ada dua hal yang tidak pernah lepas dari web server, yang pertama adalah domain name system (dns) yang telah kita bahas di bab sebelumnya. Hal ini dibuktikan dengan kebiasaan kita membuka suatu halaman web menggunakan domain name, bahkan kita tidak pernah membukanya dengan ip address. Inilah fungsi dari domain name system. Hal yang kedua adalah database, seluruh data/informasi yang ditampilkan di halaman web tersimpan didalam database server. Mulai dari artikel, jumlah pengunjung, daftar barang, harga barang, dll.

Pada bab ini kita akan membahas konfigurasi web server dan beberapa extra konfigurasi yang kadang kala dibutuhkan dalam sebuah jaringan. Nantinya kita akan membahas materi tentang database server di pertengahan bab ini.

Konfigurasi Web Server

Topologi yang akan kita gunakan pada sub bab ini adalah sebagai berikut



Gambar 8.1 Topologi jaringan untuk praktik web server

Diasumsikan bahwa di server dan client telah dikonfigurasi ip address sesuai topologi diatas. Diasumsikan juga bahwa di server telah dikonfigurasi sebagai primary dns server seperti yang telah kita bahas pada sub bab primary dns server.

Terdapat beberapa aplikasi yang bisa kita manfaatkan untuk membuat sebuah web server, diantaranya apache, nginx, tomcat, dll. Diantara beberapa macam aplikasi tersebut, yang akan dibahas pada bab ini adalah pembuatan web server menggunakan apache.

Berikut perintah yang digunakan untuk menginsatall aplikasi apache.

```
root@forkits:~# apt-get install apache2 php5
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-mpm-prefork apache2-utils apache2.2-bin apache2.2-common
  libapache2-mod-php5 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libonig2 libqdbm14 php5-cli php5-common ssl-cert
Suggested packages:
  apache2-doc apache2-suexec apache2-suexec-custom php-pear
openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-mpm-prefork apache2-utils apache2.2-bin
apache2.2-common
  libapache2-mod-php5 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libonig2 libqdbm14 php5 php5-cli php5-common ssl-cert
0 upgraded, 16 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/7594 kB of archives.
After this operation, 23.5 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 8.2 Proses instllasi aplikasi untuk web server

Perhatikan gambar diatas, terlihat bahwa kita tidak hanya menginstall apalikasi apache saja, ada tambahan satu aplikasi yaitu php5. Aplikasi ini diinstall dengan tujuan agar web server kita bisa support php.

Selanjutnya kita harus melakukan konfigurasi virtual host. Konfigurasi virtual host nantinya akan mewakili setiap website yang dibuat. Jadi misalkan kita mempunyai dua website, maka kita harus membuat dua konfigurasi virtual host.

Pada sub bab ini kita akan membuat dua website, sehingga kita perlu melakukan konfigurasi dua buah virtual host. Untuk membuat virtual host, kita bisa mengcopy dari file contoh yang telah disediakan oleh apache. Lokasinya berada di direktori */etc/apache2/sites-available*.

```
root@forkits:~# cd /etc/apache2/sites-available/
root@forkits:/etc/apache2/sites-available# ls
default   default-ssl
root@forkits:/etc/apache2/sites-available# cp default www
root@forkits:/etc/apache2/sites-available# cp default web
root@forkits:/etc/apache2/sites-available# nano www
<VirtualHost *:80>
    ServerAdmin admin@forkits.com
    ServerName forkits.com
    ServerAlias www.forkits.com
    DocumentRoot /var/www/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/www>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
    ..........................................................................................
    ..........................................................
    ..................................................
root@forkits:/etc/apache2/sites-available# nano web
<VirtualHost *:80>
    ServerAdmin admin@forkits.com
    ServerName web.forkits.com
    DocumentRoot /var/www/web
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/web>
        Options Indexes FollowSymLinks MultiViews

        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
    ..........................................................................................
    ..........................................................
```

Gambar 8.3 Konfigurasi virtualhost

Berikut penjelasan beberapa perinah dan syntak diatas

Syntak	Deskripsi
cp default www	Perintah ini digunakan untuk membuat konfigurasi virtualhost untuk website www.forkits.com. Untuk membuatnya kita tinggal copy dari file contoh yang telah disediakan apache (default).
nano www	Digunakan untuk mengkonfigurasi virtualhost untuk website www.forkits.com
<VirtualHost *:80>	Menunjukkan bahwa web server berjalan di port 80
ServerAdmin admin@forkits.com	Menunjukkan alamat email yang harus dihubungi oleh client saat web server mengalami gangguan. Email ini milik administrator server
ServerName forkits.com	Menunjukkan website dari virtual host tersebut. Jadi virtualhost ini ditujukan untuk website forkits.com
ServerAlias www.forkits.com	Menunjukkan website alias dari virtual host tersebut. Jadi virtual host ini ditujukan untuk website forkits.com dan www.forkits.com. Nantinya kedua website ini akan mempunyai tampailan yang sama
DocumentRoot /var/www/www	Menunjukkan lokasi penyimpanan file-file website.
<Directory /var/www/www>	Menunjukkan lokasi penyimpanan file-file website.

Selanjutnya kita harus mengaktifkan konfigurasi virtualhost yang telah kita buat dengan perintah berikut

```
root@forkits:/etc/apache2/sites-available# a2dissite default
Site default disabled.
To activate the new configuration, you need to run:
  service apache2 reload
root@forkits:/etc/apache2/sites-available# a2ensite www web
Enabling site www.
Enabling site web.
To activate the new configuration, you need to run:
  service apache2 reload
root@forkits:/etc/apache2/sites-available#
```

Gambar 8.4 Mengaktifkan virtualhost

Perhatikan gambar diatas, perintah pertama digunakan untuk mennonaktifkan virtual host default. Karena kita tidak membutuhkan virtual host tersebut.

Perintah kedua digunakan untuk mengaktifkan virtualhost www dan web. Selanjutnya kita harus membuat web direktori yang dibutuhkan oleh kedua virtual host yang telah kita buat sebelumnya

```
root@forkits:/etc/apache2/sites-available# cd /var/www/
root@forkits:/var/www# mkdir www
root@forkits:/var/www# mkdir web
root@forkits:/var/www# nano www/index.html
<!DOCTYPE html>
<html>
<head>
    <title>www.forkits.com</title>
</head>
<body>
    <center><h1>Welcome to WWW.FORKITS.COM</h1></center>
</body>
</html>
root@forkits:/var/www# nano web/index.php
<?php
    echo "Welcome To WEB.FORKITS.COM";
?>
```

Gambar 8.5 Membuat file-file konfigurasi website di web directory

Perhatikan gambar diatas, terlihat bahwa kita membuat sebuah file html dengan nama index.html di web direktori www.forkits.com, dan sebuah file php dengan nama index.php di web direktori web.forkits.com. Jangan terlalu khawatir dengan perbedaan tersebut. Intinya saya hanya ingin memberitahu, bahwa kita bisa saja menggunakan file html ataupun php pada web server.

Disini kita harus sedikit-sedikit paham tentang bahasa pemrograman html dan php. Karena suatu saat seorang SysAdmin juga harus menguasai sebuah bahasa pemrograman, tidak cukup jika hanya bisa mengkonfigurasi server saja.

Langkah terahir yang perlu kita lakukan adalah restart service apahce.

```
root@forkits:/var/www# service apache2 restart
[ ok ] Restarting web server: apache2 ... waiting .
root@forkits:/var/www#
```

Gambar 8.6 Proses restart service web server

Sebelum melakukan pengujian, pastikan bahwa komputer client sudah bisa meresolve domain yang dimiliki oleh server. Telah disepakati sebelumnya, bahwa praktik pada bab ini mempunyai asumsi bahwa komputer server sudah dikonfigurasi dns server seperti pembahasan materi di bab 6.

```
admin@ubuntu:~$ nslookup forkits.com
Server:       192.168.10.1
Address:      192.168.10.1#53

Name:   forkits.com
Address: 192.168.10.1

admin@ubuntu:~$ nslookup www.forkits.com
Server:       192.168.10.1
Address:      192.168.10.1#53

Name:   www.forkits.com
Address: 192.168.10.1

admin@ubuntu:~$ nslookup web.forkits.com
Server:       192.168.10.1
Address:      192.168.10.1#53

web.forkits.com canonical name = www.forkits.com.
Name:   www.forkits.com
Address: 192.168.10.1

admin@ubuntu:~$
```

Gambar 8.7 Pengujian dns server dari komputer client

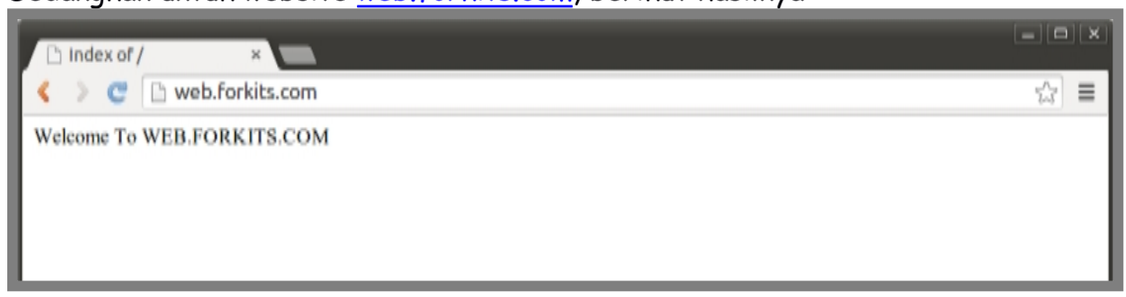
Berikut hasil pengujian saat mengakses website forkits.com

Gambar 8.8 Pengujian website forkits.com

Berikut hasil pengujian saat mengakses website www.forkits.com

Gambar 8.9 Pengujian website www.forkits.com

Sedangkan untuk website web.forkits.com, berikut hasilnya



Gambar 8.10 Pengujian website web.forkits.com

Konfigurasi Virtual Direktori

Pada umumnya, sebuah website mempunyai tampilan berbentuk html atau php. Namun ada beberapa website yang menampilkan kumpulan beberapa direktori dengan tujuan tertentu.

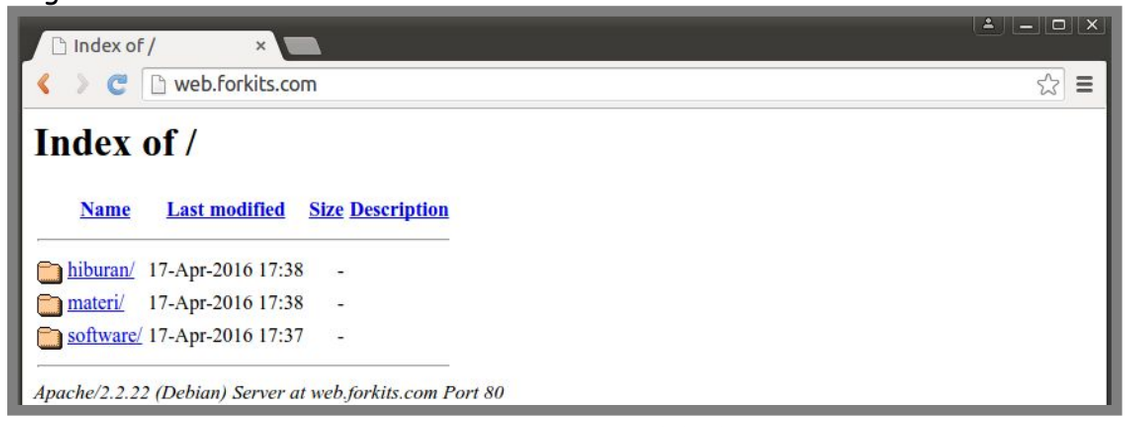
Sebagai contoh kasus, kita menginginkan agar website web.forkits.com mempunyai beberapa list direktori, yaitu software, materi, dan hiburan.

Untuk mengerjakan contoh kasus diatas, hal yang harus dilakukan adalah menghapus file index yang ada di web direktori dari website web.forkits.com, dan membuat list direktori yang diinginkan.

```
root@forkits:/var/www# rm web/index.php
root@forkits:/var/www# mkdir web/software
root@forkits:/var/www# mkdir web/materi
root@forkits:/var/www# mkdir web/hiburan
root@forkits:/var/www# ls web/
hiburan  materi  software
root@forkits:/var/www#
```

Gambar 8.11 Konfigurasi virtual direktori

Berikut hasil percobaan mengakses website web.forkits.com setelah melakukan langkah diatas



Gambar 8.12 Pengujian virtual direktori

Disable Signature Apache

Pada sub bab ini, kita akan membahas tentang keamanan web server paling dasar yang bisa kita lakukan. Yaitu mencegah web server memberikan informasi tentang sistem operasi server, aplikasi web server dan versi yang digunakan. Karena memberikan informasi tersebut ke orang lain akan mempermudah orang lain menentukan tool-tool yang bisa digunakan untuk membobol server kita.

Berikut contoh informasi yang tidak perlu ditampilkan

Gambar 8.13 Contoh informasi yang tidak perlu ditampilkan

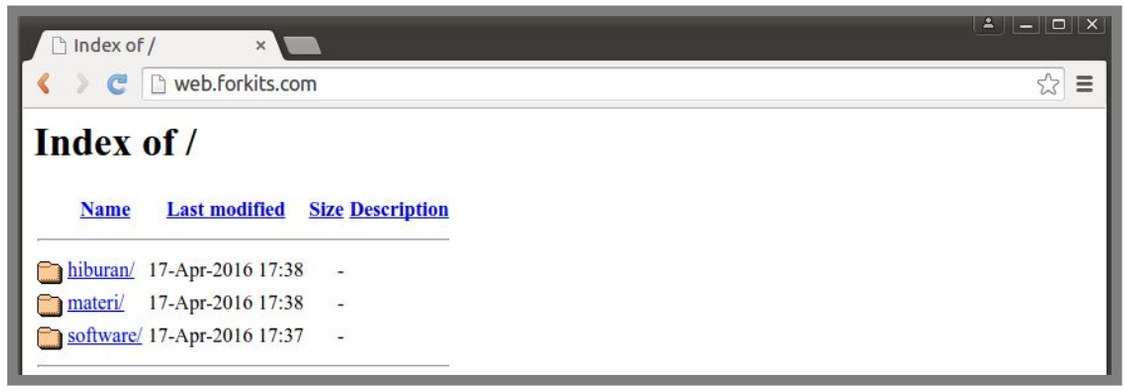
Untuk menghilangkan informasi yang tidak perlu diatas, berikut langkah-langkah yang bisa kita lakukan

```
root@forkits:~# nano /etc/apache2/apache2.conf
.....
.....
.....
# Include of directories ignores editors' and dpkg's backup files,
# see the comments above for details.

# Include generic snippets of statements
Include conf.d/

# Include the virtual host configurations:
Include sites-enabled/
ServerSignature Off
root@forkits:~# service apache2 restart
[ ok ] Restarting web server: apache2 ... waiting .
root@forkits:~#
```

Gambar 8.14 Merubah konfigurasi apache

Perhatikan gambar diatas, terlihat bahwa kita menambahkan satu baris konfigurasi di ahir file konfigurasi apache (apache2.conf). Selanjutnya kita merestart service apache.

Berikut hasil pengujian yang dilakukan setelah melakukan langkah diatas, telrihat bahwa saat ini tidak ada informasi mengenai web server yang ditampilkan di browser client



Gambar 8.15 Pengujian dari komputer client

Konfigurasi Virtual Webpages

Virtual webpages dimaksudkan untuk membuat sebuah website yang berbeda untuk masing-masing user. Hal ini adalah metode yang digunakan pada sebuah server hosting, dimana setiap user akan mempunyai sebuah website.

Diasumsikan bahwa di komputer server telah dikonfigurasi dns server dan juga web server seperti pembahasan sebelumnya. Untuk mengkonfigurasi virtual webpages bisa mengikuti langkah berikut

```
root@forkits:/var/www# a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
  service apache2 restart
root@forkits:/var/www# service apache2 restart
[ ok ] Restarting web server: apache2 ... waiting .
root@forkits:/var/www#
```

Gambar 8.16 Mengaktifkan modul userdir

Perhatikan gambar diatas, perintah pertama digunakan untuk mengaktifkan modul userdir yang digunakan untuk membuat virtual webpages. Perintah kedua digunakan untuk merestart service apache.

Hal yang harus dilakukan selanjutnya adalah membuat web direktori untuk virtual webpages tersebut. Web direktori sebuah virtual webpages berada di direktori *public_html/* yang berada di home direktori masing-masing user, oleh karena itu

kita harus membuat direktori *public_html* didalam home direktori masing-masing user.

```
root@forkits:/var/www# cd /home/forkits/
root@forkits:/home/forkits# mkdir public_html
root@forkits:/home/forkits# nano public_html/index.php
<?php

  echo "Welcome To FORKITS Virtual Webpages";

?>
root@forkits:/home/forkits# chown forkits:forkits public_html -R
root@forkits:/home/forkits#
```

Gambar 8.17 Konfigurasi web directory untuk userdir

Perhatikan gambar diatas, terlihat bahwa kita membuat sebuah file index.php didalam web direktori. Hal tersebut dimaksudkan untuk membuat sebuah halaman web seperti pada umumnya, namun jika kita menginginkan virtual webpages ini menampilkan list direktori, kita bisa menghapus file index.php tersebut dan membuat direktori yang diinginkan seperti yang telah dibahas di sub bab sebelumnya.

Perintah chown diatas bertujuan agar direktori public_html menjadi milik user dan group forkits. Hal ini dimaksudkan agar user forkits memiliki hak akses maksimal terhadap direktori tersebut

Saat ini, kita telah selesai mengkonfigurasi virtual webpages, kita bisa mengaksesnya dengan url_domain/~nama_user. Sebagai contoh, www.forkits.com/~forkits. www.forktis.com adalah domain yang dimiliki server, sedangkan forkits adalah sebuah user yang ada diserver.

Berikut hasil pengujian terhadap virtual webpages milik user forkits

Gambar 8.18 Pengujian dari client

Perhatikan gambar diatas, terlihat bahwa keluaran yang dihasilkan sama persis dengan kode php yang kita tulis di file index.php. Hal ini menunjukkan bahwa

virtual webpages belum support php. Berikut cara yang harus dilakukan agar virtual webpages bisa support php

```
root@forkits:/home/forkits# nano /etc/apache2/mods-enabled/php5.conf
.....
.....
.....
<IfModule mod_userdir.c>
    <Directory /home/*/public_html>
#           php_admin_value engine Off
    </Directory>
</IfModule>
root@forkits:/home/forkits#
```

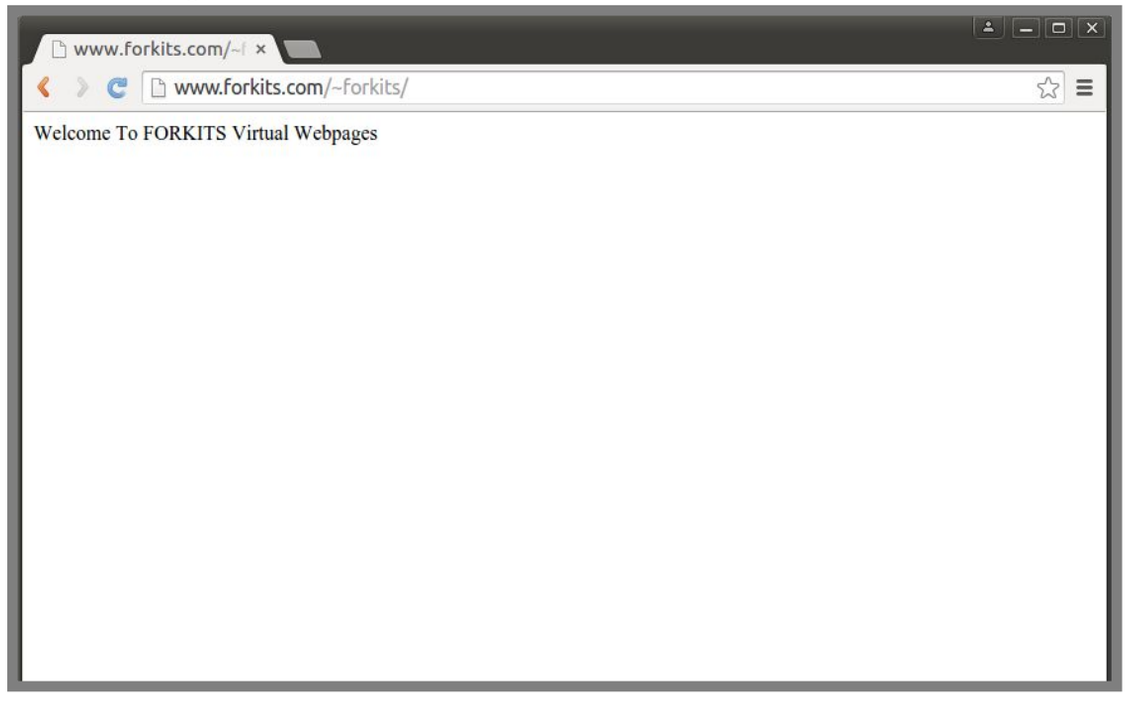
Gambar 8.19 Mengaktifkan dukungan php pada virtual webpages

Perhatikan gambar diatas, terlihat bahwa kita hanya melakukan perubahan pada baris nomer tiga dari bawah pada file tersebut. Selanjutnya restart service apache.

```
root@forkits:/home/forkits# service apache2 restart
[ ok ] Restarting web server: apache2 ... waiting ..
root@forkits:/home/forkits#
```

Gambar 8.20 Merestart service web server

Berikut hasil pengujian yang dilakukan setelah mengaktifkan modul php untuk virtual webpages.



Gambar 8.21 Pengujian yang dilakukan client

Konfigurasi Web Server Authentication

Ada kalanya kita ingin melindungi website dari orang-orang yang tidak berhak mengakses website. Salah satu cara yang paling mudah adalah melindungi website tersebut dengan password.

Sebagai contoh kasus, misal kita menginginkan agar jika ada orang yang membuka website www.forkits.com harus memasukkan username dan password. Untuk mengerjakan contoh kasus tersebut, berikut langkah-langkah yang perlu dilakukan

```
root@forkits:~# cd /etc/apache2/sites-available/
root@forkits:/etc/apache2/sites-available# nano www
.....
.....
.....
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/www>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride AuthConfig
        Order allow,deny
        allow from all
    </Directory>
.....
.....
.....
```

Gambar 8.22 Konfigurasi virtualhost untuk mendukung authentikasi

Perhatikan gambar diatas, terlihat bahwa kita melakukan sedikit perubahan pada file virtual host www.forkits.com. Selanjutnya kita harus membuat file .htaccess didalam web direktori www.forkits.com.

```
root@forkits:/etc/apache2/sites-available# cd /var/www/www/
root@forkits:/var/www/www# nano .htaccess
AuthUserFile /var/www/www/.htpasswd
AuthName "Silahkan Login Dengan User yang Falid"
AuthType Basic

<Limit GET POST>
require valid-user
</Limit>
```

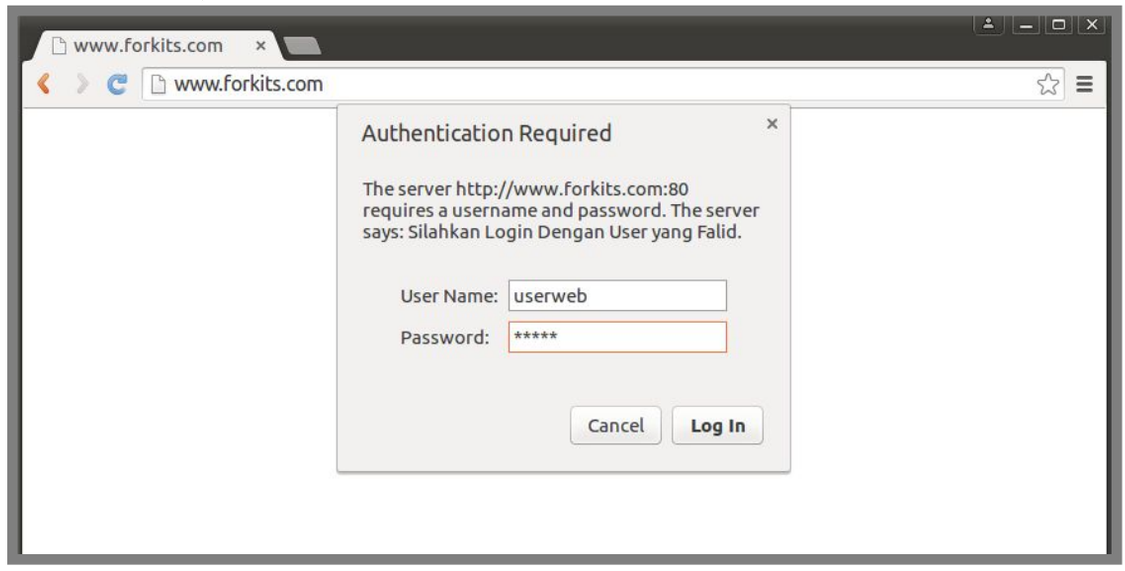
Gambar 8.23 Konfigurasi authentikasi pada web server

Selanjutnya kita harus membuat user dan password sesuai yang diinginkan. Berikut perintah yang digunakan untuk membuat user dan password

```
root@forkits:/var/www/www# htpasswd -c .htpasswd userweb
New password: (tidak terlihat)
Re-type new password: (tidak terlihat)
Adding password for user userweb
root@forkits:/var/www/www# service apache2 restart
[ ok ] Restarting web server: apache2 ... waiting .
root@forkits:/var/www/www#
```

Gambar 8.24 Membuat username dan password untuk web server

Perintah diatas digunakan untuk menambahkan user web dengan nama userweb, dilanjutkan dengan perintah untuk restart service apache. Berikut hasil pengujian yang dilakukan saat mengakses www.forkits.com setelah melakukan langkah-langkah diatas, terlihat bahwa client akan diminta untuk memasukkan username dan password



Gambar 8.25 Pengujian dari komputer client

Konfigurasi Web Server HTTPS

Sama halnya dengan web server yang telah kita bahas sebelumnya (http), https juga bertugas menampilkan sebuah halaman web pada web browser.

Perbedaan utama antara http dan https terletak pada tingkat keamanan yang dimiliki, dimana https adalah versi secure (lebih aman) dari http. Selain keamanan, perbedaan lain terletak pada port yang digunakan. Protocol http menggunakan port 80, sedangkan https menggunakan port 443.

Untuk melakukan konfigurasi https, sangat disarankan untuk memahami materi tentang ssl dan CA terlebih dahulu (bab 7). Jika belum terlalu faham mengenai

konsep ssl, bisa mengulangi membaca bab 7 sebelum melanjutkan materi https. Sebagai contoh kasus, kita menginginkan agar website www.forkits.com dan web.forkits.com bisa diakses melalui protocol https, yaitu https://www.forkits.com dan https://web.forkits.com.

Untuk mengerjakan contoh kasus tersebut, hal pertama yang harus dilakukan adalah membuat private key dan CSR. Kemudian mengirimkan CSR kepada CA dan menunggu sertifikat yang telah ditandatangani CA.

Pada sub bab ini hanya akan ditunjukkan cara membuat private key dan CSR. Untuk langkah selanjutnya, seperti penyetujuan CSR oleh CA dll tidak akan dibahas pada bab ini. Silahkan melihat kembali materi bab 7 jika belum bisa.

```
root@forkits:/var/www/www# mkdir /etc/apache2/ssl
root@forkits:/var/www/www# cd /etc/apache2/ssl/
root@forkits:/etc/apache2/ssl# openssl genrsa -aes256 2048 > forkits.key
Generating RSA private key, 2048 bit long modulus
.....................................................................+++
.....................................+++
e is 65537 (0x10001)
Enter pass phrase: (tidak terlihat)
Verifying - Enter pass phrase: (tidak terlihat)
root@forkits:/etc/apache2/ssl# openssl req -new -key forkits.key -out
forkits.csr
Enter pass phrase for forkits.key:(tidak terlihat)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ID
State or Province Name (full name) [Some-State]:Jawa Timur
Locality Name (eg, city) []:Blitar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Komunitas IT
Organizational Unit Name (eg, section) []:SysAdmin
Common Name (e.g. server FQDN or YOUR name) []:*.forkits.com
Email Address []:admin@forkits.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:(tidak terlihat)
An optional company name []:(tidak terlihat)
root@forkits:/etc/apache2/ssl#
```

Gambar 8.26 Pembuatan private key dan csr pada server

Perhatikan gambar diatas, terlihat bahwa domain yang kita masukkan adalah *.forkits.com, itu artinya seluruh subdomain dari forkits.com, entah itu www.forkits.com, ataupun web.forkits.com.

Langkah selanjutnya adalah mengirimkan CSR kepada CA dan menunggu sertifikat yang telah ditandatangani CA. Diasumsikan bahwa kita telah menerima sertifikat yang telah ditandatangani oleh CA dan telah kita simpan di direktori /etc/apache2/ssl dengan nama forkits.crt

```
root@forkits:/etc/apache2/ssl# ls
forkits.crt  forkits.csr  forkits.key
root@forkits:/etc/apache2/ssl#
```

Gambar 8.27 Melihat file private key dan sertifikat

Perhatikan gambar diatas, terlihat bahwa kita telah mempunya file private key (*forkits.key*) dan juga file sertifikat (*forkits.crt*) yang telah ditandatangani oleh CA. Selanjutnya kita harus mengaktifkan modul ssl

```
root@forkits:/etc/apache2/ssl# cd /etc/apache2/sites-available/
root@forkits:/etc/apache2/sites-available# a2enmod ssl
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to
configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
root@forkits:/etc/apache2/sites-available#
```

Gambar 8.28 Mengaktifkan modul ssl

Langkah selanjutnya kita harus membuat virtualhost ssl untuk https://www.forkits.com dan https://web.forkits.com. Berikut konfigurasi virtualhost untuk https://www.forkits.com

```
root@forkits:/etc/apache2/sites-available# cp default-ssl www-ssl
root@forkits:/etc/apache2/sites-available# nano www-ssl
NameVirtualHost 192.168.10.1:443
<IfModule mod_ssl.c>
<VirtualHost 192.168.10.1:443>
        ServerAdmin admin@forkits.com
        ServerName www.forkits.com
        DocumentRoot /var/www/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/www>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        .....
        .....
        .....
        .....
        .....
        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2.2-common/README.Debian.gz for more i$
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/apache2/ssl/forkits.crt
        SSLCertificateKeyFile /etc/apache2/ssl/forkits.key
        .....
        .....
        .....
        .....
        .....
```

Gambar 8.29 Konfigurasi virtualhost untuk https://www.forkits.com

Berikut konfigurasi virtualhost untuk https://web.forkits.com

```
root@forkits:/etc/apache2/sites-available# cp default-ssl web-ssl
root@forkits:/etc/apache2/sites-available# nano web-ssl
<IfModule mod_ssl.c>
<VirtualHost 192.168.10.1:443>
        ServerAdmin admin@forkits.com
        ServerName web.forkits.com
        DocumentRoot /var/www/web
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/web>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        .....
        .....
        .....
        .....
        .....
        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2.2-common/README.Debian.gz for more i$
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/apache2/ssl/forkits.crt
        SSLCertificateKeyFile /etc/apache2/ssl/forkits.key
        .....
        .....
        .....
        .....
        .....
```

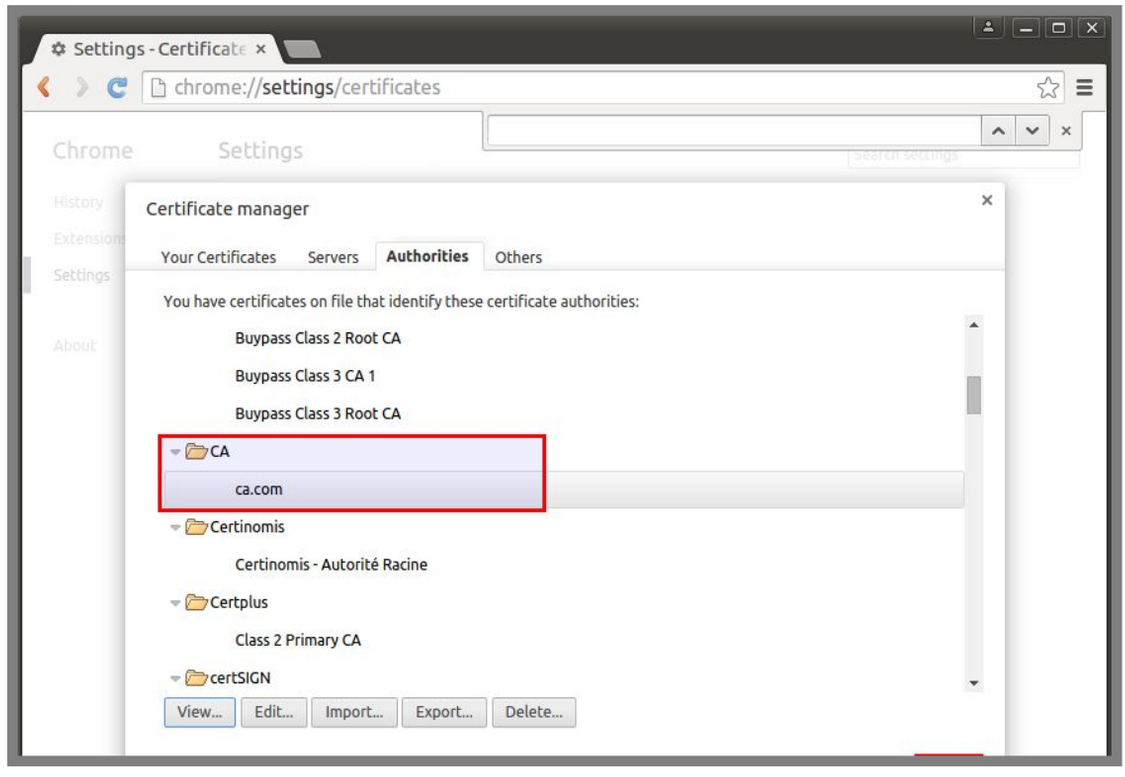
Gambar 8.30 Konfigurasi virtualhost untuk https://web.forkits.com

Selanjutnya aktifkan kedua virtual host tersebut dan restart service apache

```
root@forkits:/etc/apache2/sites-available# a2ensite www-ssl web-ssl
Enabling site www-ssl.
Enabling site web-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@forkits:/etc/apache2/sites-available# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server www.forkits.com:443 (RSA)
Enter pass phrase: (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:/etc/apache2/sites-available#
```

Gambar 8.31 Mengaktifkan virtualhost

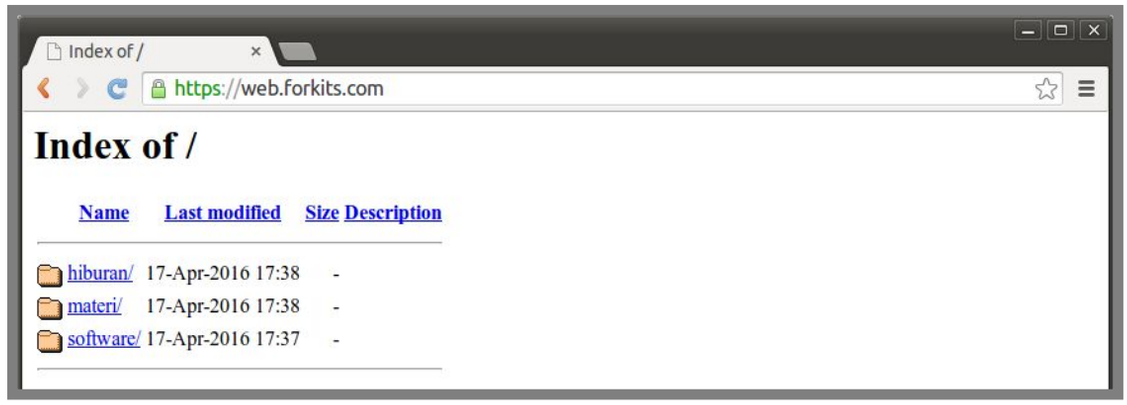
Sebelum melakuka pengujian, pastikan bahwa CA yang kita buat telah terdaftar sebagai CA terpercaya di web browser client



Gambar 8.32 Import sertifikat CA di client

Berikut pengujian saat mengakses www.forkits.com dan web.forkits.com menggunakan protocol https

Gambar 8.33 Pengujian website https://www.forkits.com



Gambar 8.34 Pengujian website https://web.forkits.com

Perhatikan dua gambar diatas, terlihat bahwa tidak ada peringatan/warning ssl yang terlihat. Hal ini menunjukkan bahwa CA yang kita buat sudah terpercaya.

Redirect HTTP to HTTPS

Pada sub bab ini kita akan membahas cara agar jika client mengakses website menggunakan protocol http, otomatis diredirect ke protocol https.

Sebagai contoh kasus, kita menginginkan agar jika ada yang mengakses http://web.forkits.com, otomatis diredirect ke https://web.forkits.com.

Berikut langkah kerja yang perlu dilakukan

```
root@forkits:/etc/apache2/sites-available# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  service apache2 restart
root@forkits:/etc/apache2/sites-available#root@forkits:/etc/apache2/sit
```

Gambar 8.35 Mengaktifkan modul rewrite

Perintah diatas digunakan untuk mengaktifkan modul rewrite, selanjutnya edit file virtual host web.forkits.com

```
root@forkits:/etc/apache2/sites-available# nano web
    .....
    .....
    .....
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
</VirtualHost>
```

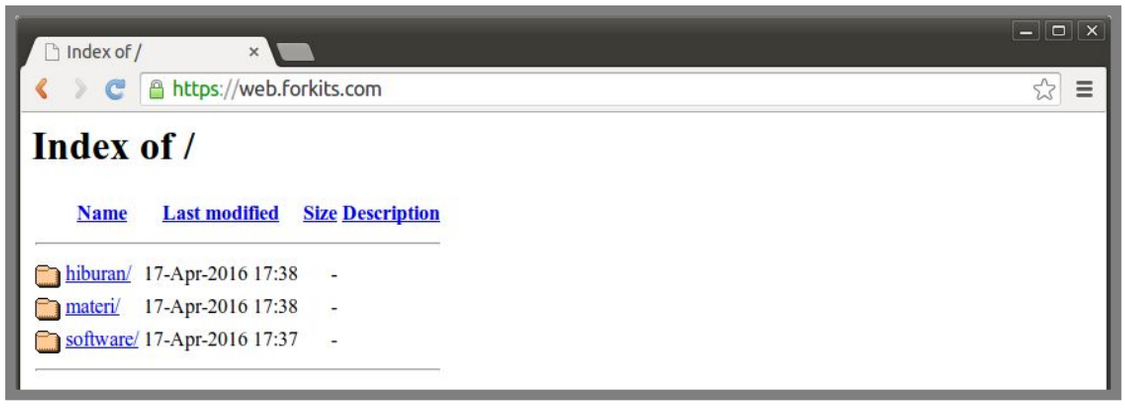
Gambar 8.36 Konfigurasi redirect http ke https

Perhatikan gambar diatas, terlihat bahwa kita menambahkan tiga baris konfigurasi (teks warna hijau) pada file virtual host web.forkits.com. Selanjutnya restart service apache

```
root@forkits:/etc/apache2/sites-available# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server www.forkits.com:443 (RSA)
Enter pass phrase: (tak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:/etc/apache2/sites-available#
```

Gambar 8.37 Merestart service web server

Selanjutnya lakukan pengujian dan buktikan bahwa setiap kita mencoba mengakses http://web.forkits.com maka akan diarahkan ke https://web.forkits.com secara otomatis



Gambar 8.38 Pengujian dari client

Installasi Database Server

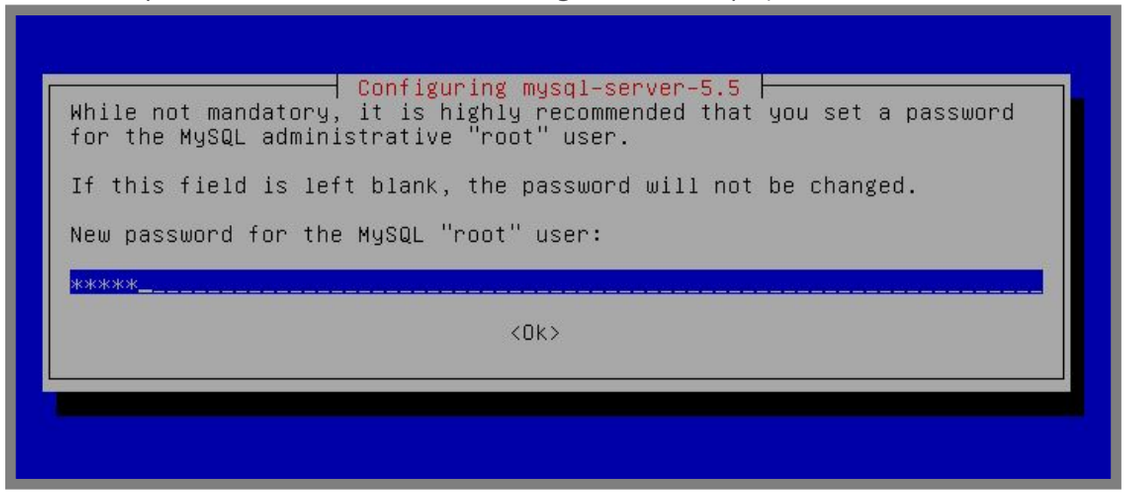
Telah dijelaskan diawal bab, bahwa database server adalah sebuah server yang bertugas menyimpan seluruh informasi yang berada di sebuah website.

Banyak aplikasi yang bisa kita manfaatkan untuk membuat database server. Salah satu yang paling populer adalah MySql. Pada bab inipun, kita akan menggunakan aplikasi MySql sebagai database server. Berikut perintah yang bisa kita gunakan untuk menginstall MySql

```
root@forkits:~# apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
  mysql-client-5.5 mysql-common mysql-server-5.5 mysql-server-core-5.5
Suggested packages:
  libipc-sharedcache-perl libterm-readkey-perl tinyca
The following NEW packages will be installed:
  libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
  mysql-client-5.5 mysql-common mysql-server mysql-server-5.5
  mysql-server-core-5.5
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/8478 kB of archives.
After this operation, 91.8 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

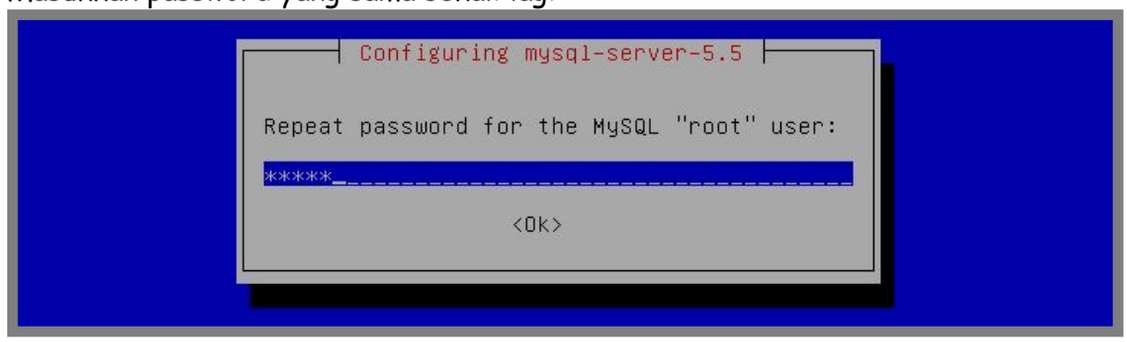
Gambar 8.39 Menginstall aplikasi database server

Masukkan password untuk user root sebagai user di mysql



Gambar 8.40 Konfigurasi password untuk user mysql

Masukkan password yang sama sekali lagi



Gambar 8.41 Konfigurasi password untuk user mysql

Berikut perintah untuk menggunakan mysql, serta beberapa perintah dasar yang berkaitan dengan database

```
root@forkits:~# mysql -u root -p
Enter password: (tidak terlihat)
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.5.37-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE kits;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+--------------------------+
| Database                 |
+--------------------------+
| information_schema       |
| kits                     |
| mysql                    |
| performance_schema       |
+--------------------------+
4 rows in set (0.00 sec)

mysql> USE kits;
Database changed
```

Gambar 8.42 Penggunaan perintah dasar mysql

Berikut penjelasan dari masing-masing perintah diatas

Syntak	Deskripsi
mysql -u root -p	Digunakan untuk login ke mysql dengan user root
CREATE DATABASE kits;	Digunakan untuk membuat database dengan nama kits
SHOW DATABASES;	Digunakan untuk melihat database yang ada di mysql
USE kits;	Digunakan untuk menggunakan database kits

Selanjutnya, berikut perintah-perintah yang berkaitan dengan tabel pada database yang biasa digunakan

```
mysql> USE kits;
Database changed
mysql> CREATE TABLE biodata(
    -> id varchar(4) NOT NULL,
    -> nama varchar(30) NOT NULL,
    -> alamat text
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> SHOW TABLES;
+---------------------+
| Tables_in_kits      |
+---------------------+
| biodata             |
+---------------------+
1 row in set (0.01 sec)

mysql> INSERT INTO biodata VALUES ('0001','Ahmad Rosid','Nglegok');
Query OK, 1 row affected (0.06 sec)

mysql> SELECT * FROM biodata;
+--------+----------------+-------------+
| id     | nama           | alamat      |
+--------+----------------+-------------+
| 0001   | Ahmad Rosid    | Nglegok     |
+--------+----------------+-------------+
1 row in set (0.01 sec)

mysql> QUIT;
Bye
root@forkits:~#
```

Gambar 8.43 Penggunaan perintah dasar mysql

Berikut penjelasan dari masing-masing perintah diatas

Syntak	Deskripsi
CREATE TABLE biodata.....	Digunakan untuk membuat sebuah tabel dengan nama biodata dan mempunyai kolom id, nama, dan alamat.
SHOW TABLES;	Digunakan untuk melihat tabel apa saja yang ada didatabase
INSERT INTO biodata.....	Digunakan untuk menambahkan data pada tabel biodata
SELECT * FROM biodata	Digunakan untuk melihat isi tabel biodata
QUIT;	Digunakan untuk keluar dari mysql

Masih banyak perintah-perintah yang bisa kita gunakan pada mysql. Teman-teman bisa mencari buku yang husus membahas materi mysql jika ingin lebih menguasai materi tentang mysql.

Installasi PhpMyAdmin

Tentu akan sangat merepotkan jika kita harus menghafal seluruh perintah yang ada pada mysql. Untuk itu kita bisa memanfaatkan sebuah aplikasi yang bernama phpmyadmin.

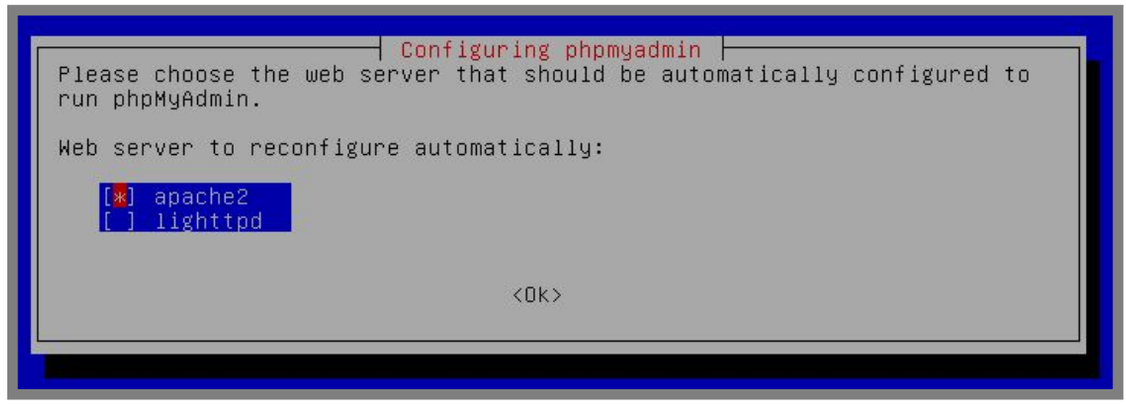
PhpMyAdmin adalah sebuah aplikasi berbasi web yang bisa kita manfaatkan untuk meremote mysql. Jadi kita bisa membuat database, membuat table, dan input ke table menggunakan aplikasi berbasis web.

Untuk menginstall aplikasi ini, komputer server harus sudah terinstall aplikasi webserver, yaitu apache dan php5. Berikut perintah yang digunakan untuk menginstall aplikasi phpmyadmin

```
root@forkits:~# apt-get install phpmyadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
.....
.....
The following NEW packages will be installed:
  dbconfig-common fontconfig-config libfontconfig1 libgd2-xpm libjpeg8
  libltdl7 libmcrypt4 libpng12-0 libxpm4 php5-gd php5-mcrypt php5-mysql
  phpmyadmin ttf-dejavu-core
0 upgraded, 14 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/8689 kB of archives.
After this operation, 22.3 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

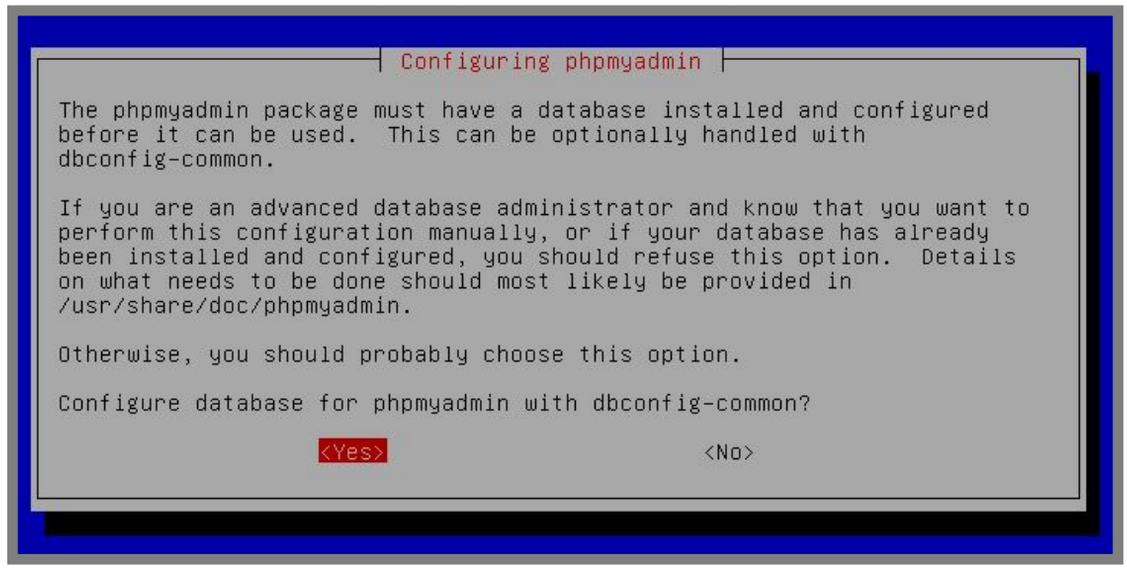
Gambar 8.44 Menginstall phpmyadmin

Pilih web server yang kita gunakan. Ingat bahwa pada pembahasan sebelumnya kita menggunakan aplikasi apache sebagai web server



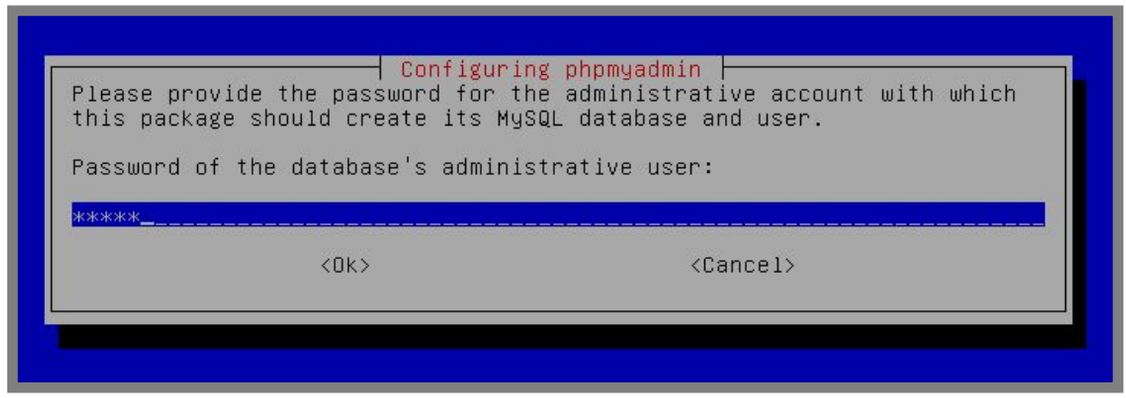
Gambar 8.45 Pemilihan web server yang digunakan

PHPMyAdmin mengharuskan kita untuk membuat dan mengkonfigurasi sebuah database sebelum phpmyadmin dapat digunakan. Pilih yes agar phpmyadmin otomatis membuat dan mengkonfigurasi database yang dibutuhkan



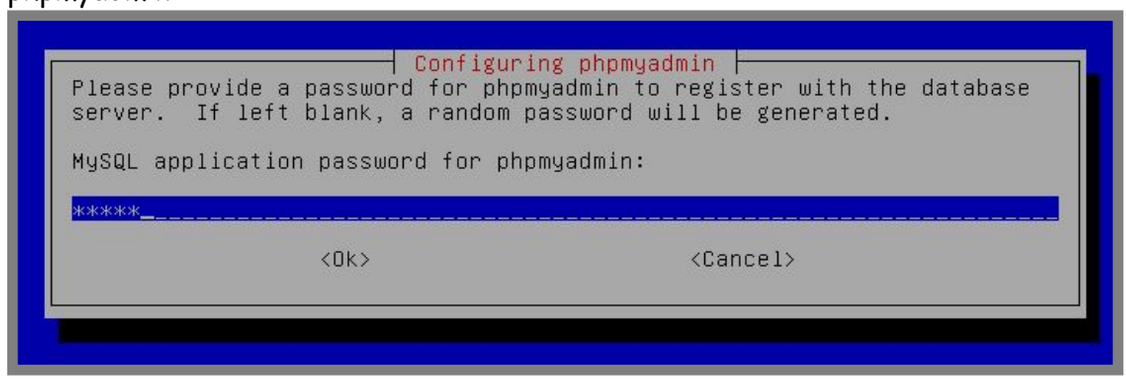
Gambar 8.46 Menyetujui pembuatan database untuk phpmyadmin

Karena phpmyadmin akan membuat sebuah database di mysql, maka kita diminta untuk memasukkan password dari user root untuk login ke mysql



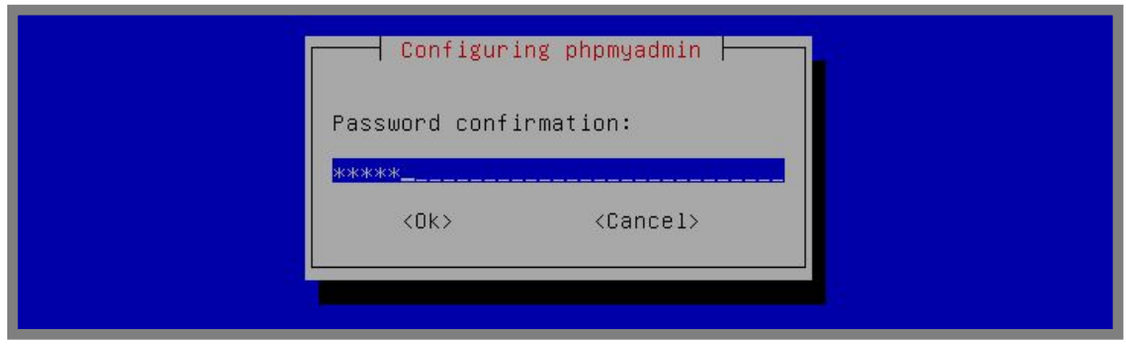
Gambar 8.47 memasukkan password untuk user root di mysql

Selanjutnya masukkan password yang ingin kita gunakan ketika login ke phpmyadmin



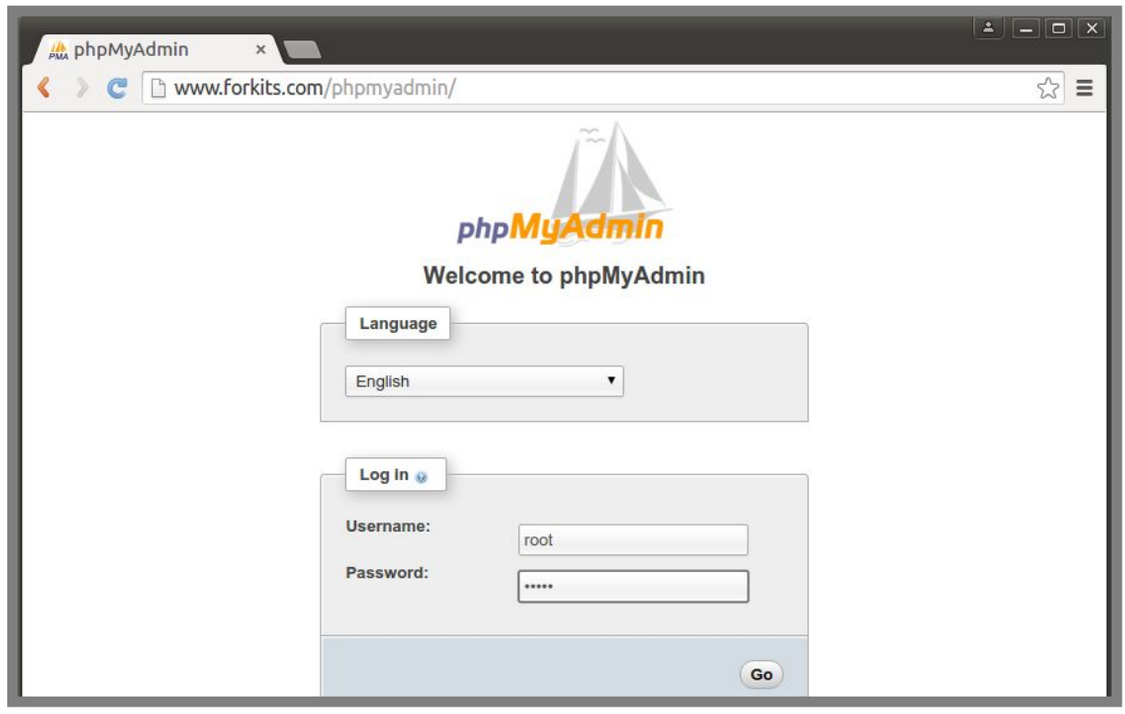
Gambar 8.48 Konfigurasi password untuk database phpmyadmin

Masukkan password yang sama sekali lagi untuk melakukan konfirmasi



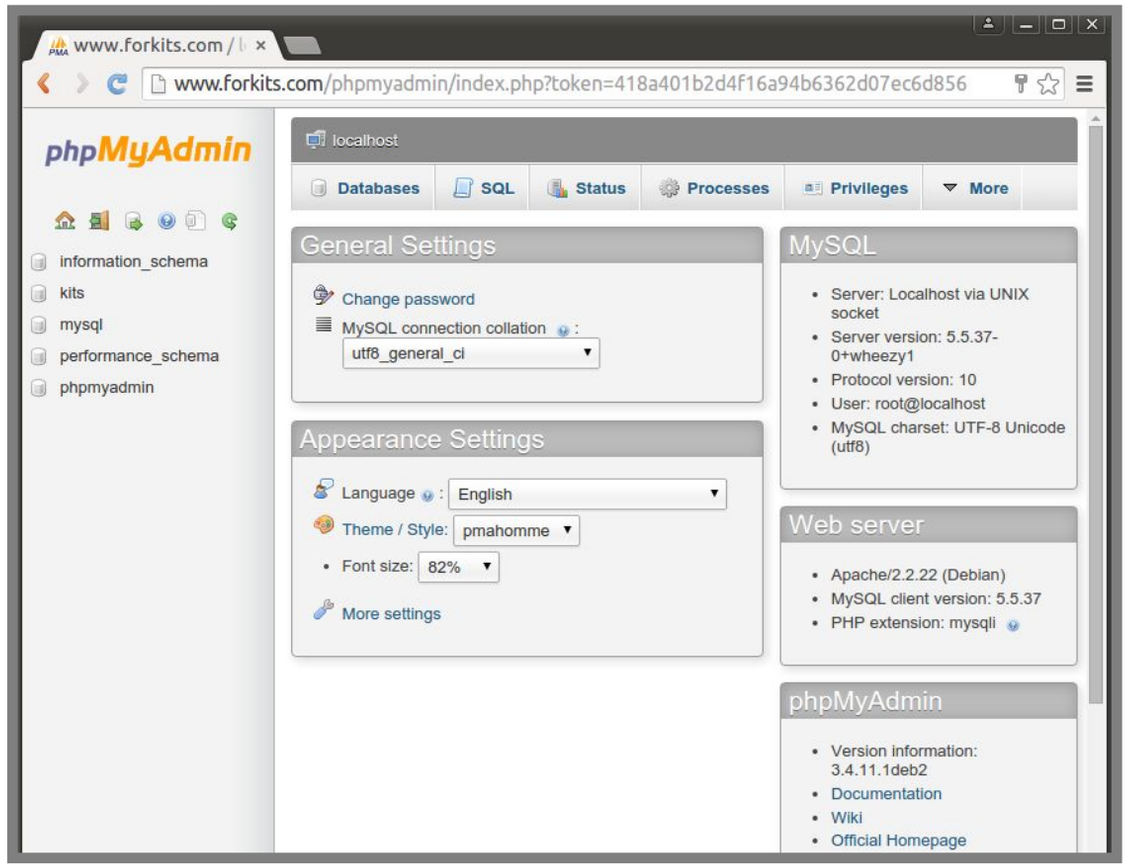
Gambar 8.49 Konfigurasi password untuk database phpmyadmin

Setelah selesai install phpmyadmin, kita bisa membuka aplikasi phpmyadmin dengan url *domain/phpmyadmin.* Berikut hasil pengujian yang dilakukan



Gambar 8.50 Pengujian phpmyadmin dari client

Berikut tampilan halaman depan aplikasi phpmyadmin. Untuk melakukan pekerjaan yang berkaitan dengan database dan tabel, kita hanya perlu bermain dengan mouse (klick and klick).



Gambar 8.51 Halaman utama phpmyadmin

Merubah URL phpMyAdmin

Secara default, phpmyadmin dapat diakses dengan url *domain/phpmyadmin.* Kita tentunya menyadari, bahwa semua orang pasti juga tahu kalau url untuk mengakses phpmyadmin adalah seperti itu. Untuk alasan keamanan, kita diharuskan mengganti url tersebut.

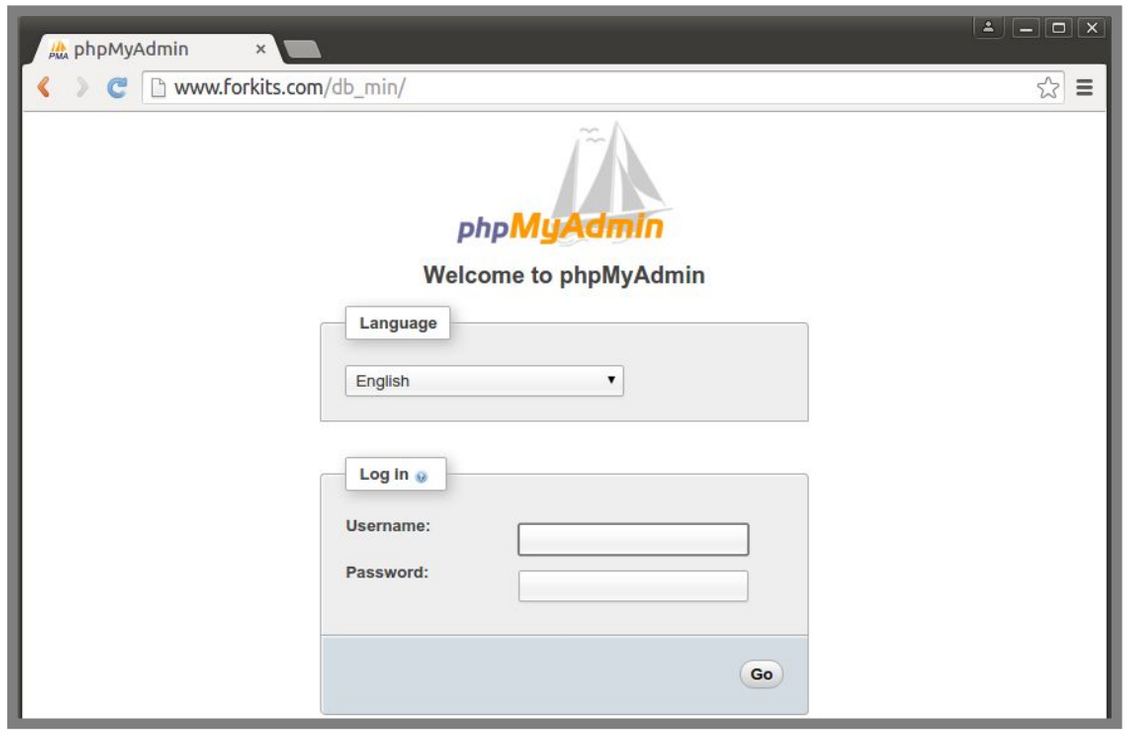
Sebagai contoh kasus, misalkan kita menginginkan agar phpmyadmin dapat diakses menggunakan url *domain/db_min.*

Berikut langkah-langkah yang perlu dilakukan untuk mengerjakan contoh kasus diatas.

```
root@forkits:~# nano /etc/phpmyadmin/apache.conf
# phpMyAdmin default Apache configuration

Alias /db_min /usr/share/phpmyadmin
.....................................................................
.....................................................................
.....................................................
root@forkits:~# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server www.forkits.com:443 (RSA)
Enter pass phrase:  (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:~#
```

Gambar 8.52 Konfigursi url phpmyadmin

Berikut pengujian mengakses phpmyadmin setelah melakukan langkah diatas



Gambar 8.53 Pengujian dari client

Installasi CMS Joomla

Content Management System (CMS) adalah sebuah aplikasi website. Jika kita ingat, website yang kita buat pada sub bab sebelumnya hanya mempunyai sebuah fil index (entah itu html atau php) dengan tampilan yang sangat sederhana.

Tentu kita tidak menginginkan website kita dipublikasikan dengan tampilan yang sangat sederhana seperti itu. Kita tentu menginginkan agar website kita mempunyai tampilan yang menarik, mempunyai informasi yang lengkap, artikel yang tidak membosankan, dan juga tingkat keamanan yang tinggi.

Untuk membuat sebuah website dengan kriteria yang disebutkan diatas tidaklah mudah. Kita harus coding mulai dari awal, mulai koding front end (html, css, javascript) hingga coding untuk bagian back end (php, mysql).

Namun saat ini telah ada sebuah aplikasi yang dinamakan CMS. Untuk membuat sebuah website yang berpenampilan menarik, kita hanya tinggal install CMS tersebut didalam website kita. Banyak terdapat CMS open source di internet, seperti joomla, wordpress, phpbb, mybb, drupal, dll.
Pada sub bab ini secara husus akan menggunakan joomla sebagai CMS. Kita bisa download joomla dari website resminya di www.joomla.org

Gambar 8.54 Download installser joomla

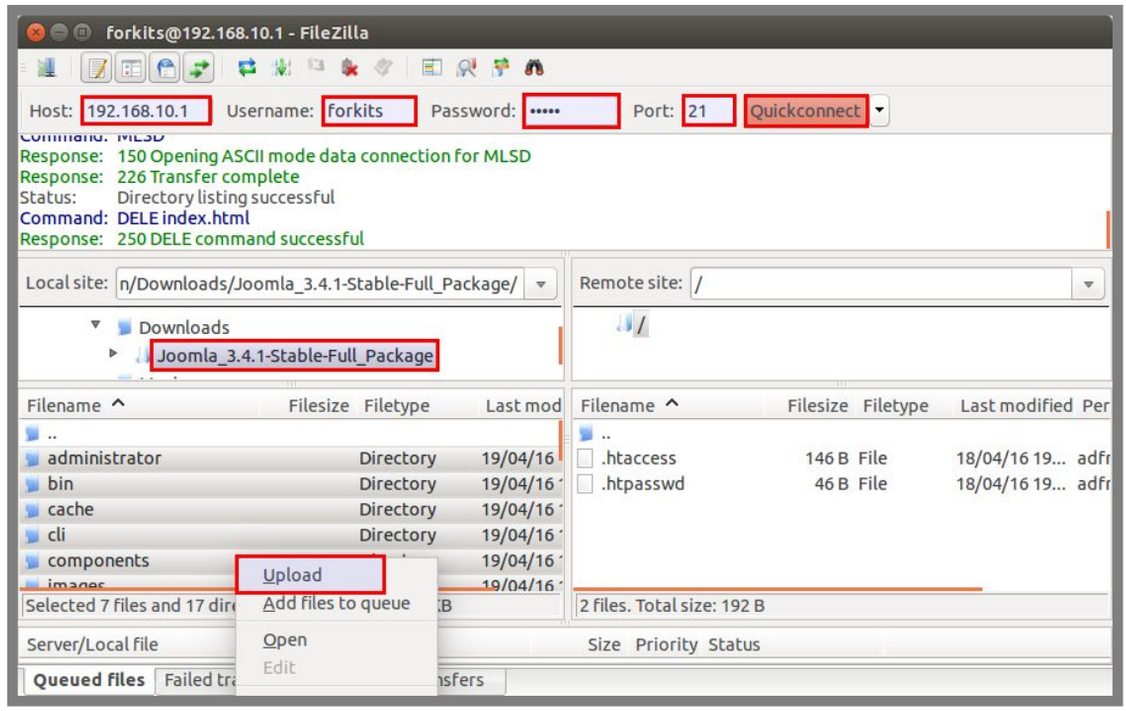
Setelah selesai download file tersebut, hal selanjutnya yang harus kita lakukan adala mengextract file tersebut, kemudian menguploadnya ke web server.

Untuk upload ke server, kita bisa memanfaatkan SFTP yang telah dibahas pada bab 5, namun ada baiknya kita upload menggunakan FTP. Karena pada umumnya, untuk melakukan upload file-file web ke server, kita menggunakan FTP. (Silahkan baca dulu bab 9 sebelum melanjutkan materi pada sub bab ini).

Diasumsikan bahwa kita telah menginstall dan mengkonfigurasi ftp server. Diasumsikan juga bahwa user forkits adalah admin untuk website www.forkits.com, sehingga saat user forkits login ftp, akan diarahkan ke web direktori dari www.forkits.com.

Untuk melakukan konfigurasi ftp seperti yang diasumsikan diatas, silahkan membaca materi pada bab 9. Jika asumsi diatas telah terpenuhi, kita bisa upload file joomla yang telah diextract ke server, namun sebelum upload ke server, pastikan bahwa tidak ada file index (entah itu index.html ataupun index.php) didalam web direktori. Jika ada hapus terlebih dahulu

Berikut hal-hal yang harus dilakukan untuk melakukan upload file joomla ke server. Pastikan bahwa yang diupload adalah isi dari file joomla tersebut. Jangan sampai file-file tersebut masih terbungkus suatu direktori.



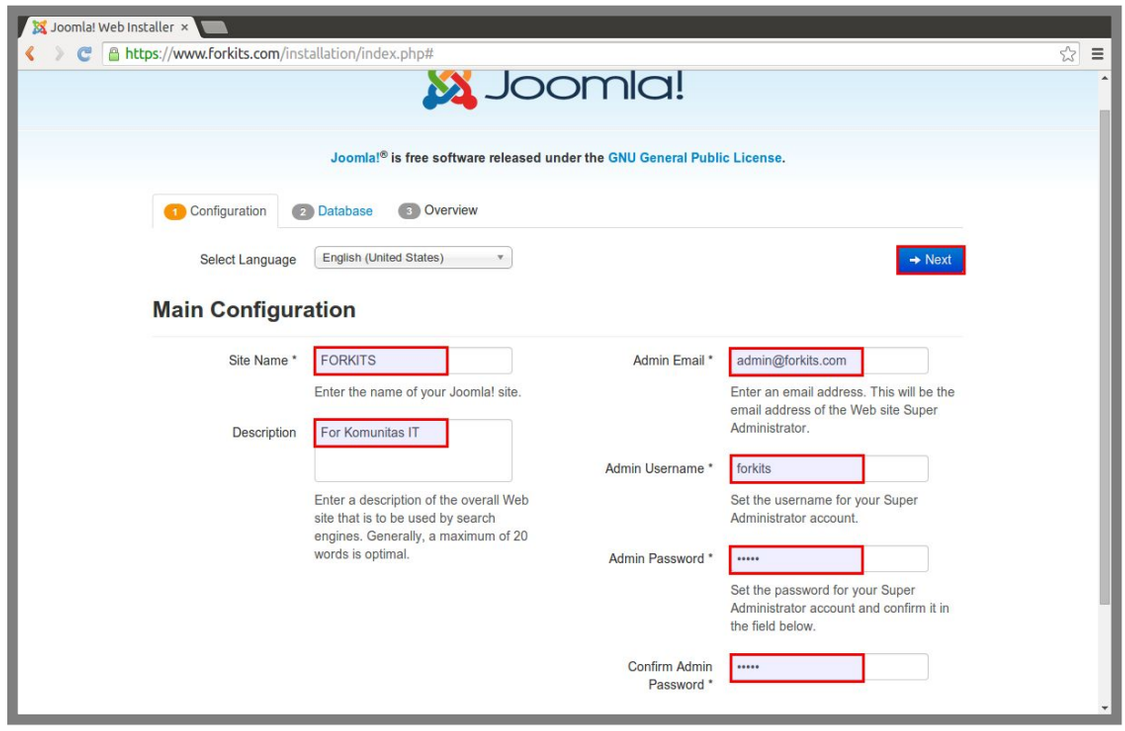
Gambar 8.55 Upload file joomla ke web server

Selanjutnya pastikan bahwa file-file yang kita upload tersebut telah berada didalam web direktori dari www.forkits.com dan rubah hak aksesnya

```
root@forkits:~# ls /var/www/www/
LICENSE.txt   cache       images      joomla.xml  logs       robots.txt.dist
README.txt    cli         includes    language    media      templates
administrator components  index.php   layouts     modules    tmp
bin           htaccess.txt installation libraries  plugins    web.config.txt
root@forkits:~# chmod 755 /var/www/www/
root@forkits:~#
```

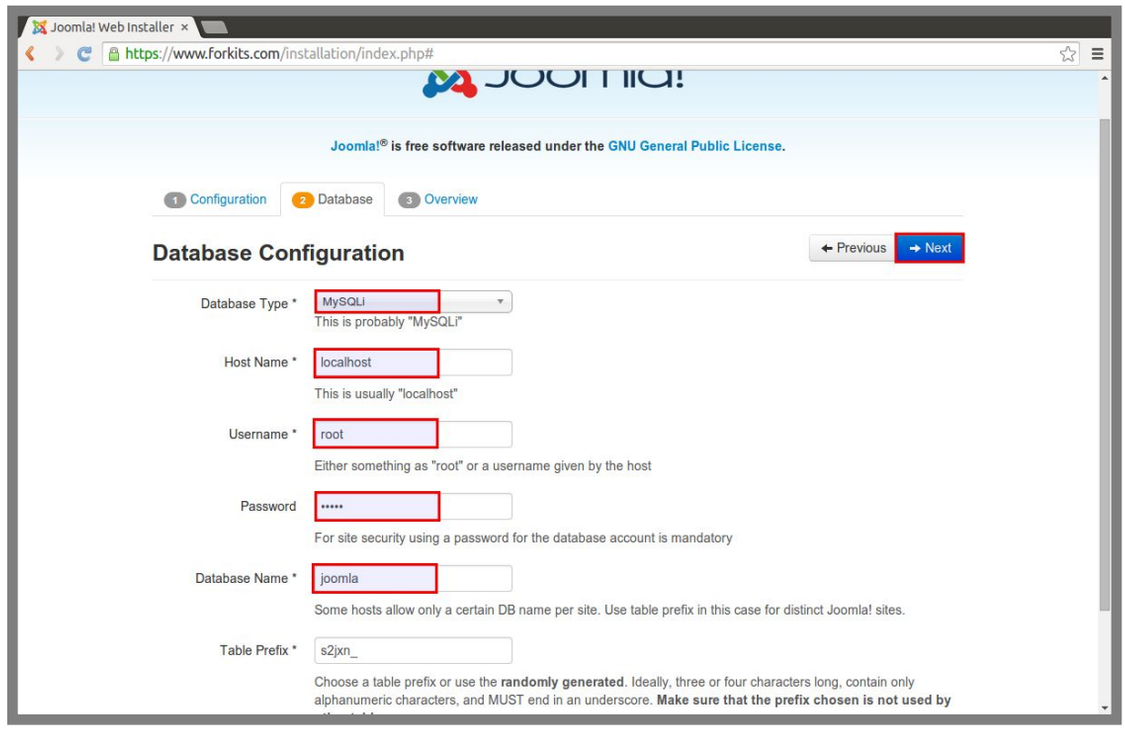
Gambar 8.56 Merubah hak akses file joomla

Untuk melakukan installasi joomla, gunakan web browser untuk membuka url https://www.forkits.com



Gambar 8.57 Proses installasi joomla

Isikan informasi yang diinginkan pada gambar diatas sesuai dengan informasi masing-masing. Selanjutnya kita diminta untuk mengisikan informasi yang berkaitan dengan database.



Gambar 8.58 Proses installasi joomla

Sebelum melanjutkan proses pada gambar diatas, kita harus membuat sebuah database sesuai dengan database yang kita masukkan pada gambar diatas.

Terlihat bahwa nama database yang kita masukkan adalah *joomla*. Maka dari itu kita harus membuat sebuah database dengan nama joomla

```
root@forkits:~# mysql -u root -p
Enter password: (tak terlihat)
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 5.5.37-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE joomla;
Query OK, 1 row affected (0.05 sec)

mysql> QUIT;
Bye
root@forkits:~#
```

Gambar 8.59 Membuat sebuah database

Setelah membuat database, kita bisa melanjutkan proses pada gambar 8.58.

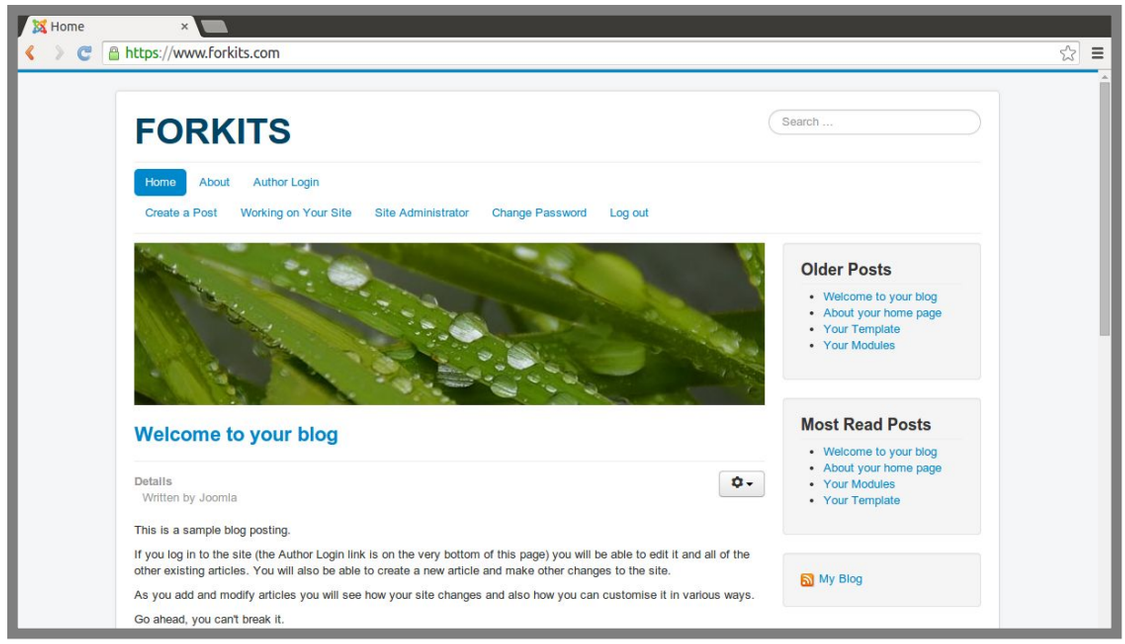
Gambar 8.60 Proses installasi joomla

Setelah proses installasi selesai, kita akan diminta menghapus direktori installation. Berikut perintah yang bisa kita gunakan untuk menghapus direktori tersebut

```
root@forkits:~# rm -rf /var/www/www/installation/
root@forkits:~#
```

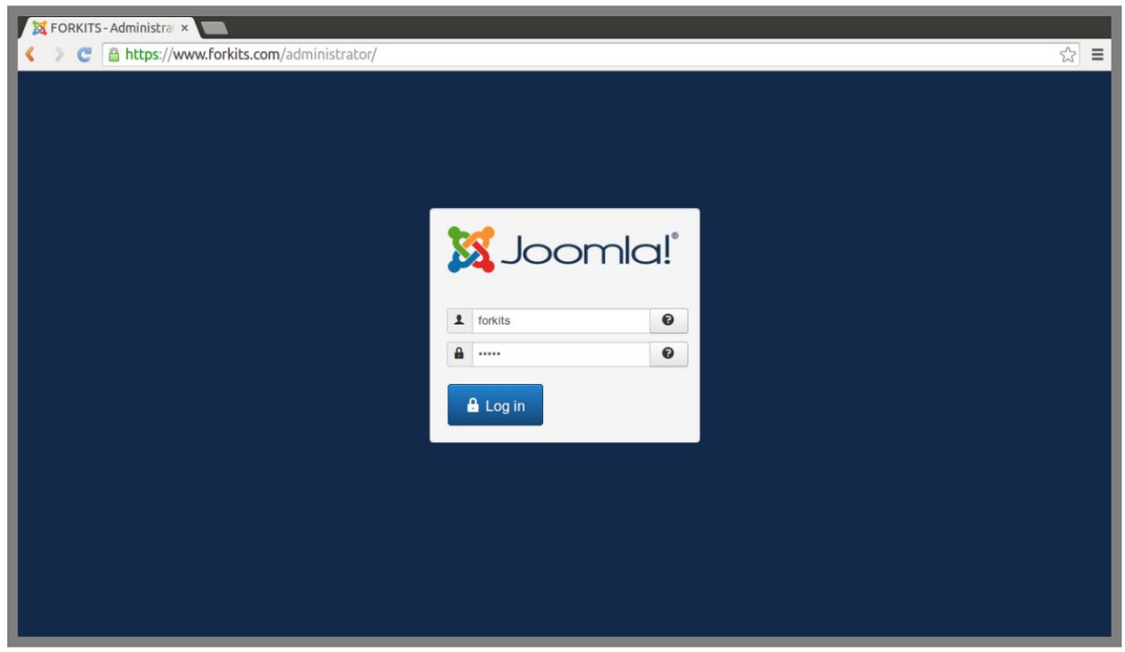
Gambar 8.61 Menghapus direktori *installation* milik joomla

Setelah menghapus direktori *installation*, coba buka lagi https://www.forkits.com



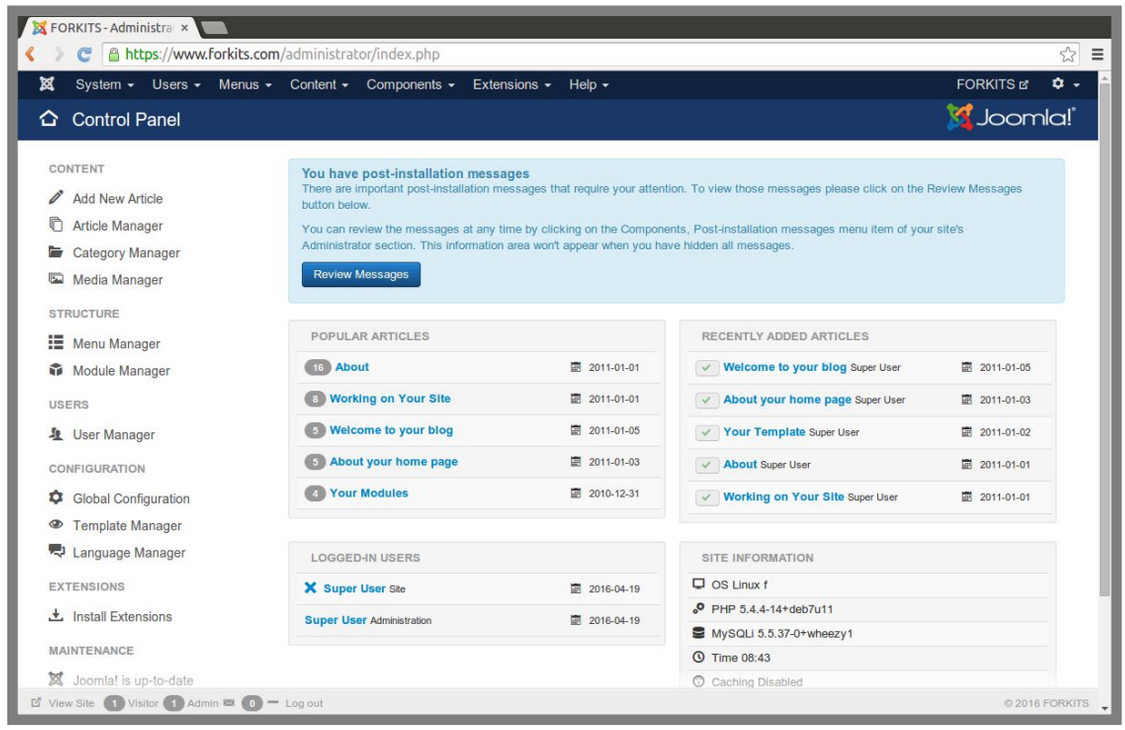
Gambar 8.62 Halaman utama joomla

Untuk melakukan konfigurasi joomla, kita bisa login sebagai administrator di url https://www.forkits.com/administrator



Gambar 8.63 Halaman login administrator joomla

Berikut halaman utama control panel dari joomla, disini kita bisa melakukan konfigurasi pada website joomla, mulai dari membuat postingan artikel, hingga merubah tampilan dari website joomla.



Gambar 8.64 Halaman utama administrator joomla

---END OF CHAPTER---

Bab 9
File Transfer Protocol Server

File transfer protocol (FTP) adalah salah satu protocol yang dapat digunakan untuk transfer file atau data melalui jaringan. Protocol ini sering digunakan untuk melakukan transfel file-file web ke web server.

Pada bab ini, seluruh konfigurasi akan mengacu pada topologi jaringan yang sama dengan topologi yang digunakan pada bab web & database server (gambar 8.1).

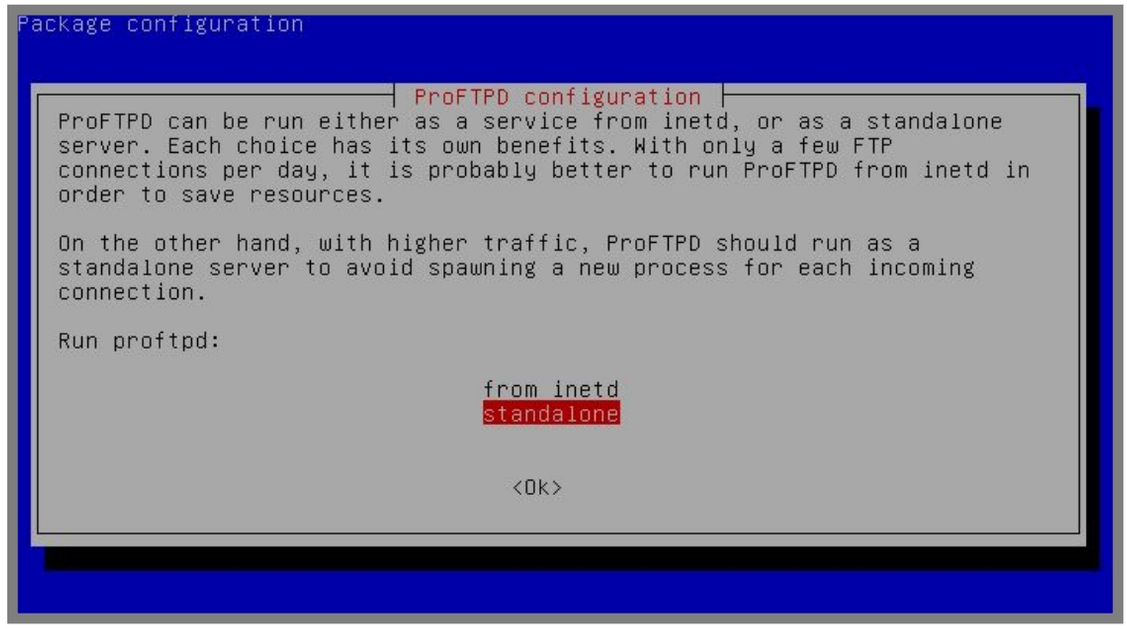
Konfigurasi FTP Server

Terdapat beberapa aplikasi yang bisa kita manfaatkan untuk membuat ftp server, salah satu aplikasi yang sangat populer adalah proftpd. Berikut perintah yang dapat kita gunakan untuk menginstall aplikasi proftpd

```
root@forkits:~# apt-get install proftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'proftpd-basic' instead of 'proftpd'
The following extra packages will be installed:
  proftpd-mod-vroot
Suggested packages:
  proftpd-doc proftpd-mod-ldap proftpd-mod-mysql proftpd-mod-odbc
  proftpd-mod-pgsql proftpd-mod-sqlite
The following NEW packages will be installed:
  proftpd-basic proftpd-mod-vroot
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/2537 kB of archives.
After this operation, 4131 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

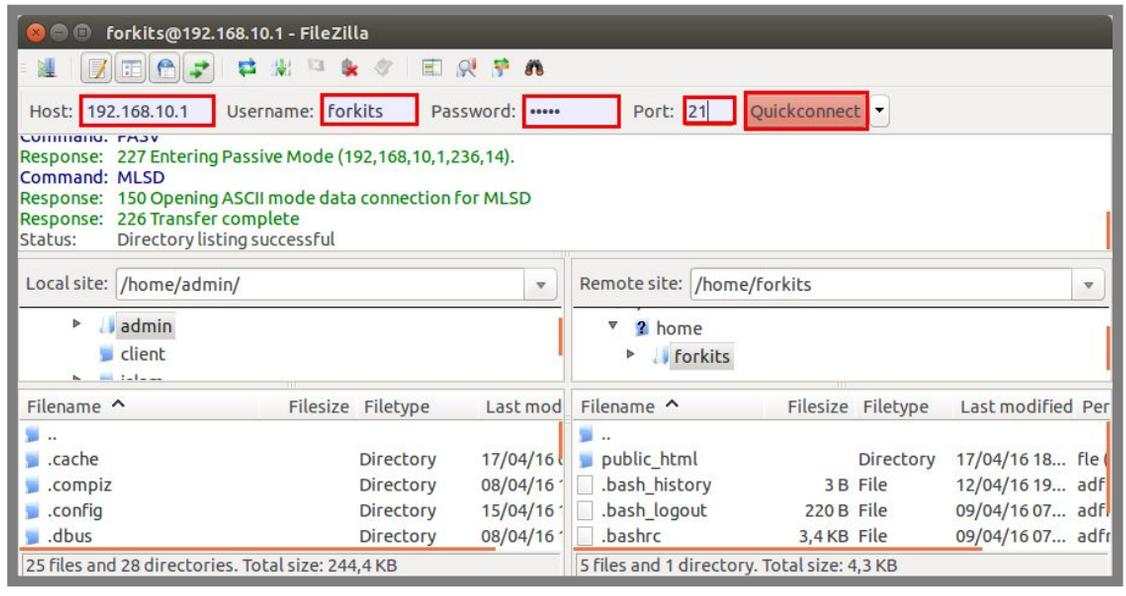
Gambar 9.1 Installasi proftpd

Jika nantinya ftp server ditugaskan untuk menangani trafik yang besar, disarankan kita memilih standalone.



Gambar 9.2 Proses installasi proftpd

Sampai saat ini kita telah berhasil install ftp server menggunakan proftpd. Selanjutnya untuk melakukan pengujian kita bisa login ke ftp server menggunakan aplikasi filezilla. Kita bisa login menggunakan salah satu user yang ada di komputer server



Gambar 9.3 Pengujian ftp menggunakan filezilla

Perhatikan gambar diatas, terlihat bahwa saat kita login ftp, maka otomatis akan diarahkan ke home direktori user masing-masing.

Konfigurasi FTP Root Direktori

Pembahasan pada sub bab ini dimaksudkan untuk merubah default direktori saat user login ke server melalui ftp. Seperti yang kita lihat pada hasil pengujian ftp sebelumnya, terlihat bahwa setiap user akan diarahkan ke home direktori masing-masing setiap kali login menggunakan ftp.

Jika sebuah web server dikonfigurasi virtual webpages (lihat materi bab 8), maka kita harus mengkonfigurasi ftp agar setiap kali user login menggunakan ftp, otomatis diarahkan ke direktori public_html yang berada didalam direktori masing-masing user.

Untuk melakukan hal tersebut, berikut langkah yang perlu dilakukan

```
root@forkits:~# nano /etc/proftpd/proftpd.conf
.....
.....
# Use this to jail all users in their homes
DefaultRoot             ~/public_html
# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
.....
.....
root@forkits:~# service proftpd restart
[ ok ] Stopping ftp server: proftpd.
[....] Starting ftp server: proftpdforkits proftpd[2951]:
mod_tls_memcache/0.1: notice: unable to register 'memcache' SSL session
cache: Memcache support not enabled
. ok
root@forkits:~#
```

Gambar 9.4 Konfigurasi default directory proftpd

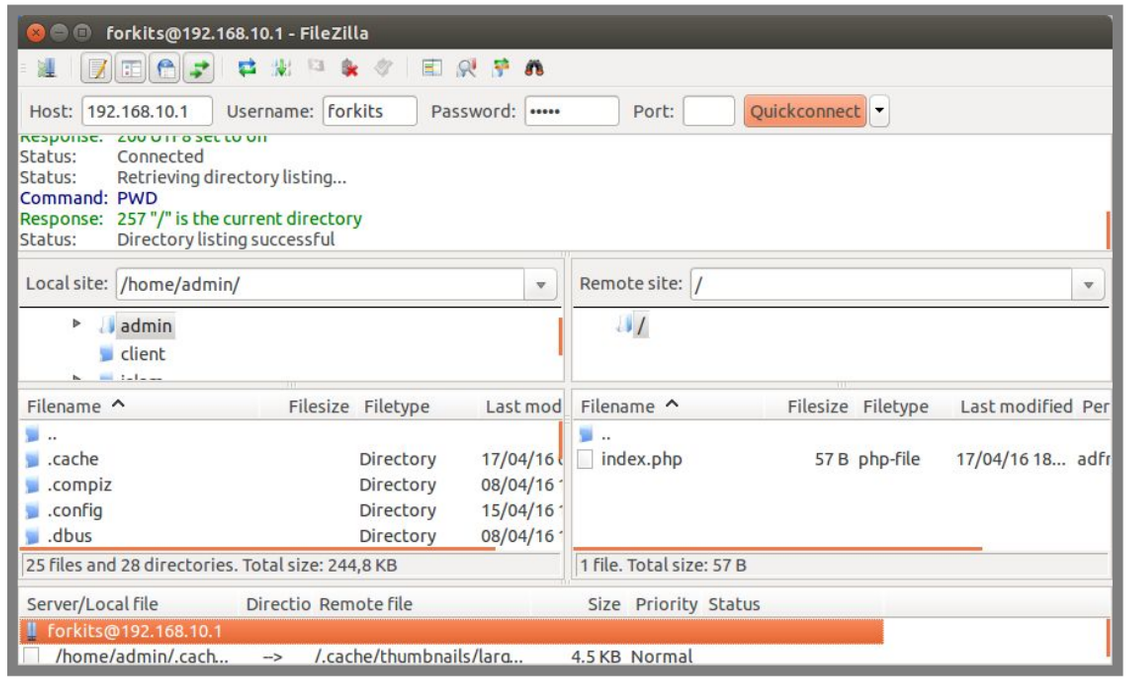
Perhatikan gambar diatas, terlihat bahwa kita melakukan perubahan pada file konfigurasi proftpd (proftpd.conf), kemudian dilanjutkan dengan merestart service profptd.

Berikutnya kita akan melakukan pengujian login ke ftp menggunakan user forkits. Namun sebelumnya kita akan melihat isi direktori public_html yang berada di home direktori forkits.

```
root@forkits:~# ls /home/forkits/public_html/
index.php
root@forkits:~#
```

Gambar 9.5 Isi directory public_html user forkits

Berikut hasil pengujian saat login menggunakan user forkits, perhatikan bahwa isi root direktorinya sama dengan isi direktori public_html yang telah kita lihat sebelumnya (gambar 9.5).



Gambar 9.6 Pengujian menggunakan filezilla

Ada satu lagi contoh kasus yang akan sering kita jumpai di dunia nyata (praktik kerja). Misal kita mempunyai beberapa website, sebut saja www.forkits.com dan web.forkits.com (sesuai pembahasan materi pada bab 8). Kita menginginkan agar masing-masing dari website tersebut mempunyai user admin yang berbeda, yaitu user forkits sebagai admin www.forkits.com dan user administrator sebagai admin web.forkits.com.

Contoh kasus seperti diatas mengharuskan agar saat sebuah user login ftp ke server, maka otomatis diarahkan ke web direktori masing-masing website. Jadi saat user forkits login ftp, akan diarahkan ke web direktori www.forkits.com, begitu juga saat user administrator login ftp, akan diarahkan ke web direktori web.forkits.com.

Diasumsikan bahwa web direktori dari www.forkits.com berada di /var/www/www, dan web direktori dari web.forkits.com berada di /var/www/web. Maka konfigurasi yang perlu dilakukan adalah sebagai berikut

```
root@forkits:~# nano /etc/proftpd/proftpd.conf
.....
.....
.....
.....
# Use this to jail all users in their homes
#DefaultRoot                    ~/public_html
# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
.....
.....
.....
.....
.....
# Include other custom configuration files
Include /etc/proftpd/conf.d/
<Anonymous /var/www/www>
User forkits
</Anonymous>
<Anonymous /var/www/web>
User administrator
</Anonymous>
```

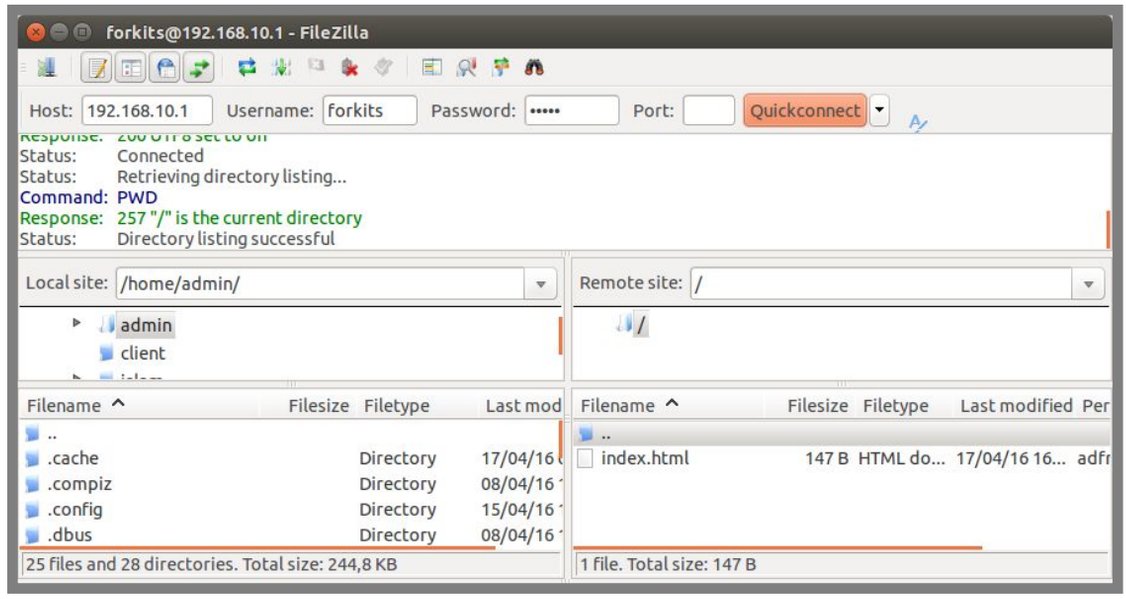
Gambar 9.7 Konfigurasi proftpd

Perhatikan gambar diatas, kita menambahkan beberapa baris konfigurasi pada file proftpd.conf yang ditunjukkan dengan teks warna hijau. Selanjutnya restart service proftpd

```
root@forkits:~# service proftpd restart
[ ok ] Stopping ftp server: proftpd.
[....] Starting ftp server: proftpdforkits proftpd[2951]:
mod_tls_memcache/0.1: notice: unable to register 'memcache' SSL session
cache: Memcache support not enabled
. ok
root@forkits:~#
```

Gambar 9.8 Merestart service proftpd

Berikut hasil pengujian saat login menggunakan user forkits,



Gambar 9.9 Pengujian menggunakann filezilla

Perhatikan gambar diatas, terlihat bahwa saat login menggunakan user forkits, telah diarahkan ke direktori /var/www/www. Namun jika kita mencoba melakukan upload file web, akan terjadi error. Hal ini karena hak akses user forkits terhadap direktori tersebut tidak mengizinkan untuk write. Untuk mengatasi hal tersebut kita bisa merubah hak akses direktori tersebut, atau yang lebih disarankan adalah merubah hak kepemilikan direktori tersebut

```
root@forkits:~# chown forkits:forkits /var/www/www/ -R
root@forkits:~#
```

Gambar 9.10 Merubah kepemilkan web directory www.forkits.com

Lakukan langkah yang sama dengan gambar diatas untuk user administrator terhadap direktori /var/www/web.

Konfigurasi FTP Anonymous Login

Seluruh skenario yang telah kita bahas di sub bab sebelumnya menggunakan metode authentikasi username dan password. Pada sub bab ini kita akan membahas konfigurasi ftp tanpa menggunakan authentikasi sama sekali. Hal ini biasa diterapkan jika ftp server yang dibuat mempunyai tujuan untuk sharing file biasa.

Berikut konfigurasi yang dapat kita lakukan

```
root@forkits:~# nano /etc/proftpd/proftpd.conf
.....
.....
.....
.....
.....
# Include other custom configuration files
Include /etc/proftpd/conf.d/
<Anonymous /ftp>
User free
UserAlias anonymous free
</Anonymous>
root@forkits:~# service proftpd restart
[ ok ] Stopping ftp server: proftpd.
[....] Starting ftp server: proftpdforkits proftpd[3333]:
mod_tls_memcache/0.1: notice: unable to register 'memcache' SSL session
cache: Memcache support not enabled
. ok
root@forkits:~#
```

Gambar 9.11 Konfigurasi proftpd untuk anonymous login

Perhatikan gambar diatas, terlihat bahwa kita menambahkan beberapa baris konfigurasi pada file proftpd.conf kemudian kita merestart service proftpd. Selanjutnya kita harus membuat direktori /ftp (perhatikan bahwa kita menambahkan baris konfigurasi *<Anonymous /ftp>* yang artinya direktori ftp berada di /ftp.

```
root@forkits:~# mkdir /ftp
root@forkits:~# mkdir /ftp/First
root@forkits:~# mkdir /ftp/Second
root@forkits:~# chmod 777 /ftp/First/
root@forkits:~# ls -l /ftp/
total 8
drwxrwxrwx 2 root root 4096 Apr 19 21:53 First
drwxr-xr-x 2 root root 4096 Apr 19 21:53 Second
root@forkits:~#
```

Gambar 9.12 Membuat direktory untuk ftp

Perintah-perintah diatas digunakan untuk membuat direktori */ftp* dan direktori *First* serta *Second* didalam direktori */ftp*. Kemudian merubah hak akses direktori *First* menjadi full access. Hal ini dimaksudkan bahwa direktori *First* ditujukan untuk keperluan upload dan download, sedangkan direktori *Second* ditujukan untuk keperluan download saja.

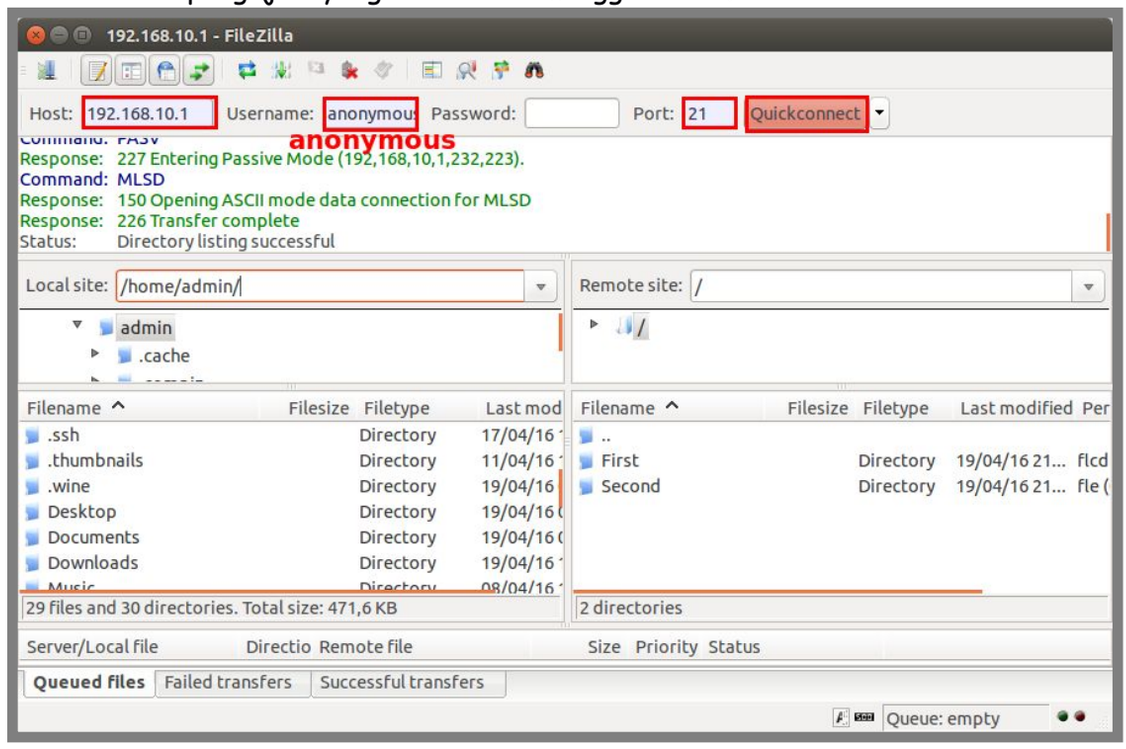
Jika kita perhatikan pada file *proftpd.conf*, kita juga melihat script *User free,* hal ini artinya kita juga harus membuat sebuah user dengan nama free

```
root@forkits:~# useradd -d /ftp/ free
root@forkits:~# passwd free
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@forkits:~#
```

Gambar 9.13 Menambahkan user untuk ftp

Perintah diatas digunakan untuk membuat sebuah user dengan nama free dengan home direktorinya berada di /ftp, dilanjutkan dengan perintah untuk memberikan password pada user tersebut.

Berikut hasil pengujian yang dilakukann menggunakan filezilla



Gambar 9.14 Pengujian anonymous login menggunakan filezilla

Konfigurasi SSL/TLS di FTP

Pada sub bab ini kita akan membahas konfigurasi FTP dengan menggunakan protocol keamanan ssl/tls. Untuk konfigurasi ftp menggunakan ssl/tls, hal pertama yang harus dilakukan adalah membuat file private key dan csr, kemudian mengajukan csr kepada CA, seperti yang telah dijelaskan pada bab 7.

Pada sub bab ini, diasumsikan bahwa kita telah mempunyai file private key dan juga file sertifikat yang telah ditandatangani oleh CA. Langkah-langkah untuk membuat private key sama persis dengan pembahasan pada bab https di bab 8. Jika tidak mau repot, kita bisa langsung copy private key dan juga file sertifikat yang telah ditandatangani CA dari yang telah kita buat pada bab https di bab 8.

```
root@forkits:~# cp -rf /etc/apache2/ssl/ /etc/proftpd/
root@forkits:~# ls /etc/proftpd/ssl/
forkits.crt   forkits.csr   forkits.key
root@forkits:~#
```

Gambar 9.15 Copy file private key dan sertifikat

Perhatikan gambar diatas, terlihat bahwa kita telah mempunyai file private key dengan nama *forkits.key* dan file sertifikat dengan nama *forkits.crt*. Sedangkan file CSR dengan nama *forkits.csr* sebenarnya sudah tidak kita butuhkan (bisa dihapus).

Selanjutnya rubah konfigurasi profptd

```
root@forkits:~# nano /etc/proftpd/proftpd.conf
.....
.....
.....
# Alternative authentication frameworks
#
#Include /etc/proftpd/ldap.conf
#Include /etc/proftpd/sql.conf

#
# This is used for FTPS connections
#
Include /etc/proftpd/tls.conf

#
# Useful to keep VirtualHost/VirtualRoot directives separated
#
#Include /etc/proftpd/virtuals.conf
.....
.....
.....
```

Gambar 9.16 Mengaktifkan tls pada proftpd

Cari dan rubah pada baris konfigurasi yang ditunjukkan dengan teks warna hijau. Selanjutnya rubah konfigurasi pada file tls.conf

```
root@forkits:~# nano /etc/proftpd/tls.conf
.....
.....
.....
<IfModule mod_tls.c>
TLSEngine                               on
TLSLog                                  /var/log/proftpd/tls.log
TLSProtocol                             SSLv23
#
.....
.....
.....
# The proftpd.key file must be readable by root only. The other file can be
# readable by anyone.
#
# chmod 0600 /etc/ssl/private/proftpd.key
# chmod 0640 /etc/ssl/private/proftpd.key
#
TLSRSACertificateFile                   /etc/proftpd/ssl/forkits.crt
TLSRSACertificateKeyFile                /etc/proftpd/ssl/forkits.key
#
.....
.....
.....
```

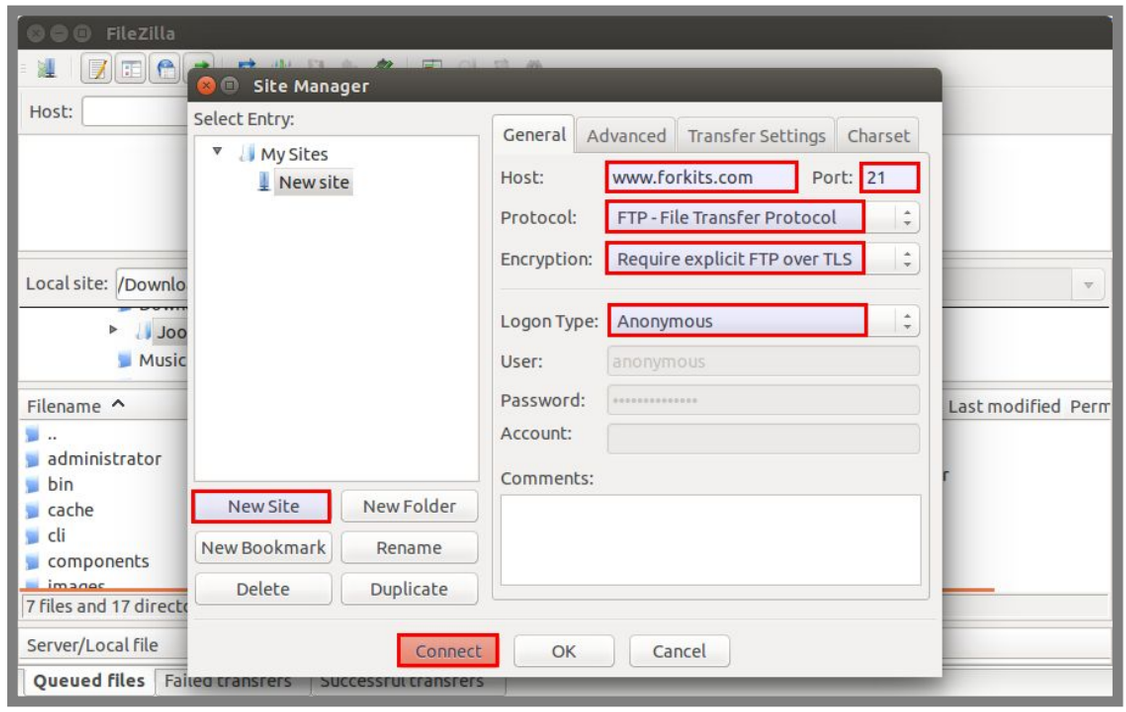
Gambar 9.17 Konfigurasi tls pada proftpd

Cari dan rubah baris konfigurasi yang ditandai teks warna hijau. Langkah terahir restart service profptd

```
root@forkits:~# service proftpd restart
[ ok ] Stopping ftp server: proftpd.
[....] Starting ftp server: proftpdforkits proftpd[3569]:
mod_tls_memcache/0.1: notice: unable to register 'memcache' SSL session
cache: Memcache support not enabled

Please provide passphrases for these encrypted certificate keys:
RSA key for the 127.0.1.1#21 (Debian) server: (tidak terlihat)
Verifying - RSA key for the 127.0.1.1#21 (Debian) server: (tidak terlihat)
. ok
root@forkits:~#
```

Gambar 9.18 Merestart service ftp server

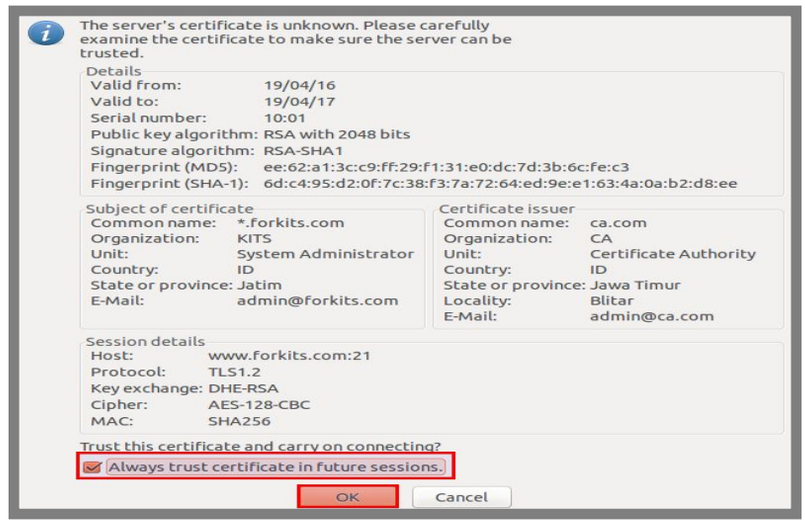
Untuk melakukan pengujian, buka filezilla kemudian klik *file > site manager > new site*



Gambar 9.19 Konfigurasi ssl pada filezilla

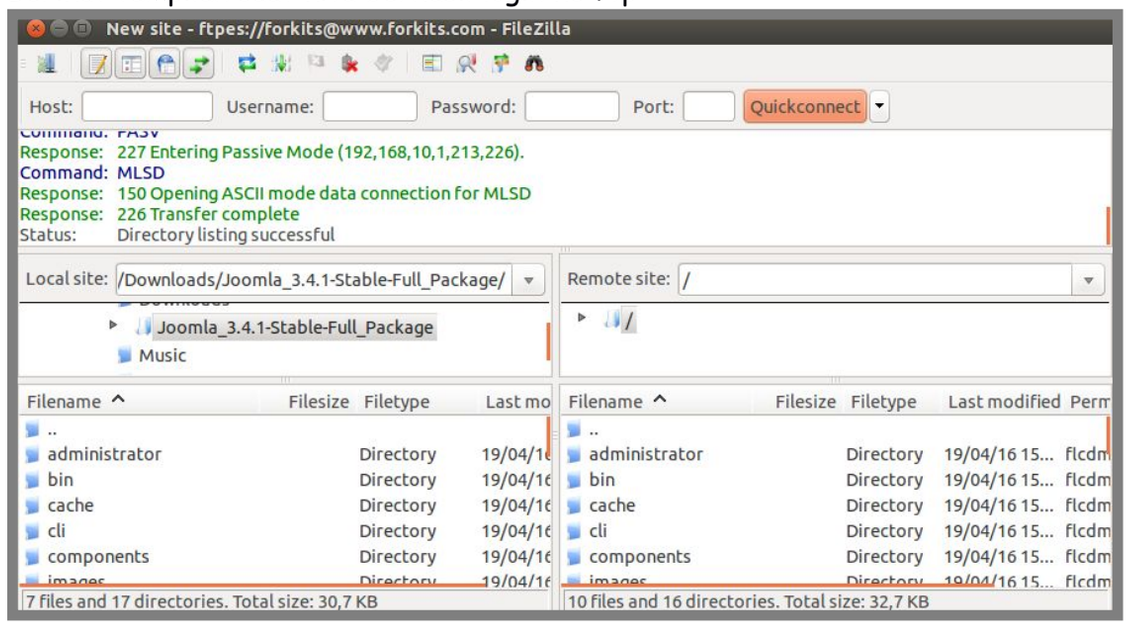
Perhatikan gambar diatas, terlihat bahwa host yang kita gunakan adlah domain milik server, yaitu www.forkits.com. Sebenarnya kita bisa saja menggunakan ip address server, namun bukannya lebih mudah mengingat nama domain daripada ip address??

Perhatikan juga bahwa metode enkripsi yang digunakan adalah tls. Sedangkan login type yang digunakan adalah anonymous, hal ini karena kita telah konfigurasi anonymous login di proftpd, namun jika proftpd tidak dikonfigurasi anonymous login, maka login type yang dipilih adalah ask for password. Selanjutnya akan muncul sebuah peringatan tentang sertifikat yang kita miliki dan sertifikat dari CA yang kita gunakan



Gambar 9.21 Peringatan keamanan SSL/TLS

Berikut tampilan saat kita berhasil login ke ftp



Gambar 9.22 Hasil pengujian ssl/tls pada filezilla

---END OF CHAPTER---

Bab 10
Mail & Webmail Server

Mail server adalah sebuah server yang digunakan untuk mengirim surat elektronik (email) melalui internet.

Berikut gambaran umum proses pengiriman sebuah email dari seseorang (sebut saja A) ke orang lain (sebut saja B). Misal suatu saat A mengirim email ke B, sebenarnya A tidak mengirim ke B, melainkan mengirim ke mail server. Selanjutnya barulah si B mengambil email yang dikirimkan oleh A dari mail server.

Jika kita perhatikan, proses pengiriman suatu email yang telah dijelaskan diatas mempunyai dua tahap utama. Yaitu proses pengiriman email dari A ke mail server dan proses pengambilan email oleh B dari mail server. Masing-masing tahap tersebut mempunyai protocol yang mengatur. Yaitu SMTP sebagai protocol untuk mengirim email dan POP/IMAP sebagai protocol untuk mengambil email.

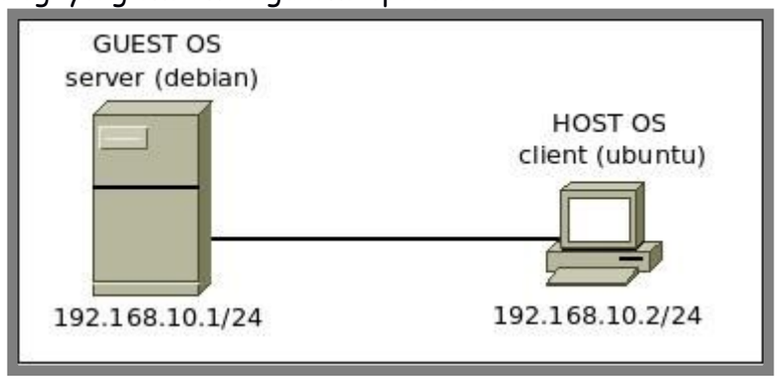
Simple Mail Transfer Protocol (SMTP) adalah sebuah protocol yang bertugas untuk mengirim sebuah email yang bergerak pada port 25.

Post Office Protocol (POP) adalah sebuah protocol yang bertugas untuk menerima sebuah email yang bergerak pada port 110. Jika protocol yang digunakan adalah POP, apabila kita membaca email yang masuk (inbox), maka seluruh inbox yang ada di mail server akan didownload dan dihapus, sehingga kita hanya bisa membaca email yang masuk dari satu komputer/laptop yang mendownload inbox tersebut.

Internet Mail Application Protocol (IMAP). Sama halnya dengan POP, protocol ini juga digunakan untuk menerima email. Perbedaannya terdapat pada metode menerima email. Jika pada POP, seluruh inbox akan didownload dan dihpus. Sedangkan pada IMAP, inbox akan didownload namun tidak dihapus. Hal ini memungkinkan kita untuk membaca inbox dari komputer/laptop lain. IMAP menggunakan port nomor 143.

Konfigurasi Mail Server

Berikut topologi yang akan kita gunakan pada bab ini



Gambar 10.1 Topologi jaringan untuk praktik mail server

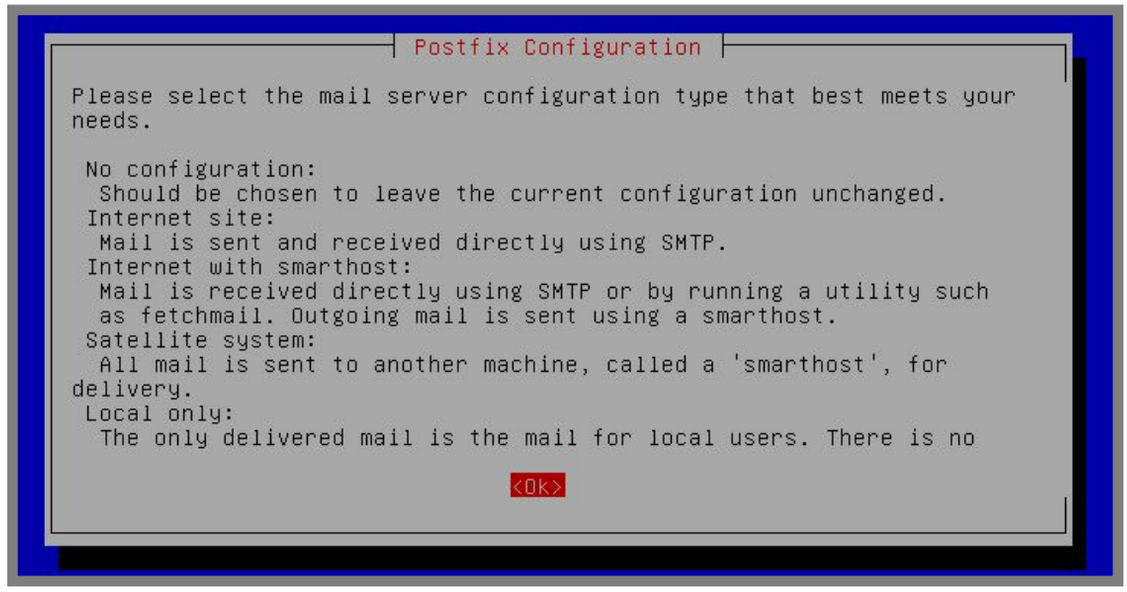
Diasumsikan bahwa komputer server dan client telah dikonfigurasi ip address sesuai topologi diatas. Diasumsikan juga bahwa komputer server telah diinstall dan dikonfigurasi sebagai dns server sesuai dengan materi pada bab 6.

Ada beberapa aplikasi yang kita perlukan untuk membuat sebuah mail server, yaitu *postfix* sebagai SMTP, *courier-pop* sebagai POP, dan *courier-imap* sebagai IMAP. Berikut perintah untuk menginstall ketiga aplikasi tersebut

```
root@forkits:~# apt-get install postfix courier-pop courier-imap
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  courier-authdaemon courier-authlib courier-authlib-userdb courier-base
  expect libfam0 tcl8.5
Suggested packages:
  courier-doc courier-imap-ssl courier-pop-ssl fam postfix-mysql
postfix-pgsql
  postfix-ldap postfix-pcre sasl2-bin dovecot-common resolvconf
postfix-cdb
  ufw postfix-doc tcl-tclreadline
The following packages will be REMOVED:
  exim4 exim4-base exim4-config exim4-daemon-light
The following NEW packages will be installed:
  courier-authdaemon courier-authlib courier-authlib-userdb courier-base
  courier-imap courier-pop expect libfam0 postfix tcl8.5
0 upgraded, 10 newly installed, 4 to remove and 0 not upgraded.
Need to get 0 B/4144 kB of archives.
After this operation, 5383 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

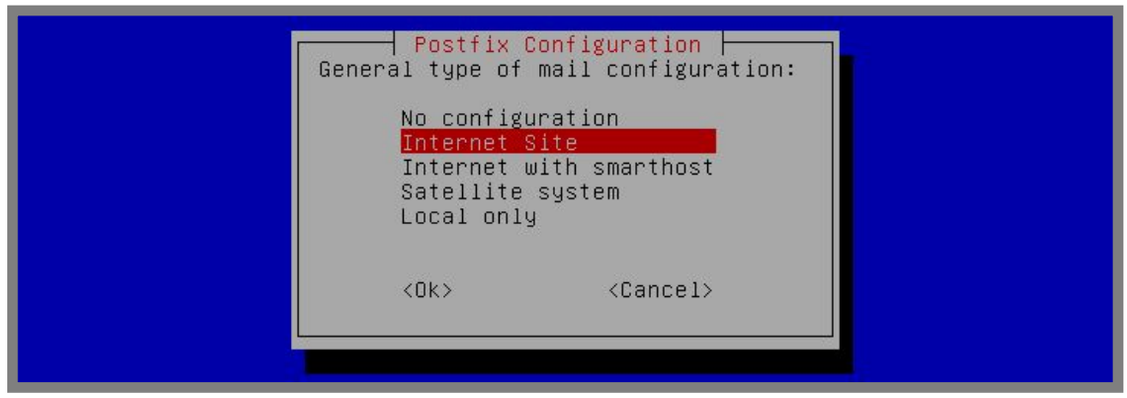
Gambar 10.2 Installasi aplikas untuk mail server

Pastikan memilih ok pada kolom persetujuan berikut



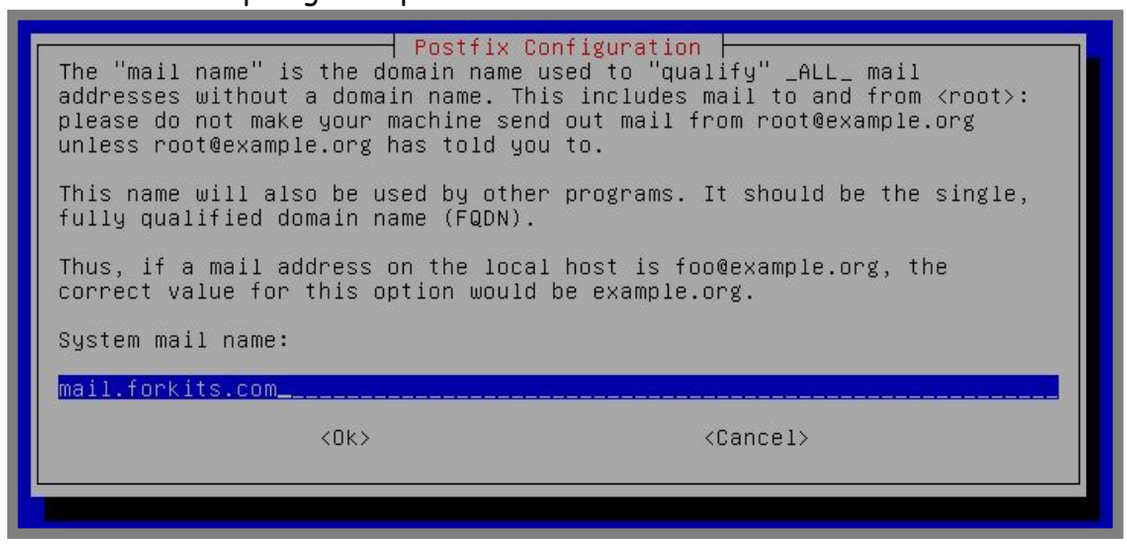
Gambar 10.3 Proses installasi postfix

Pilih type dari mail server yang akan diinstall, kita pilih internet site

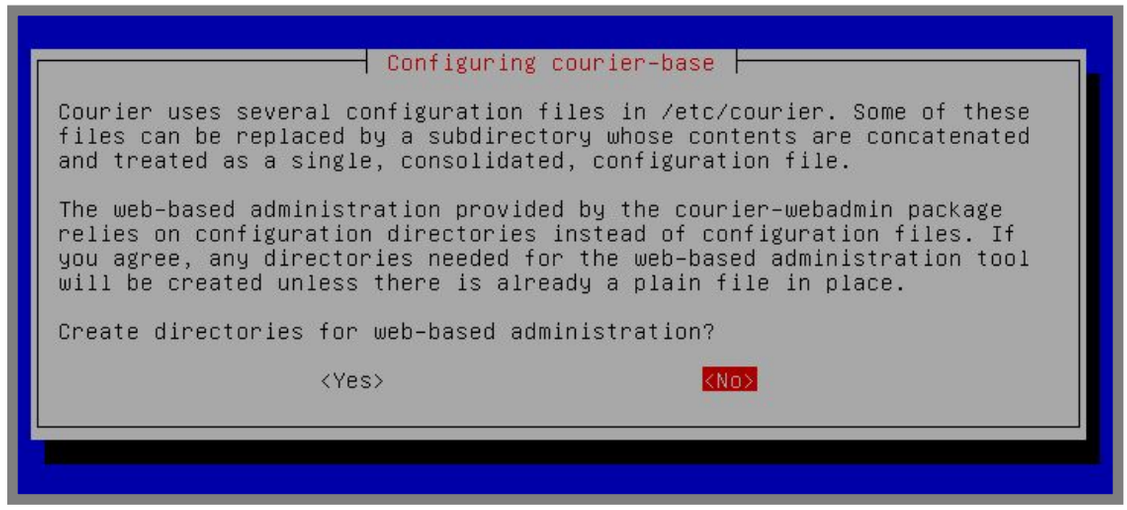


Gambar 10.4 Proses installasi postfix

Pada kolom mail name seperti gambar dibawah ini, isikan dengan domain yang memiliki nilai MX paling kecil pada dns server



Gambar 10.5 Proses installasi postfix



Gambar 10.6 Proses installasi postfix

Setelah proses installasi selesai, langkah selanjutnya yang harus kita lakukan adalah membuat sebuah direktori untuk menampung email yang masuk pada masing-masing user.

```
root@forkits:~# maildirmake /etc/skel/Maildir
root@forkits:~#
```

Gambar 10.7 Membuat maildir untuk setiap user

Maksud dari perintah diatas adalah, nantinya setiap kita menambahkan user baru, otomatis akan dibuatkan sebuah direktori *Maildir* didalam home direktori masing-masing.

Ingat, bahwa yang dibuatkan direktori *Maildir* adalah user yang dibuat setelah mengeksekusi perintah diatas, jika ada user yang dibuat sebelum mengeksekusi perintah diatas, kita harus membuatkannya secara manual. Misal kita telah mempunyai user dengan nama forkits sebelum mengeksekusi perintah pada gambar 10.7, maka berikut perintah yang bisa kita gunakan untuk membuat direktori *Maildir* didalam home direktori user forkits

```
root@forkits:~# maildirmake /home/forkits/Maildir
root@forkits:~# chown forkits:forkits /home/forkits/Maildir/ -R
root@forkits:~#
```

Gambar 10.8 Membuat maildir untuk user yang telah ada

Selanjutnya kita harus menambahkan beberapa konfigurasi pada postfix

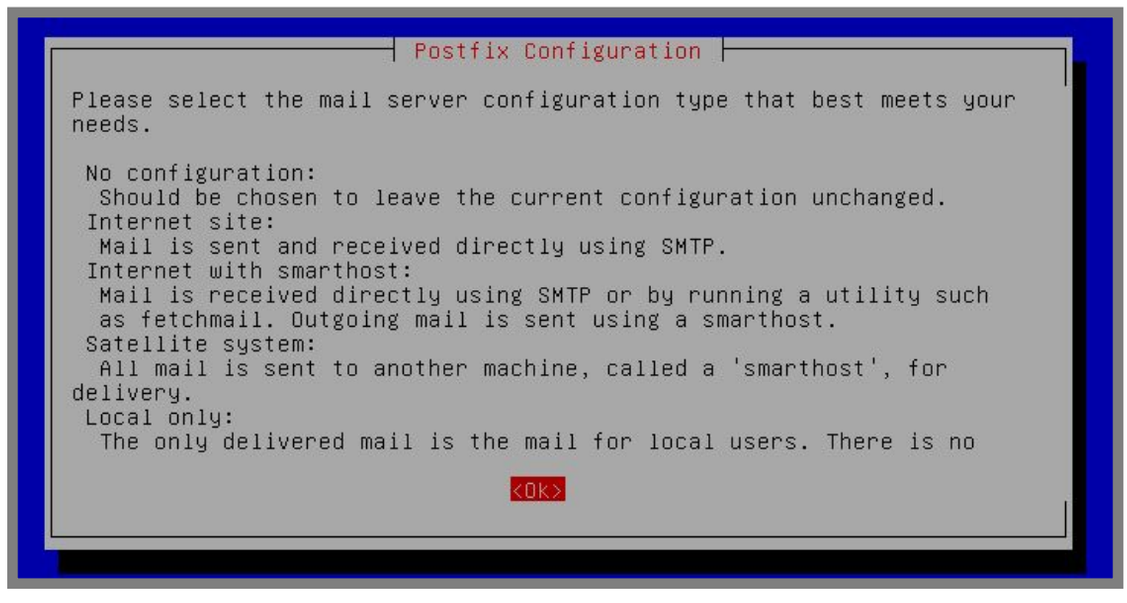
```
root@forkits:~# nano /etc/postfix/main.cf
.....
.....
.....
recipient_delimiter = +
inet_interfaces = all
home_mailbox = Maildir/
```

Gambar 10.9 Konfigurasi postfix

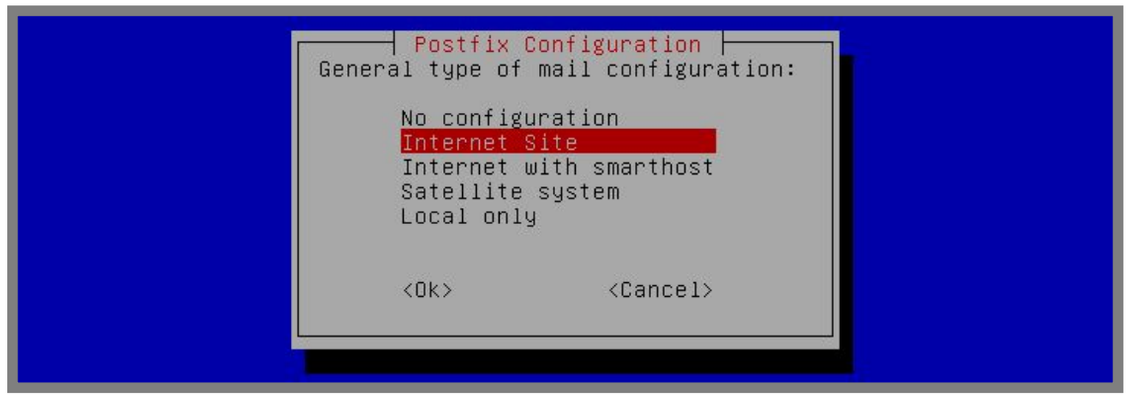
Perhatikan gambar diatas, terlihat bahwa kita menambahkan satu baris konfigurasi pada postfix. Selanjutnya kita harus reconfigure / konfigurasi ulang postfix dengan perintah berikut

```
root@forkits:~# dpkg-reconfigure postfix
```

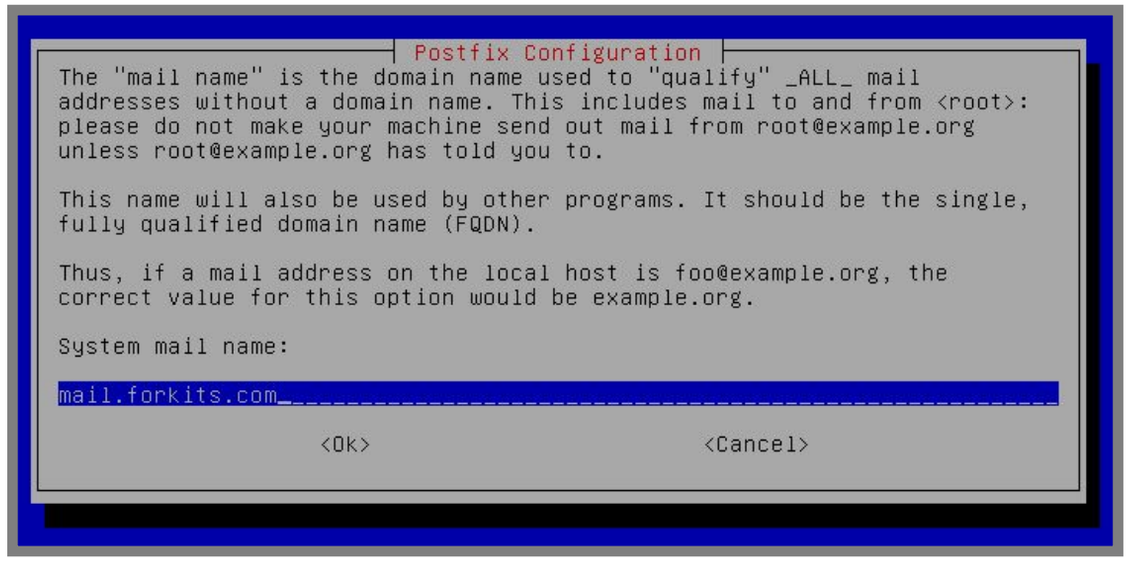
Gambar 10.10 Konfigurasi ulang postfix



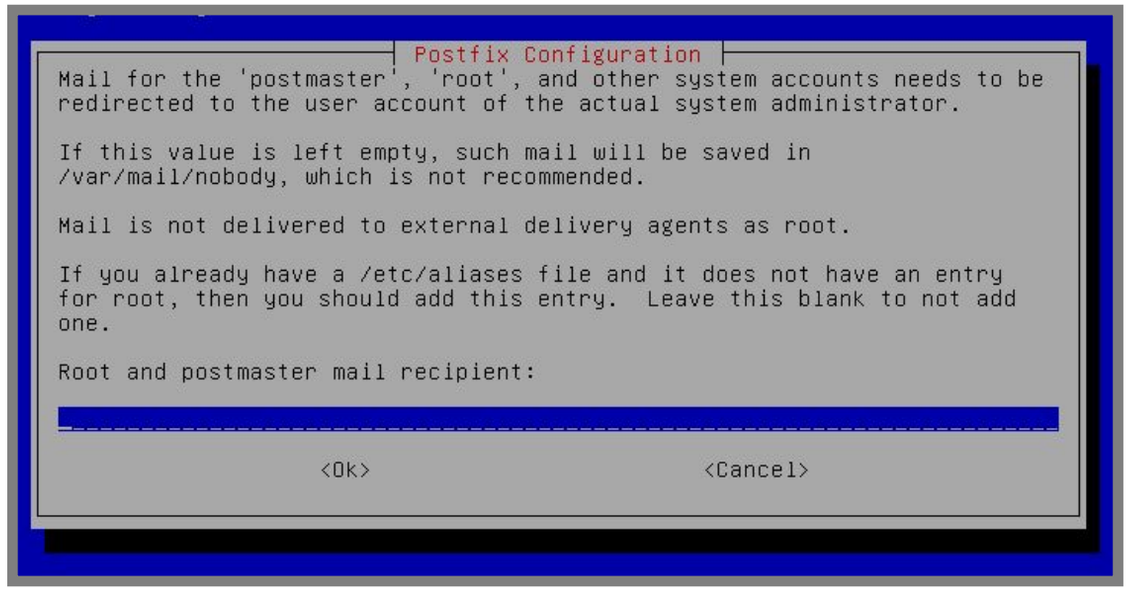
Gambar 10.11 Proses konfigurasi ulang postfix



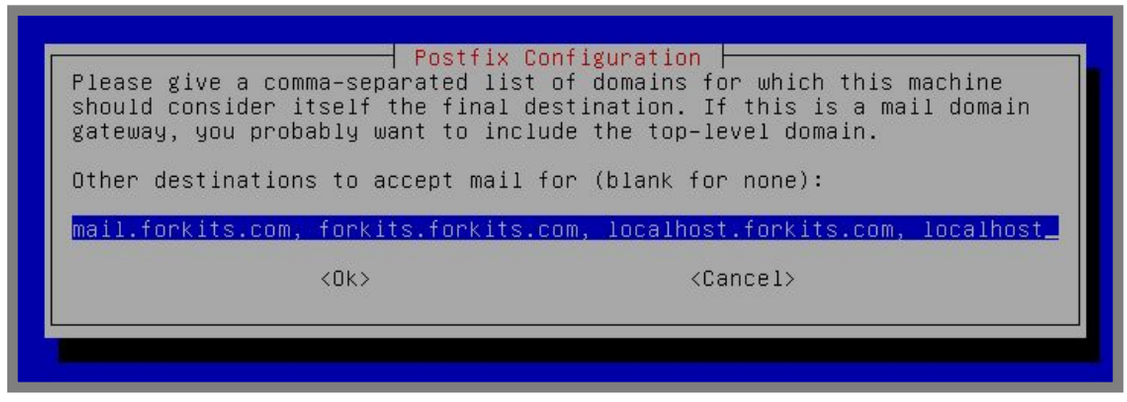
Gambar 10.12 Proses konfigurasi ulang postfix



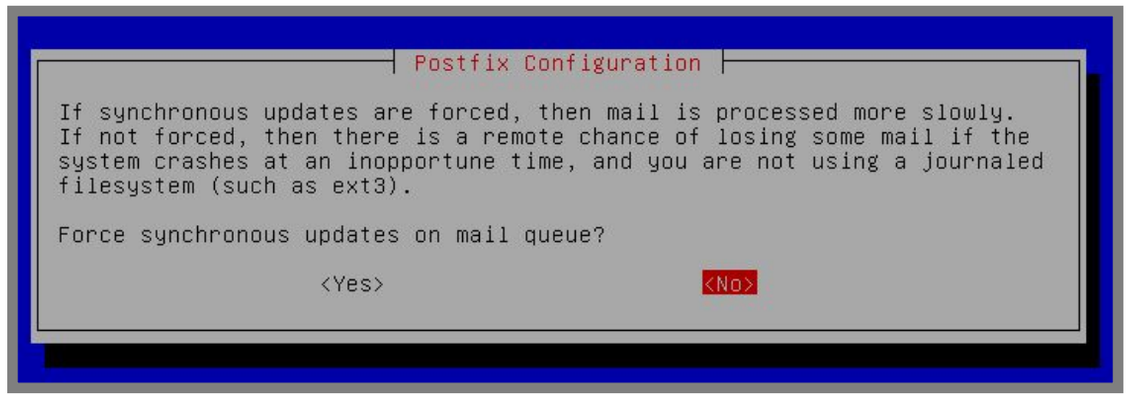
Gambar 10.13 Proses konfigurasi ulang postfix



Gambar 10.14 Proses konfigurasi ulang postfix

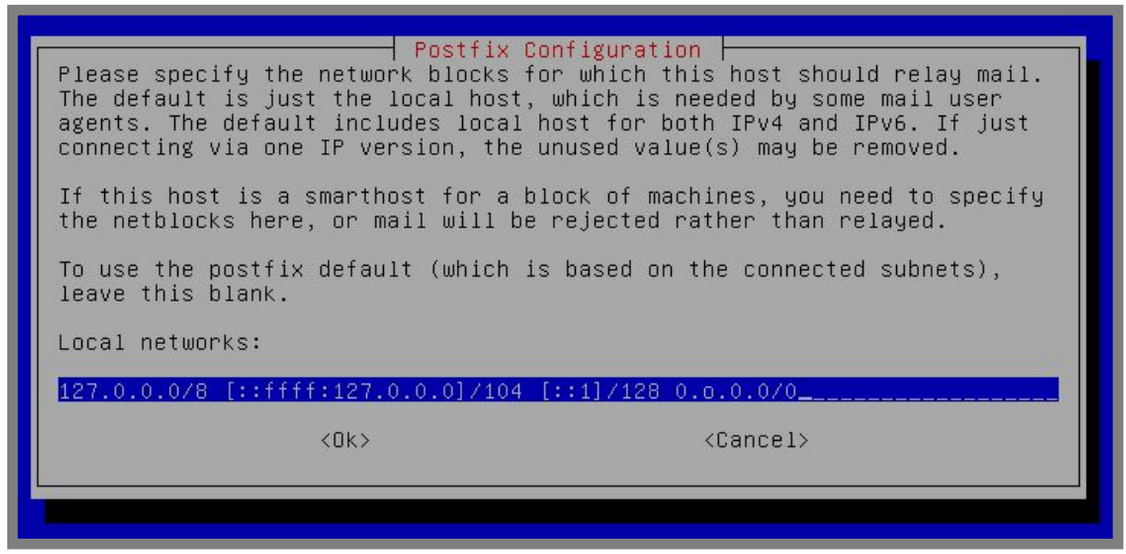


Gambar 10.15 Proses konfigurasi ulang postfix

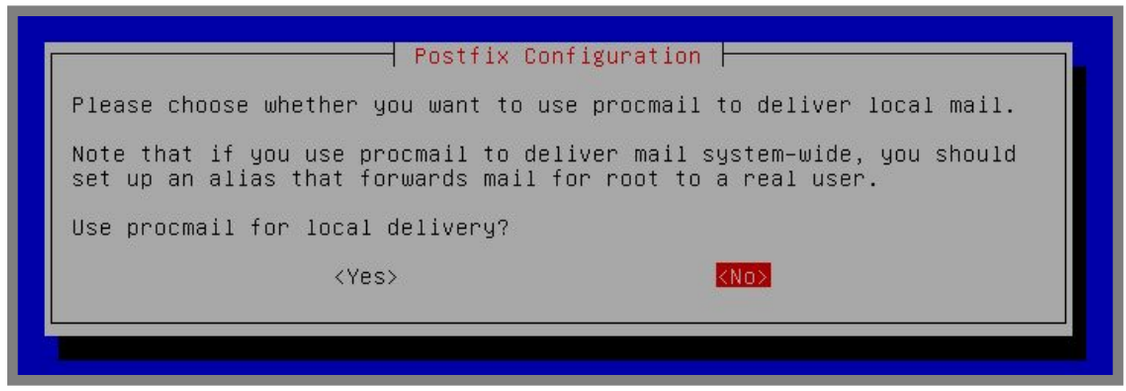


Gambar 10.16 Proses konfigurasi ulang postfix

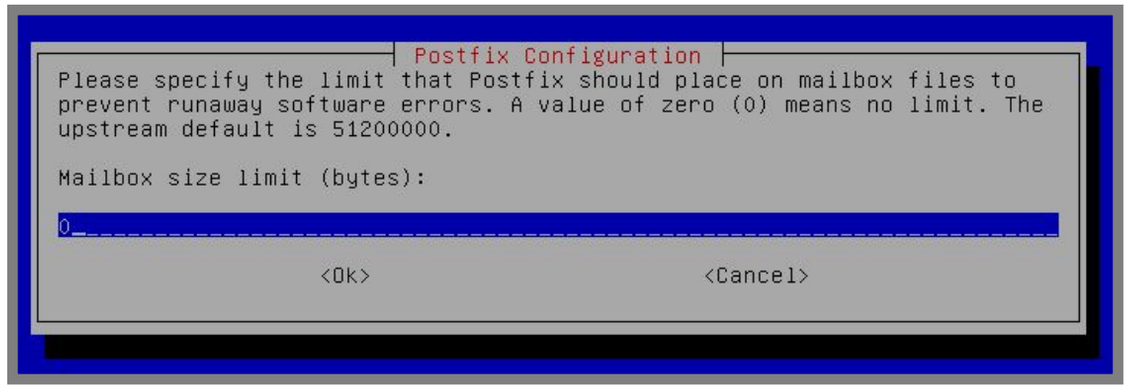
Tambahkan 0.0.0.0/0 yang artinya semua ip bisa menggunakan mail server yang kita buat



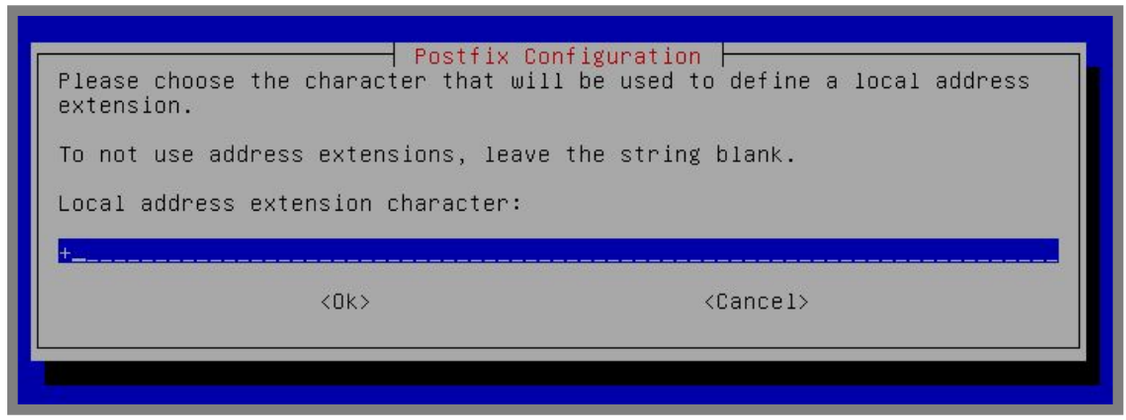
Gambar 10.17 Proses konfigurasi ulang postfix



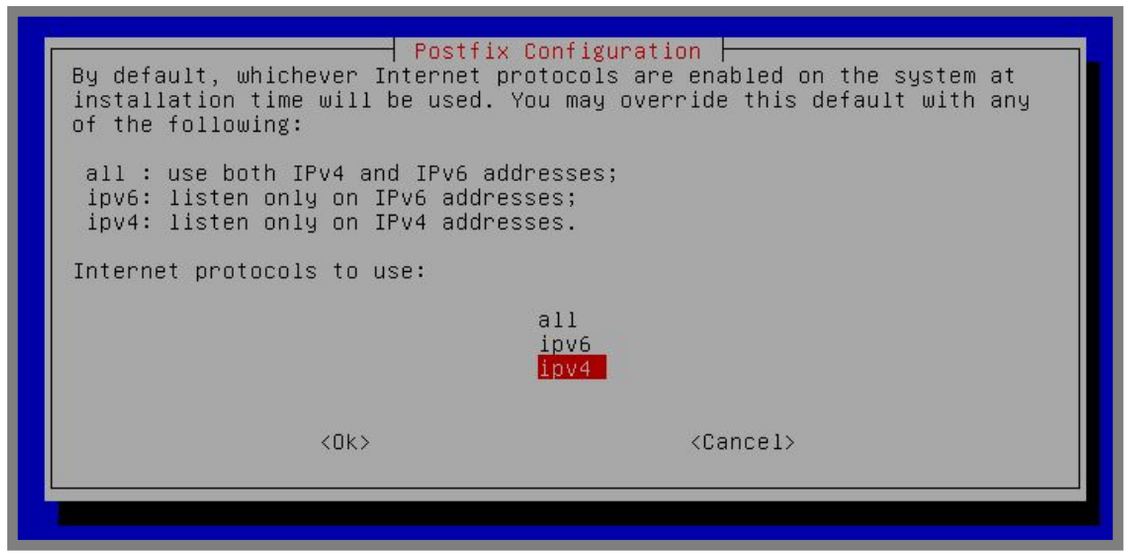
Gambar 10.18 Proses konfigurasi ulang postfix



Gambar 10.19 Proses konfigurasi ulang postfix



Gambar 10.20 Proses konfigurasi ulang postfix



Gambar 10.21 Proses konfigurasi ulang postfix

Selanjutnya tambahkan beberapa user untuk melakukan pengujian mail server. Paling tidak dua user, satu user bertindak sebagai pengirim email, dan satu user bertindak sebagai penerima email.

```
root@forkits:~# adduser mail1
Adding user `mail1' ...
Adding new group `mail1' (1007) ...
Adding new user `mail1' (1006) with group `mail1' ...
Creating home directory `/home/mail1' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: (tidak terlihat)
Retype new UNIX password: (tidak terlihat)
passwd: password updated successfully
Changing the user information for mail1
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
root@forkits:~#
```

Gambar 10.22 Menambahkan user untuk pengujian mail server

Lakukan langkah yang sama seperti diatas untuk menambahkan user mail2. Berikut pengujian yang dilakukan pada komputer client untuk mengirim email dari mail1

```
admin@ubuntu:~$ telnet mail.forkits.com 25
Trying 192.168.10.1...
Connected to mail.forkits.com.
Escape character is '^]'.
220 forkits.forkits.com ESMTP Postfix (Debian/GNU)
mail from: mail1
250 2.1.0 Ok
rcpt to: mail2
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Salam kenal dari mail1
senang berkenalan dengan mu
.
250 2.0.0 Ok: queued as 4DAE5C09F7
quit
221 2.0.0 Bye
Connection closed by foreign host.
admin@ubuntu:~$
```

Gambar 10.23 Mengirim email dari mail1 ke mail2

Berikut penjelasan dari masing-masing perintah yang digunakan diatas

Syntak	Deskripsi
telnet mail.forkits.com 25	Digunakan untuk meremote mail server pada port 25, yaitu port SMTP untuk mengirim email
mail from: mail1	Menandakan bahwa yang akan mengirim email adalah mail1
rcpt to: mail2	Menandakan bahwa mail1 akan mengirim email ke mail2
data	Digunakan untuk mengawali menulis pesan (email) yang akan dikirim
Salam kenal.....	Teks email yang dikirimkan. Untuk mengahiri menulis teks yang akan dikirim, kita bisa menggunakan tanda titik (.).
quit	Digunakan untuk keluar darii SMTP

Selanjutnya berikut perintah yang digunakan untuk melihat email yang masuk pada mail2

```
admin@ubuntu:~$ telnet mail.forkits.com 110
Trying 192.168.10.1...
Connected to mail.forkits.com.
Escape character is '^]'.
+OK Hello there.
user mail2
+OK Password required.
pass 123
+OK logged in.
stat
+OK 1 315
retr 1
+OK 315 octets follow.
Return-Path: <mail1@mail.forkits.com>
X-Original-To: mail2
Delivered-To: mail2@mail.forkits.com
Received: from unknown (unknown [192.168.10.2])
      by forkits.forkits.com (Postfix) with SMTP id 4DAE5C09F7
      for <mail2>; Thu, 21 Apr 2016 20:27:50 +0700 (WIB)

Salam kenal dari mail1
senang berkenalan dengan mu
.
quit
+OK Bye-bye.
Connection closed by foreign host.
admin@ubuntu:~$
```

Gambar 10.24 Melihat email yang masuk pada user mail2

Berikut penjelasan dari masing-masing perintah yang digunakan diatas

Syntak	Deskripsi
telnet mail.forkits.com 110	Digunakan untuk meremote mail server pada port 110, yaitu port POP untuk melihat email yang masuk
user mail2	Digunakan untuk login sebagai user mail2
pass 123	Menandakan bahwa password user mail2 adalah 123
stat	Digunakan untuk melihat kondisi inbox yang ada. Perhatikan hasilny *(OK 1 315).* Itu artinya ada satu email pada inbox
retr 1	Digunakan untuk melihat isi email yang ada pada inbox
quit	Digunakan untuk keluar darii SMTP

Konfigurasi SMTP No Relay

Jika kita melihat mail server yang ada di internet, katakanlah gmail, bisakah kita login ke gmail menggunakan akun dari yahoo? Tentu saja tidak! Karena hal ini menyangkut keamanan dari server gmail.

Secara default, kita bisa login ke mail server yang kita buat menggunakan user dari mail server lain, seperti gmail, yahoo, ymail, ataupun akun lain. Berikut contoh nyatanya

```
admin@ubuntu:~$ telnet mail.forkits.com 25
Trying 192.168.10.1...
Connected to mail.forkits.com.
Escape character is '^]'.
220 forkits.forkits.com ESMTP Postfix (Debian/GNU)
mail from: rosid@gmail.com
250 2.1.0 Ok
rcpt to: mail1
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
email dari penyusup
.
250 2.0.0 Ok: queued as DEA74C09F7
quit
221 2.0.0 Bye
Connection closed by foreign host.
admin@ubuntu:~$
```

Gambar 10.25 Pengujian yang menunjukkan smtp relay

Perhatikan gambar diatas, terlihat bahwa kita berhasil mengirim email menggunakan mail server yang kita buat meskipun menggunakan akun dari gmail. Hal seperti ini tentu akann mengancam keamanan dari server kita. Untuk mencegah hal tersebut, kita bisa mematikan fitur relay pada SMTP dengan cara berikut

```
root@forkits:~# nano /etc/postfix/main.cf
.....
.....
.....
home_mailbox = Maildir/
inet_protocols = ipv4
smtpd_sender_restrictions = reject_unknown_sender_domain
```

Gambar 10.26 Menonaktifkan smtp relay pada postfix

Perhatikan gambar diatas, terlihat bahwa kita menambahkan satu baris konfigurasi pada postfix (teks warna hijau). Selanjutnya restart service postfix

```
root@forkits:~# service postfix restart
[ ok ] Stopping Postfix Mail Transport Agent: postfix.
[ ok ] Starting Postfix Mail Transport Agent: postfix.
root@forkits:~#
```

Gambar 10.27 Merestart service postfix

Sekarang kita bisa melakukan pengujian ulang untuk mengirim email dari akun lain

```
root@forkits:~# telnet mail.forkits.com 25
Trying 192.168.10.1...
Connected to mail.forkits.com.
Escape character is '^]'.
220 forkits.forkits.com ESMTP Postfix (Debian/GNU)
mail from: rosid@gmail.com
250 2.1.0 Ok
rcpt to: mail1
450 4.1.8 <rosid@gmail.com>: Sender address rejected: Domain not found
```

Gambar 10.28 Pengujian yang menunjukkan smtp no relay

Perhatikan gambar diatas, terlihat bahwa saat ini kita sudah tidak bisa mengirim email melalui mail server kita jika menggunakan akun dari mail server lain.

Web Mail Server dengan Squirrelmail

Sejauh ini, kita masih menggunakan telnet untuk saling bertukar email. Hal ini tentu membuat para pengguna jasa email tidak nyaman, karena harus menghafal setiap perintah untuk mengirim ataupun membuka email yang masuk.

Untuk mengatasi hal tersebut, kita akan menginstall sebuah fitur email berbasis web yang dapat digunakan untuk mengirim, menerima email, ataupun pekerjaan lain yang berkaitan dengann email.

Sebelum membuat web mail server, terlebih dahulu komputer server sudah harus terinstall dan dikonfigurasi sebagai web server.

Ada beberapa aplikasi yang bisa kita manfaatkann untuk membuat web mail server. Salah satu diantaranya yang populer karena kemudahan dalam menginstall dan mengkonfigurasi adalah squirrelmail. Berikut perintah yang bisa kita gunakan untuk menginstall squirrelmail

```
root@forkits:~# apt-get install squirrelmail
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  squirrelmail-locales squirrelmail-viewashtml
Suggested packages:
  squirrelmail-decode php5-recode imapproxy php-pear php5-ldap
The following NEW packages will be installed:
  squirrelmail squirrelmail-locales squirrelmail-viewashtml
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/3897 kB of archives.
After this operation, 14.8 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 10.29 Installasi squirrelmail sebagai web mail server

Selanjutnya tambahkan konfigurasi pada web server (apache)

```
root@forkits:~# nano /etc/apache2/apache2.conf
.....
.....
# Include the virtual host configurations:
Include sites-enabled/
ServerSignature Off
Include /etc/squirrelmail/apache.conf
```

Gambar 10.30 Konfigurasi apache untuk squirrelmail

Selanjutnya restart service apache

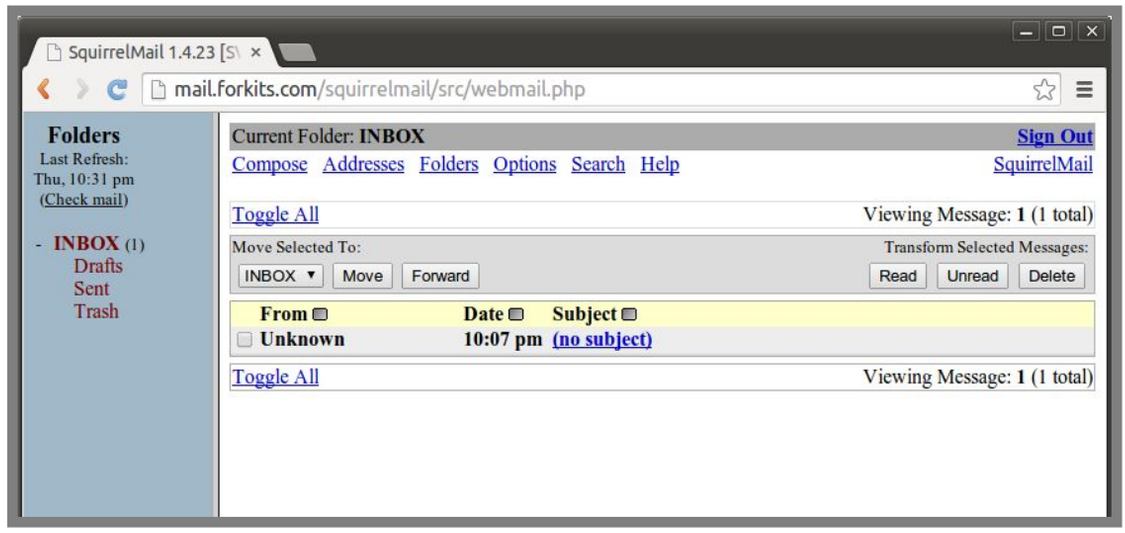
```
root@forkits:~# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server www.forkits.com:443 (RSA)
Enter pass phrase: (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:~#
```

Gambar 10.31 Merestart service apache

Sampai saat ini kita telah bisa mengakses web mail dengan url *domain/squirrelmail*

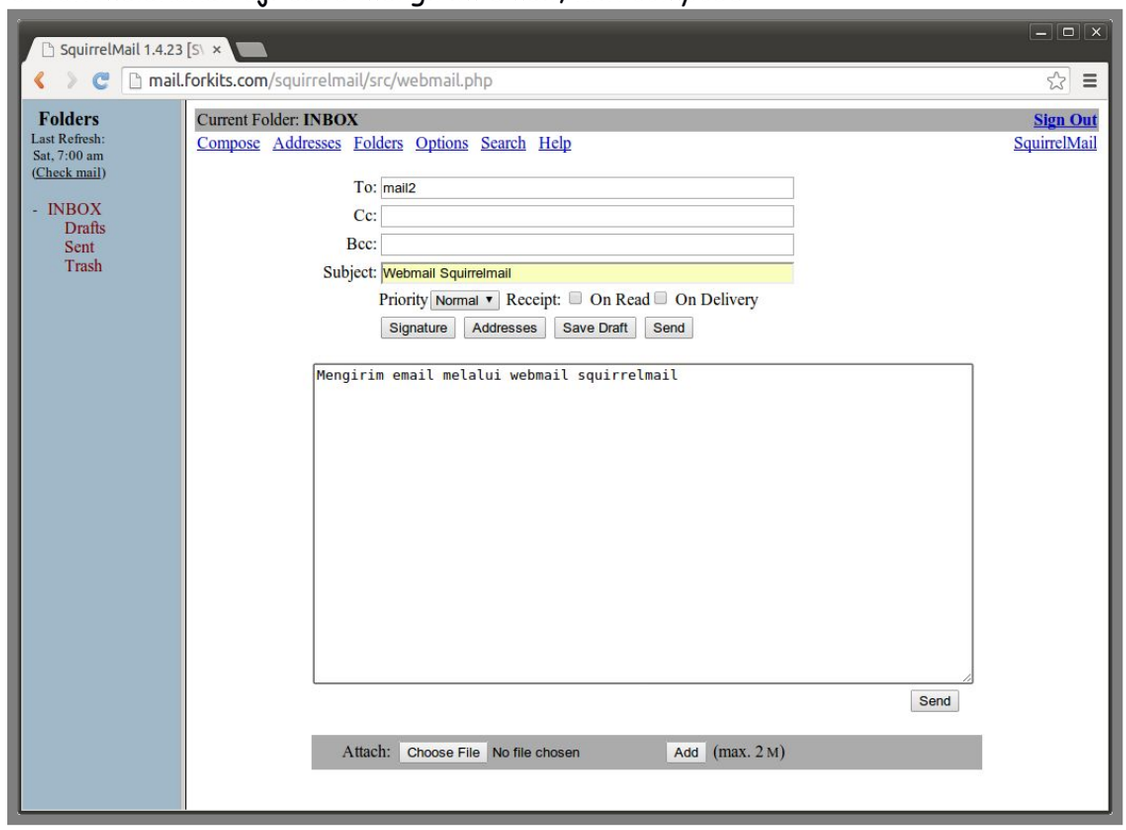
Gambar 10.32 Halaman login darii squirrelmail

Berikut tampilan halaman depan saat login ke web mail



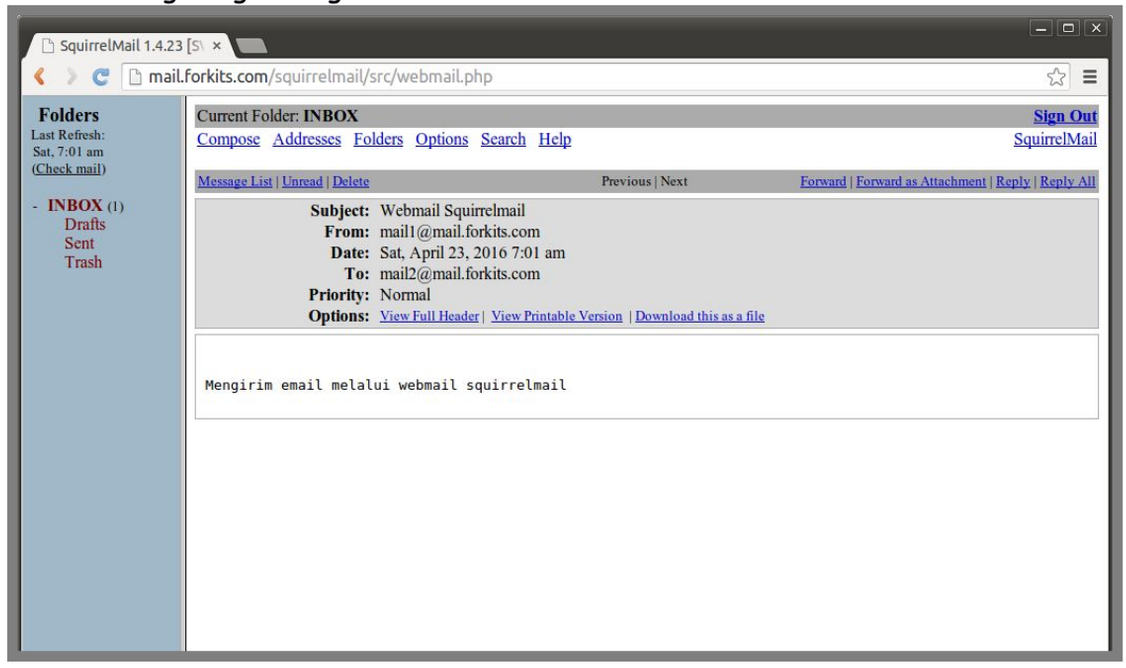
Gambar 10.33 Halaman depan squirrelmail

Untuk melakukan uji coba mengirim email, klik *compose*



Gambar 10.34 Mengirim email menggunakan squirrelmail

Setelah selesai menulis pesan, klik *send* pada gambar diatas. Selanjutnya untuk memeriksa apakah email benar-benar terkirim atau tidak, kita harus sign out kemudian login lagi sebagai mail2



Gambar 10.35 Melihat inbox pada user mail2

Perhatikan gambar diatas, terlihat bahwa email yang baru kita kirim dari user mail1 (dengan subject *Webmail Squirrelmail*) telah terkirim.

Merubah URL Squirrelmail

Secara default kita bisa mengakses squirrelmail dengan url *domain/squirrelmail.* Kita bisa merubah url tersebut sesuai kenginan kita. Sebagai contoh kasus, misal kita menginginkan agar squirrelmail bisa diakses dengan url *domain/webmail,* maka berikut langkah-langkah yang perlu kita lakukan

```
root@forkits:~# nano /etc/squirrelmail/apache.conf
Alias /webmail /usr/share/squirrelmail
.....
.....
```

Gambar 10.36 Konfigurasi squirrelmail untuk merubah url

Perhatikan gambar diatas, terlihat bahwa kita hanya melakukan sedikit perubahan pada baris pertama konfigurasi squirrelmail. Selanjutnya restart service apache

```
root@forkits:~# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server www.forkits.com:443 (RSA)
Enter pass phrase: (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:~#
```

Gambar 10.37 Merestart serive apache

Berikut hasil pengujian yang dilakukan

Gambar 10.38 Pengujian dari client

Contoh kasus yang kedua, misal kita menginginkan agar squirrelmail dapat diakses hanya menggunakan domain saja (mail.forkits.com). Maka berikut konfigurasi yang perlu dilakukan

```
root@forkits:~# nano /etc/squirrelmail/apache.conf
.....
.....
# users will prefer a simple URL like http://webmail.example.com
<VirtualHost *:80>
   DocumentRoot /usr/share/squirrelmail
   ServerName mail.forkits.com
</VirtualHost>
.....
.....
```

Gambar 10.39 Konfigurasi squirrelmail

Selanjutnya restart service apache

```
root@forkits:~# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.
Server www.forkits.com:443 (RSA)
Enter pass phrase: (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:~#
```

Gambar 10.40 Merestart service apache

Berikut hasil pengujian yang dilakukan

Gambar 10.41 Pengujian dari client

HTTPS pada Webmail Squirrelmail

Sejauh ini kita masih mengakses webmail squirrelmail menggunakan protocol http. Untuk lebih mengamankan koneksi, kita bisa merubahnya agar menggunakan protocol https.

Untuk melakukan hal tersebut, kita hanya perlu membuat virtualhost ssl seperti materi yang telah kita bahas pada bab 8. Berikut langkah-langkah konfigurasinya

```
root@forkits:~# cd /etc/apache2/sites-available/
root@forkits:/etc/apache2/sites-available# cp default-ssl squirrelmail-ssl
root@forkits:/etc/apache2/sites-available# nano squirrelmail-ssl
<IfModule mod_ssl.c>
<VirtualHost 192.168.10.1:443>
        ServerAdmin admin@forkits.com
        ServerName mail.forkits.com
        DocumentRoot /usr/share/squirrelmail
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /usr/share/squirrelmail>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
        ..........................................................................................
        ..........................................................
        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2.2-common/README.Debian.gz for info
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/apache2/ssl/forkits.crt
        SSLCertificateKeyFile /etc/apache2/ssl/forkits.key
        ..........................................................................
        ..........................................................................
        ..........................................................
root@forkits:/etc/apache2/sites-available# a2ensite squirrelmail-ssl
Enabling site squirrelmail-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@forkits:/etc/apache2/sites-available#
```

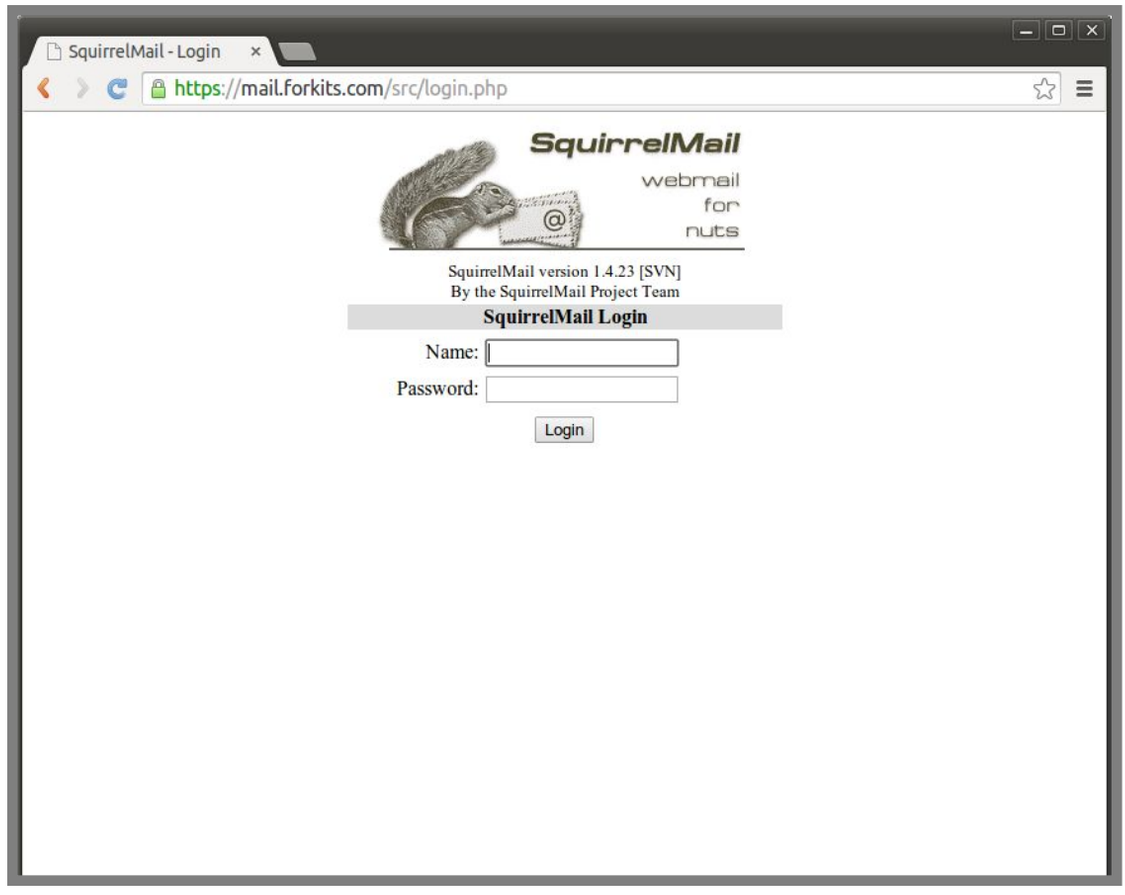
Gambar 10.42 Konfigurasi virtualhost untuk squirrelmail

Perhatikan gambar diatas, terlihat bahwa kita membuat sebuah virtualhost ssl dengan nama squirrelmail-ssl dan melakukan sedikit perubahan konfigurasi pada virtualhost tersebut. Dilanjutkan dengan perintah untuk mengaktifkan virtualhost tersebut. Selanjutnya restart service apache

```
root@forkits:/etc/apache2/sites-available# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.
Server www.forkits.com:443 (RSA)
Enter pass phrase: (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:/etc/apache2/sites-available#
```

Gambar 10.43 Merestart service apache

Berikut hasil pengujian yang dilakukan



Gambar 10.44 Pengujian dari client

Webmail Server dengan Roundcube

Terdapat satu lagi aplikasi yang sering digunakan sebagai webmail server, yaitu roundcube. Dari segi tampilan, roundcube memiliki kualitas yang jauh lebih bagus daripada squierrelmai. Jika dilihat dari segi konfigurasi, roundcube memang sedikit lebih rumit dari squirrelmail, namun bukan berarti sulit untuk dilakukan.

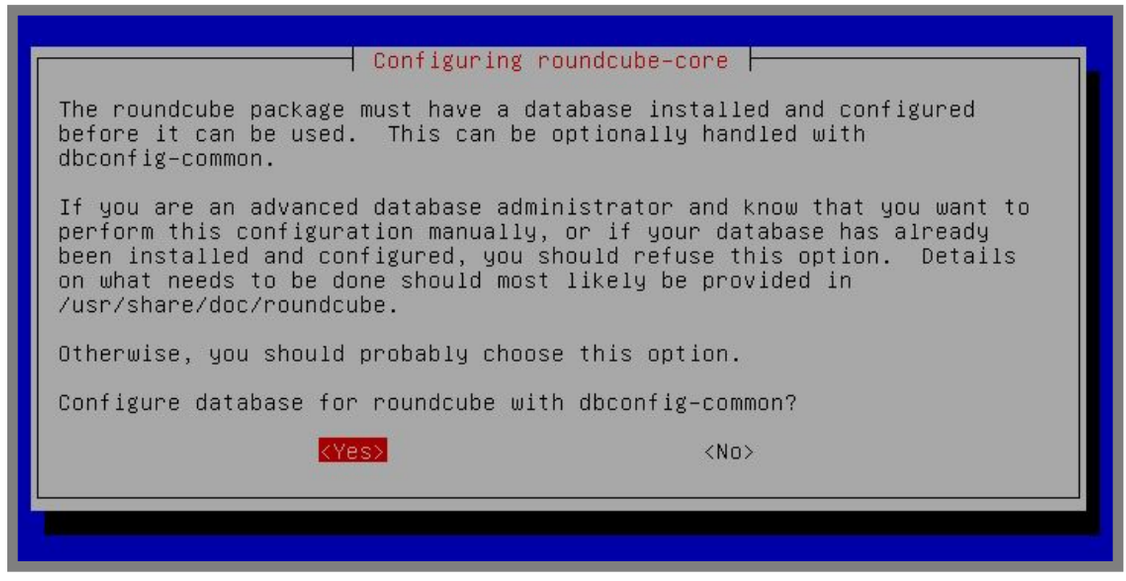
Pada sub bab ini, diasumsikan bahwa kita sudah tidak membutuhkan web mail squirrelmail. Hal ini karena nantinya domain yang akan digunakan untuk roundcube sama dengan domain yang digunakan oleh squirrelmail, yaitu mail.forkits.com. Sehingga nantinya saat kita mengakses mail.forkits.com tidak akan muncul tampilan squirrelmail, melainkan akan muncul tampilan roundcube.

Sebelum menginstall roundcube, di komputer server sudah harus terinstall dan dikonfigurasi sebagai database dan web server. Berikut perintah untuk menginstall roundcube

```
root@forkits:~# apt-get install roundcube
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  aspell aspell-en javascript-common libaspell15 libicu48 libjs-jquery
  libjs-jquery-ui php-auth php-auth-sasl php-mail-mime php-mail-mimedecode
  php-mdb2 php-mdb2-driver-mysql php-net-smtp php-net-socket php-pear
  php5-intl php5-pspell roundcube-core roundcube-mysql tinymce
  wwwconfig-common
Suggested packages:
  aspell-doc spellutils libjs-jquery-ui-docs php-log php-soap php5-dev
  php-crypt-gpg roundcube-plugins postgresql-client
The following NEW packages will be installed:
  aspell aspell-en javascript-common libaspell15 libicu48 libjs-jquery
  libjs-jquery-ui php-auth php-auth-sasl php-mail-mime php-mail-mimedecode
  php-mdb2 php-mdb2-driver-mysql php-net-smtp php-net-socket php-pear
  php5-intl php5-pspell roundcube roundcube-core roundcube-mysql tinymce
  wwwconfig-common
0 upgraded, 23 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/9064 kB of archives.
After this operation, 40.9 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 10.45 Installasi roundcube untuk web mail server

Berikut ada sebuah pertanyaan apakah kita menginginkan untuk membuat database pada proses installasi roundcube, pilih yes

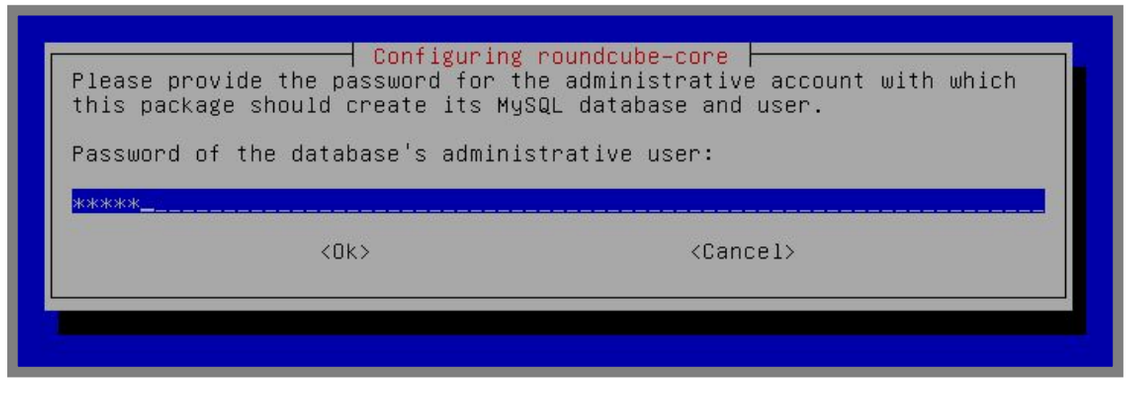


Gambar 10.46 Proses installasi roundcube

Pilih aplikasi database server yang digunakan, pada bab sebelumnya telah dibahas database server menggunakan mysql

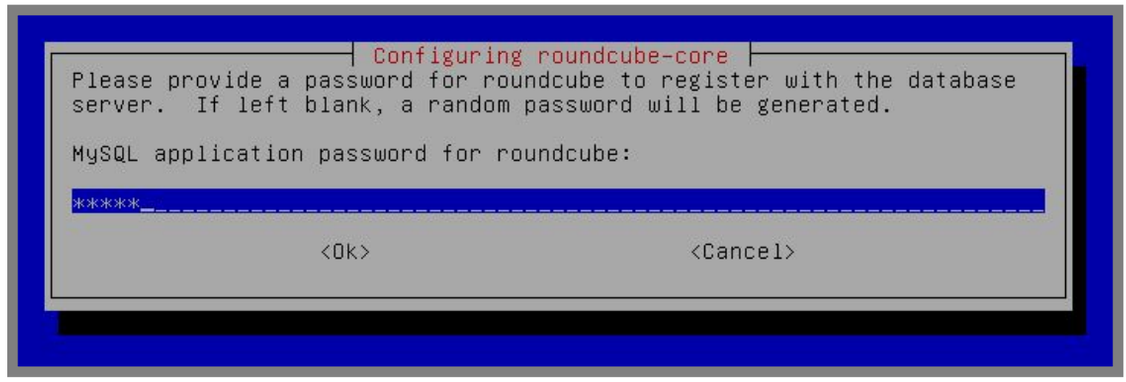
Gambar 10.47 Proses installasi roundcube

Masukkan password user root untuk login ke mysql



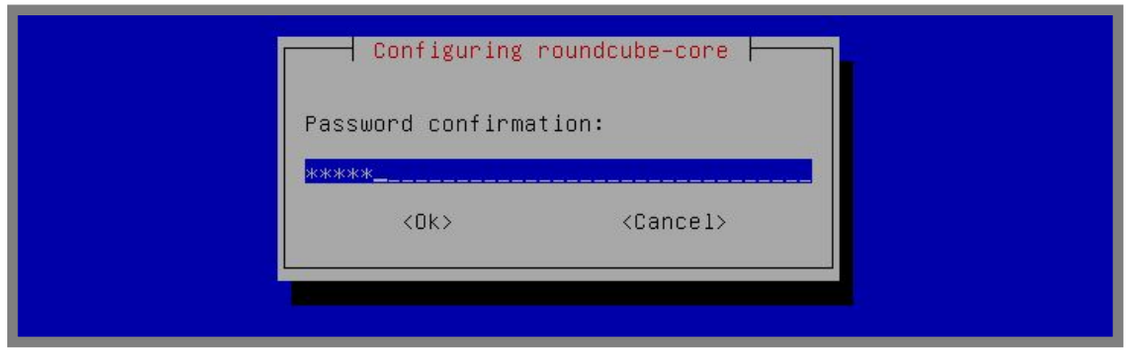
Gambar 10.48 Proses installasi roundcube

Masukkan password yang akan digunakan untuk database roundcube di mysql



Gambar 10.49 Proses installasi roundcube

Masukkan password database untuk roundcube sekali lagi



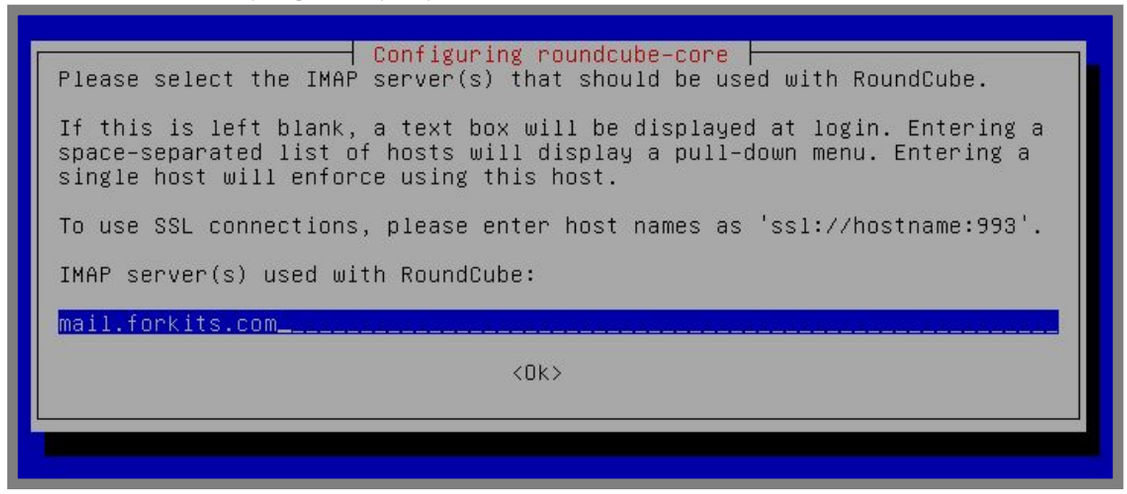
Gambar 10.50 Proses installasi roundcube

Langkah selanjutnya kita harus melakukan konfigurasi ulang pada roundcube

```
root@forkits:~# dpkg-reconfigure roundcube-core
```

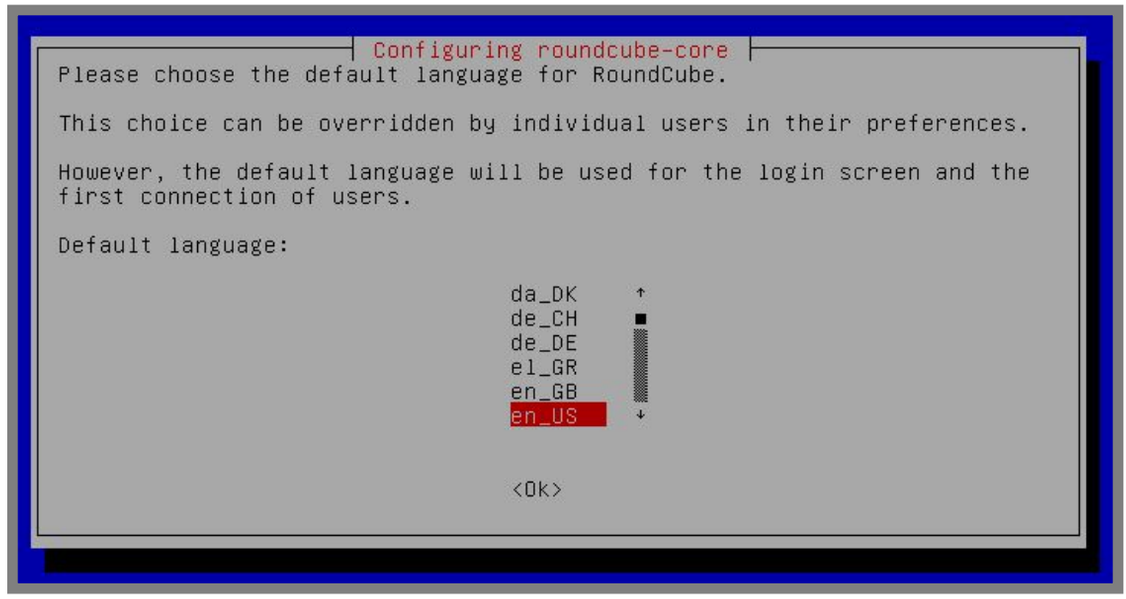
Gambar 10.51 Konfigurasi ulang roundcube

Masukkan domain yang mempunyai nilai MX terendah



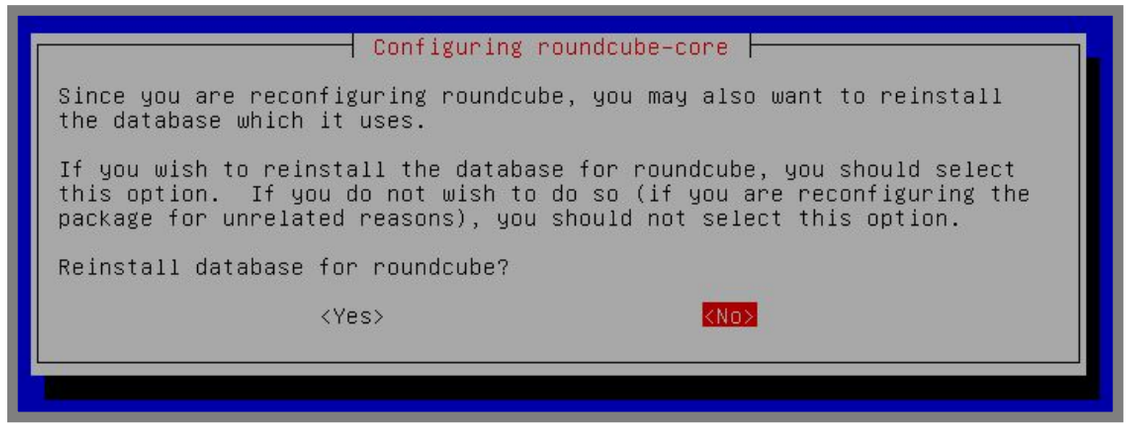
Gambar 10.52 Proses konfigurasi ulang roundcube

Pilih bahasa yang digunakan



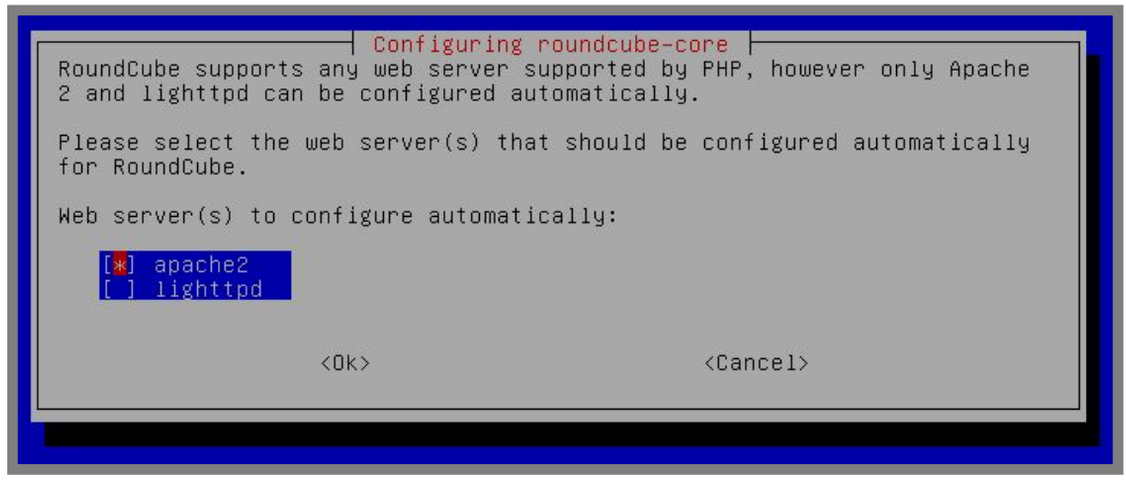
Gambar 10.53 Proses konfigurasi ulang roundcube

Jika ingin install ulang database untuk roundcube, pilih yes. Namun saya sarankan untuk memilih no saja



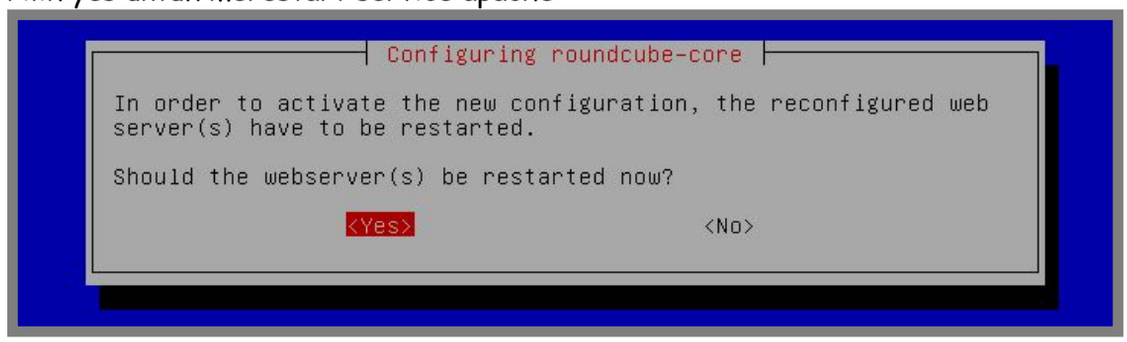
Gambar 10.54 Proses konfigurasi ulang roundcube

Pilih web server yang digunakan



Gambar 10.55 Proses konfigurasi ulang roundcube

Pilih yes untuk merestart service apache



Gambar 10.56 Proses konfigurasi ulang roundcube

Selanjutnya kita harus membuat sebuah virtualhost untuk web mail roundcube ini. Berikut konfigurasi yang perlu dilakukan

```
root@forkits:~# cd /etc/apache2/sites-available/
root@forkits:/etc/apache2/sites-available# cp default roundcube
root@forkits:/etc/apache2/sites-available# nano roundcube
<VirtualHost *:80>
    ServerAdmin admin@forkits.com
    ServerName mail.forkits.com
    DocumentRoot /usr/share/roundcube
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /usr/share/roundcube>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ..........................................................................
    ..................................................
    ..........................................................................
    ..............................................................
```

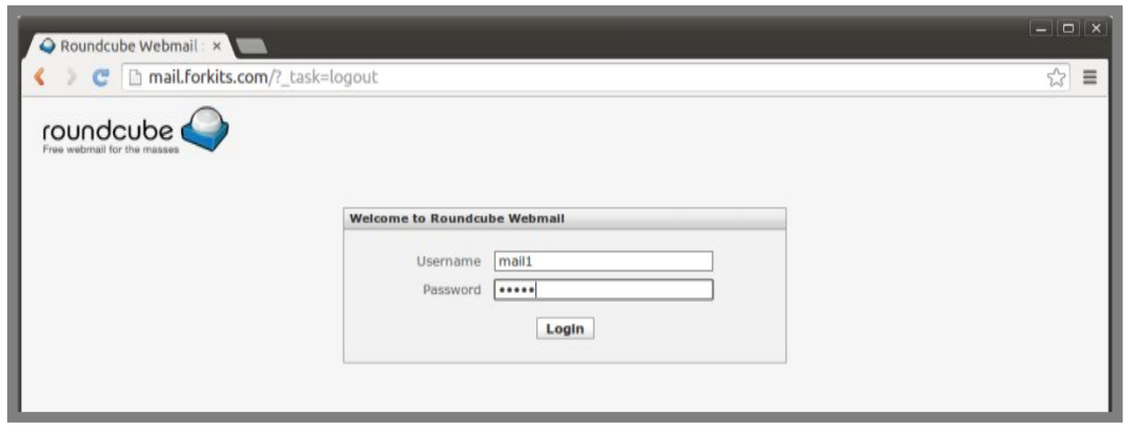
Gambar 10.57 Konfigurasi virtualhost untuk roundcube

Selanjutnya aktifkan virtualhost tersebut kemudian restart service apache

```
root@forkits:/etc/apache2/sites-available# a2ensite roundcube
Enabling site roundcube.
To activate the new configuration, you need to run:
  service apache2 reload
root@forkits:/etc/apache2/sites-available# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.
Server www.forkits.com:443 (RSA)
Enter pass phrase: (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:/etc/apache2/sites-available#
```

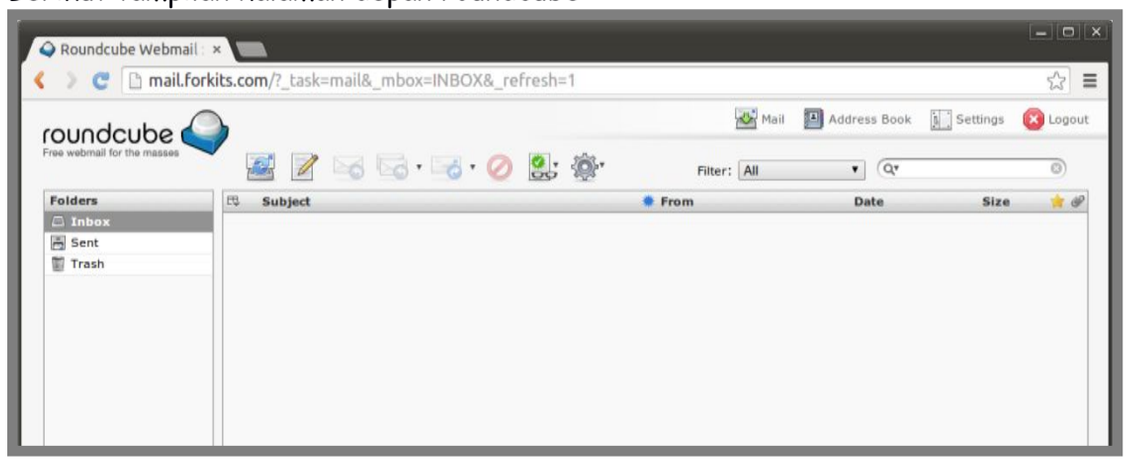
Gambar 10.58 Mengaktifkan virtualhost roundcube dan restart apache

Berikut pengujian yang dilakukan dari web browser client



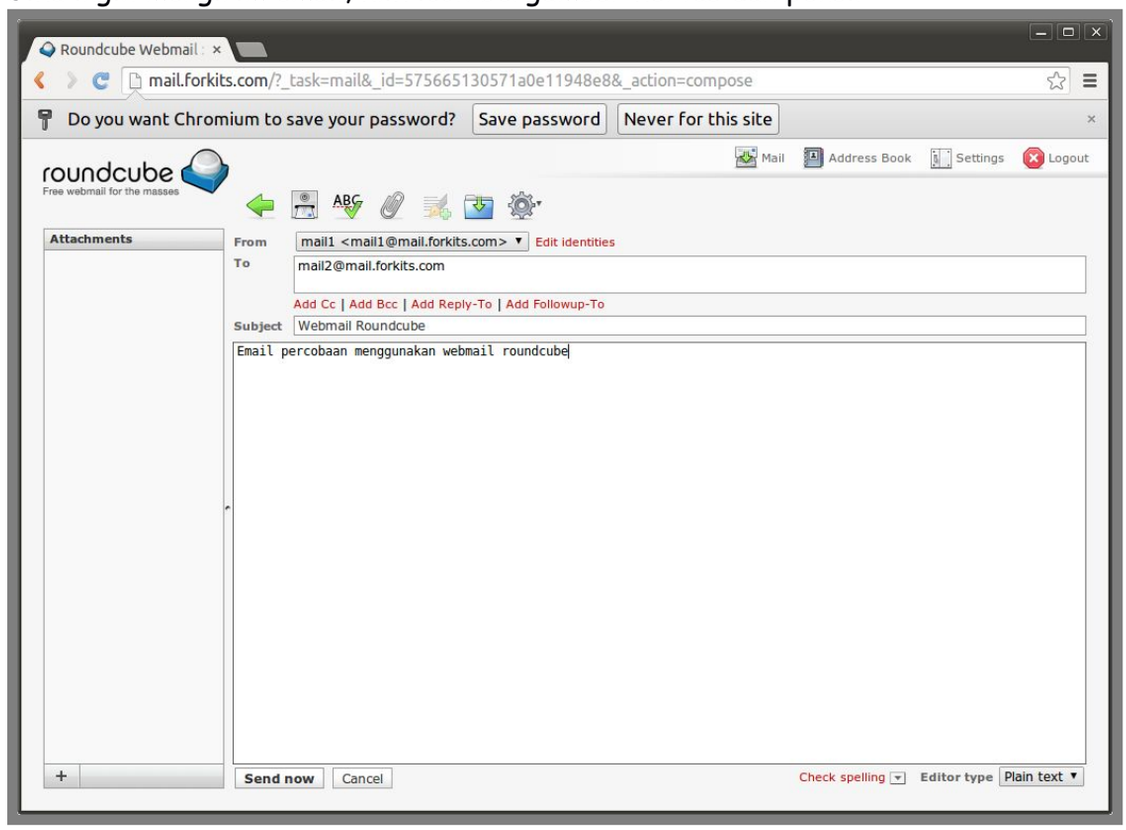
Gambar 10.59 Halaman login roundcube

Berikut tampilan halaman depan roundcube



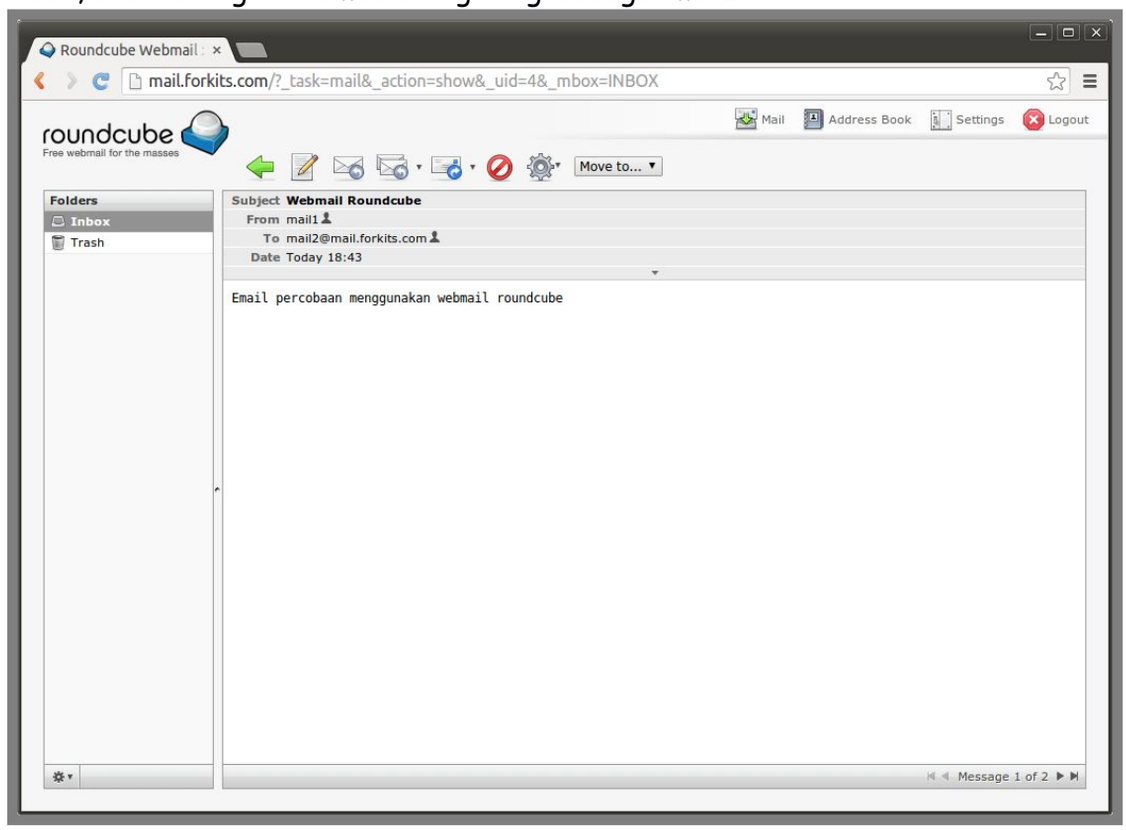
Gambar 10.60 Halaman utama roundcube

Jika ingin mengirim email, klik icon bergambar kertas dan pensil



Gambar 10.61 Mengirim email menggunakan roundcube

Untuk memastikan apakah email yang barusan kita tulis benar-benar terkirim atau tidak, silahkan logout kemudian login lagi sebagai mail2



Gambar 10.62 Melihat inbox pada user mail2

HTTPS pada Webmail Roundcube

Sama halnya dengan pembahasan https pada squirrelmail sebelumnya. Kita juga bisa mengaktifkan protocol https untuk mengakses roundcube. Untuk itu, kita harus membuat sebuah virtualhost ssl untuk roundcube. Namun karena kita tadi telah membuat virtualhost ssl untuk squirrelmail dan kita sudah tidak membutuhkan squirrelmail, kita bisa mengedit file tersebut.

```
root@forkits:/etc/apache2/sites-available# mv squirrelmail-ssl roundcube-ssl
root@forkits:/etc/apache2/sites-available# nano roundcube-ssl
<IfModule mod_ssl.c>
<VirtualHost 192.168.10.1:443>
        ServerAdmin admin@forkits.com
        ServerName mail.forkits.com
        DocumentRoot /usr/share/roundcube
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /usr/share/roundcube>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
        .....
        .....
        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2.2-common/README.Debian.gz for info
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/apache2/ssl/forkits.crt
        SSLCertificateKeyFile /etc/apache2/ssl/forkits.key
        .....
        .....
        .....
root@forkits:/etc/apache2/sites-available# a2ensite roundcube-ssl
Enabling site squirrelmail-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@forkits:/etc/apache2/sites-available#
```

Gambar 10.63 Konfigurasi virtualhost untuk roundcube

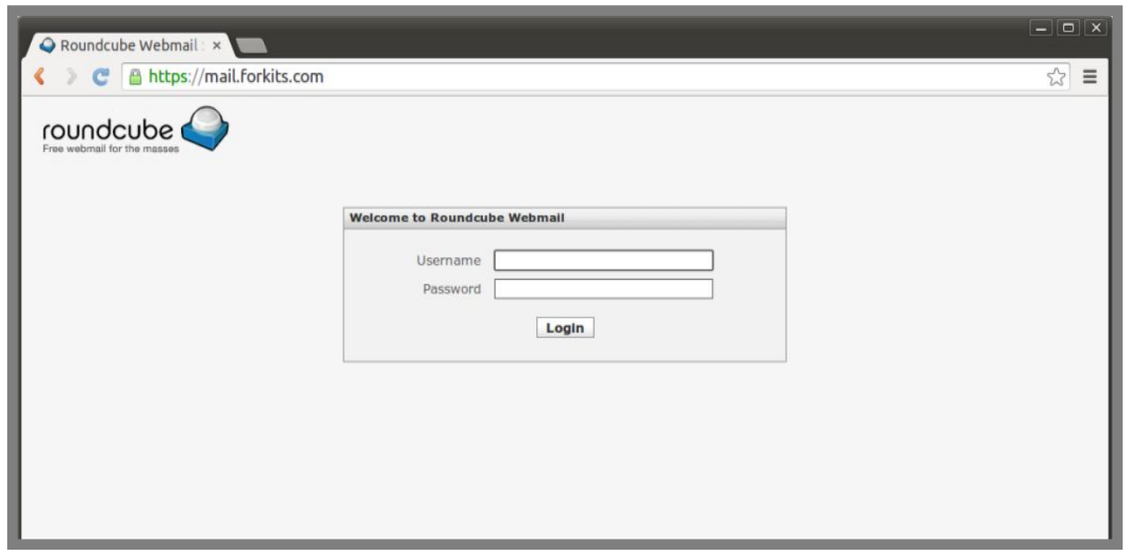
Perhatikan gambar diatas, terlihat bahwa kita merename virtualhost ssl milik squirrelmail menjadi virtualhost ssl milik roundcube. Kita juga melakukan sedikit

perubahan pada konfigurasi virtualhost tersebut. Setelah itu, dilanjutkan dengan perintah untuk mengaktifkan virtualhost tersebut. Selanjutnya kita harus merestart service apache

```
root@forkits:/etc/apache2/sites-available# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.
Server www.forkits.com:443 (RSA)
Enter pass phrase: (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:/etc/apache2/sites-available#
```

Gambar 10.64 Merestart service web server

Berikut pengujian yang dilakukan saat mengakses webmail dari client



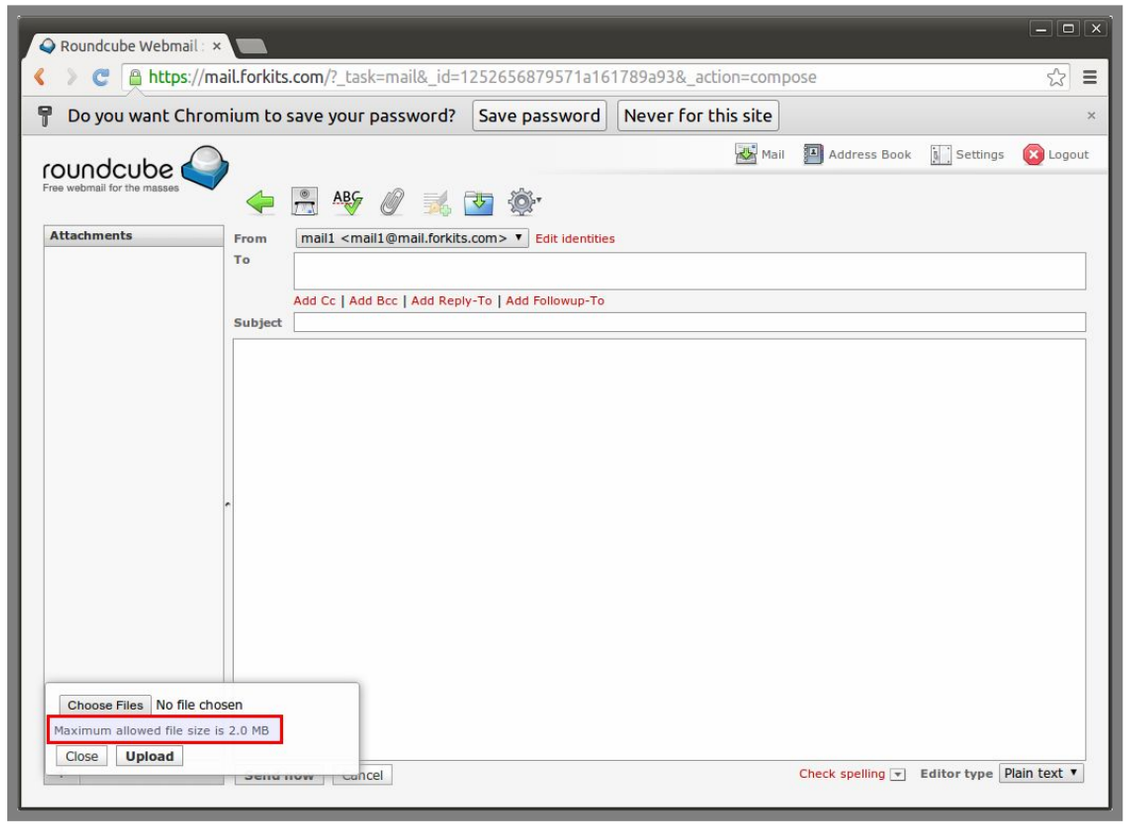
Gambar 10.65 Pengujian ssl pada webmail roundcube

Merubah Upload Maximum Mail Server

Ada sebuah fitur yang sangat sering kita gunakan saat mengirim email, yaitu melampirkan suatu file pada email yang kita kirim. File yang dilampirkan pun bervariasi, mulai dari file yang berukuran kecil hingga yang berukuran besar.

Secara default, mail server yang kita buat membatasi ukuran maksimum file yang bisa diupload, yaitu sebesar 2 MB. Hal ini tentu terlalu kecil, apalagi bagi pengguna yang terbiasa melampirkann file dengan ukuran puluhan MB.

Berikut gambar yang menunjukkan bahwa batas maximum file yang bisa diupload adalah 2 MB



Gambar 10.66 Percobaan upload file menggunakan email

Untuk mengatasi hal tersebut, berikut langkah-langkah yang perlu dilakukan

```
root@forkits:~# nano /etc/php5/apache2/php.ini
.....
.....
.....
; Maximum size of POST data that PHP will accept.
; Its value may be 0 to disable the limit. It is ignored if POST data reading
; is disabled through enable_post_data_reading.
; http://php.net/post-max-size
post_max_size = 50M
.....
.....
.....
; Maximum allowed size for uploaded files.
; http://php.net/upload-max-filesize
upload_max_filesize = 50M
.....
.....
.....
```

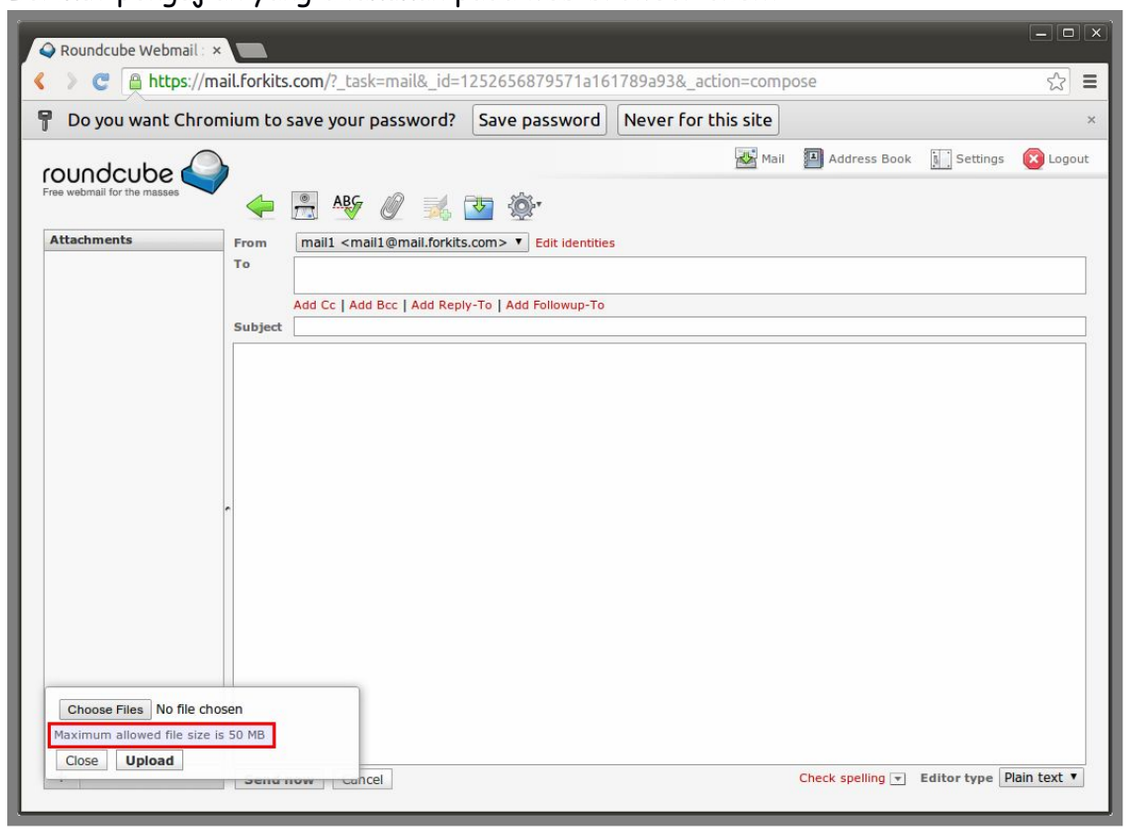
Gambar 10.67 Konfigurasi maximal upload pada email

Selanjutnya restart service apache

```
root@forkits:~# service apache2 restart
[....] Restarting web server: apache2 ... waiting Apache/2.2.22 mod_ssl/2.2.22
(Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.
Server www.forkits.com:443 (RSA)
Enter pass phrase: (tidak terlihat)

OK: Pass Phrase Dialog successful.
. ok
root@forkits:~#
```

Gambar 10.68 Merestart service web server

Berikut pengujian yang dilakukan pada web browser client



Gambar 10.69 Percobaan upload file menggunakan email

Perhatikan gambar diatas, terlihat bahwa batas maximum upload yang diperbolehkan telah berubah menjadi 50 MB, sesuai dengan yang kita konfigurasikan.

Konfigurasi User Quota

Semakin banyak user yang menggunakan mail server kita, tentu space harddisk yang kita butuhkan juga semakin banyak. Apalagi jika user terlalu serakah menggunakan space harddisk. Untuk menghindari user menggunakan space harddisk yang berlebihan, kita bisa melakukan konfigurasi quota untuk masing-masing user. Sehingga masing-masing user tidak bisa menyimpan/mengirim email yang mempunyai ukuran melebihi quota yang diberikan.

Pembahasan pada sub bab ini akan menggunakan partisi /home pada server. Pada bab installasi server, telah dijelaskan bagaimana cara membuat partisi /home. Satu hal yang sangat penting, partisi /home harus mempunyai tipe primary.

Langkah pertama yang harus dilakukan adalah memerubah konfigurasi fstab. Fstab adalah sebuah file yang mengatur bagaimana partisi pada server di mount. Berikut konfigurasi yang perlu dilakukan pada fstab

```
root@forkits:~# nano /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>

# / was on /dev/sda2 during installation
UUID=532e0227-89a5-4710-9024-0c455462d19a /    ext4    errors=re$

# /home was on /dev/sda6 during installation
#UUID=5e1ae42d-4603-482e-8507-231a8dfec881  /home   ext4 defaults$

# swap was on /dev/sda5 during installation
UUID=5a2bed55-8951-41ba-b0f1-18d96ed6c633    none    swap    sw$
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0

/dev/sda6   /home   ext4    defaults,usrquota   1   1
```

Gambar 10.70 Konfigurasi mount pada fstab

Perhatikan gambar diatas, terlihat bahwa ada sebuah kalimat yang menandakan bahwa partisi /home berada di /dev/sda6 (/home was on /dev/sda6 during installation). Perhatikan bahwa kita menonaktifkan baris konfigurasi untuk mounting partisi /home. Selanjutnya diahir konfigurasi, kita menambahkan sebuah baris untuk mounting partisi /home dengan dukungan fitur user quota.

Selanjutnya kita harus membuat file yang digunakan untuk menyimpan record quota

```
root@forkits:~# cd /home/
root@forkits:/home# touch aquota.user
root@forkits:/home# chmod 600 aquota.user
root@forkits:/home# reboot

Broadcast message from root@forkits (pts/0) (Fri Apr 22 21:01:18 2016):

The system is going down for reboot NOW!
root@forkits:/home#
```

Gambar 10.72 Membuat file yang dibutuhkan oleh usrquota

Perhatikan gambar diatas, terlihat bahwa setelah membuat file-file yang diperlukan, kita merestart server. Selanjutnya install aplikasi quota dengan perintah berikut

```
root@forkits:~# apt-get install quota quotatool
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  dbus libdbus-1-3 libnl-3-200 libnl-genl-3-200 libsystemd-login0
Suggested packages:
  dbus-x11 libnet-ldap-perl
The following NEW packages will be installed:
  dbus libdbus-1-3 libnl-3-200 libnl-genl-3-200 libsystemd-login0 quota
  quotatool
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/1347 kB of archives.
After this operation, 3265 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 10.73 Installasi aplikasi untuk quota

Selanjutnya kita bisa menentukan quota untuk masing-masing user sesuai yang diinginkan. Untuk menentukan quota, kita bisa menggunakan parameter soft limit dan hard limit.

Soft limit adalah batas yang masih bisa dilewati oleh user, namun hanya akan disimpan dalam jangka waktu tertentu (default 7 hari). Sedangkan hard limit adalah batas yang tidak bisa dilewati oleh user, jika dilewati maka akan ada pesan error.

Berikut contoh perintah yang digunakan untuk konfigurasi quota pada user mail2 dengan soft limit sebesar 5 MB dan hard limit sebesar 10 MB.

```
root@forkits:/home# quotatool -u mail2 -bq 5M -l '10 MB' /home
```

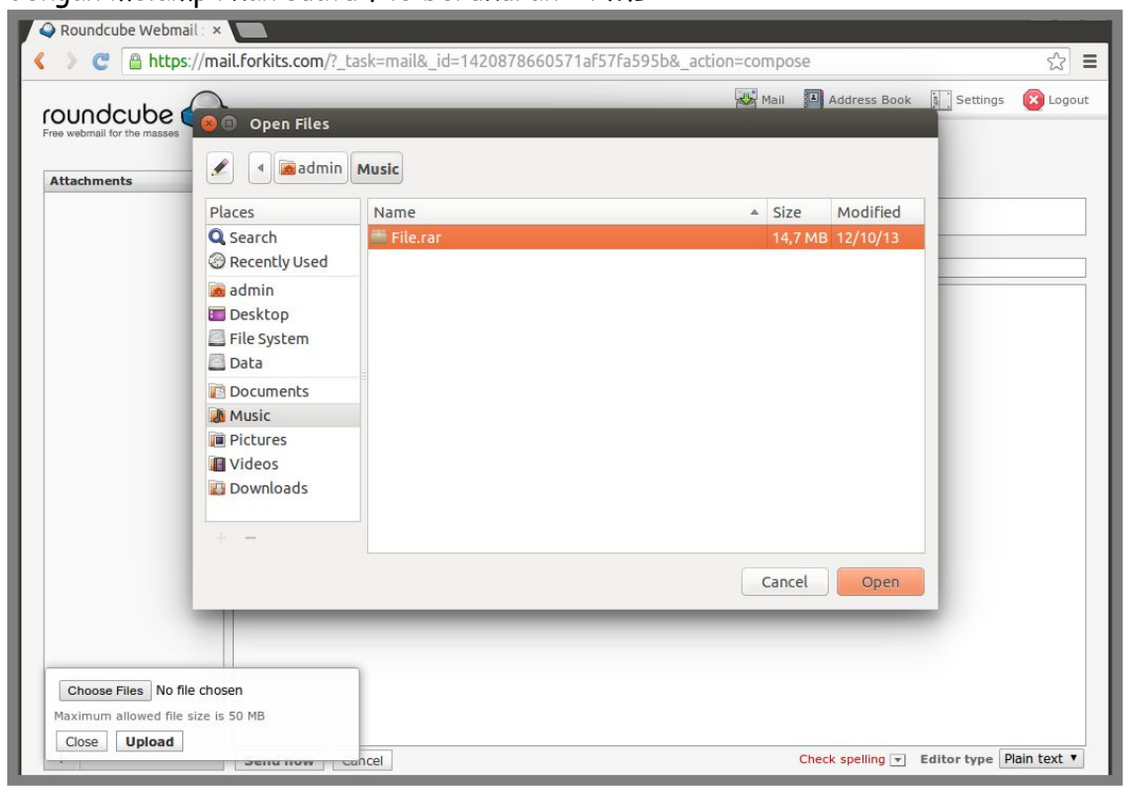
Gambar 10.74 Konfigurasi quota untuk user mail2

Untuk melihat hasil konfigurasi quota diatas, kita bisa menggunakan perintah berikut

```
root@forkits:/home# repquota /home
*** Report for user quotas on device /dev/sda2
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User                    used    soft    hard  grace used  soft  hard  grace
----------------------------------------------------------------------------
root            --      20       0       0        2     0     0
forkits         --      16       0       0        4     0     0
administrator   --      16       0       0        4     0     0
mail1           --      16       0       0        4     0     0
mail2           --      16    5120   10240        4     0     0

root@forkits:/home#
```

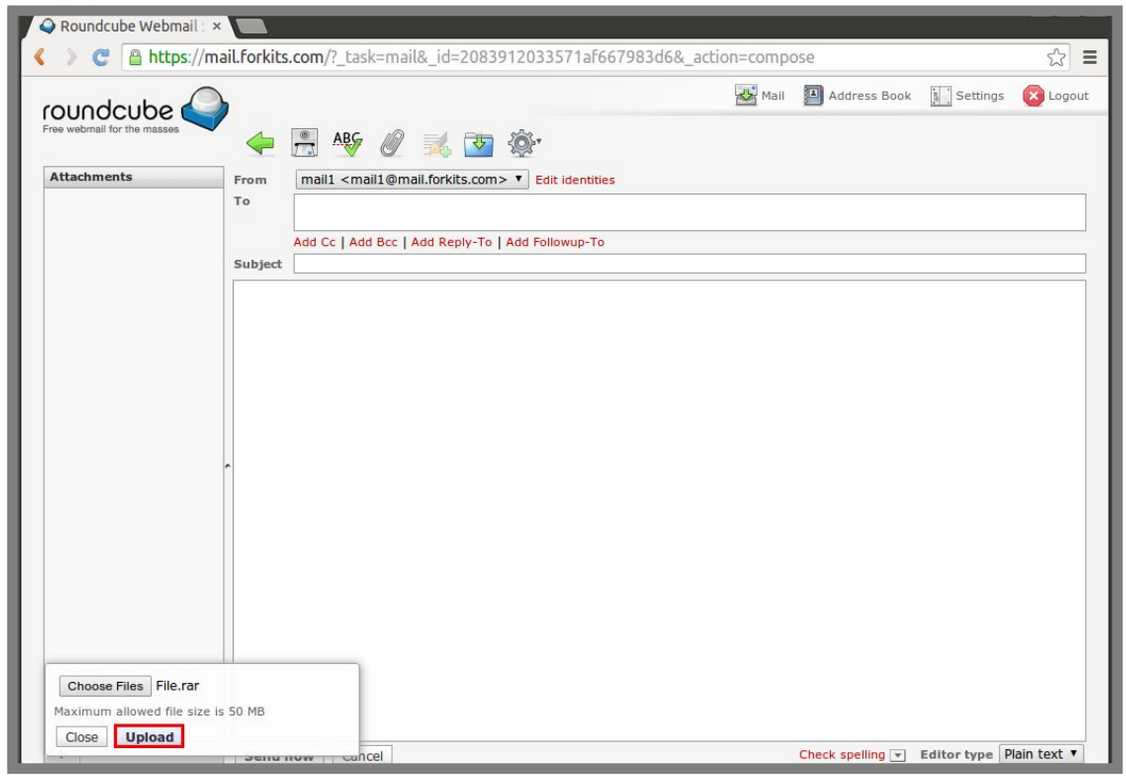
Gambar 10.75 Melihat quota yang telah diterapkan

Untuk melakukan pengujian, kita akan mencoba mengirim email dari mail1 ke mail2 dengan melampirkan suatu file berukuran 14 MB



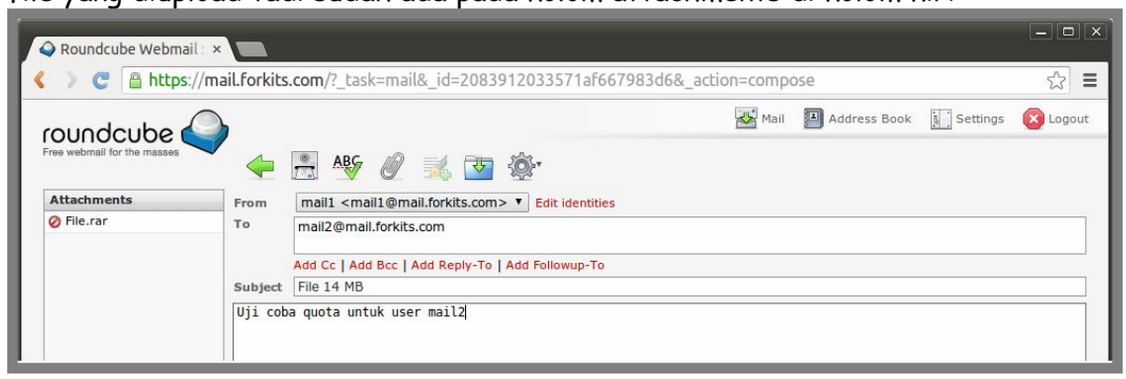
Gambar 10.76 Percobaan upload file melebihi batas quota

Setelah memilih file yang ingin diupload, jangan lupa klik tombol upload untuk mengupload file tersebut



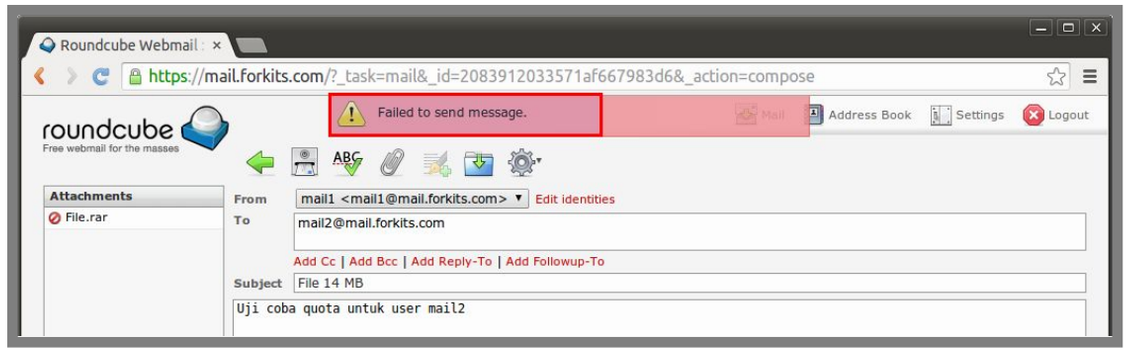
Gambar 10.77 Mengupload file

Selanjutnya tulis tujuan, subjek, dan teks emal yang diinginkan, pastikan bahwa file yang diupload tadi sudah ada pada kolom attachments di kolom kiri



Gambar 10.78 Menulis tujuan email dan subject

Berikut hasil setelah klik tombol send (mengirim email)



Gambar 10.79 Gagal mengirim email

Perhatikan gambar diatas, terlihat bahwa ada sebuah peringatan bahwa email tidak bisa dikirim. Hal ini dikarenakan kita mengirim file dengan ukuran melebihi quota dari user mail2.

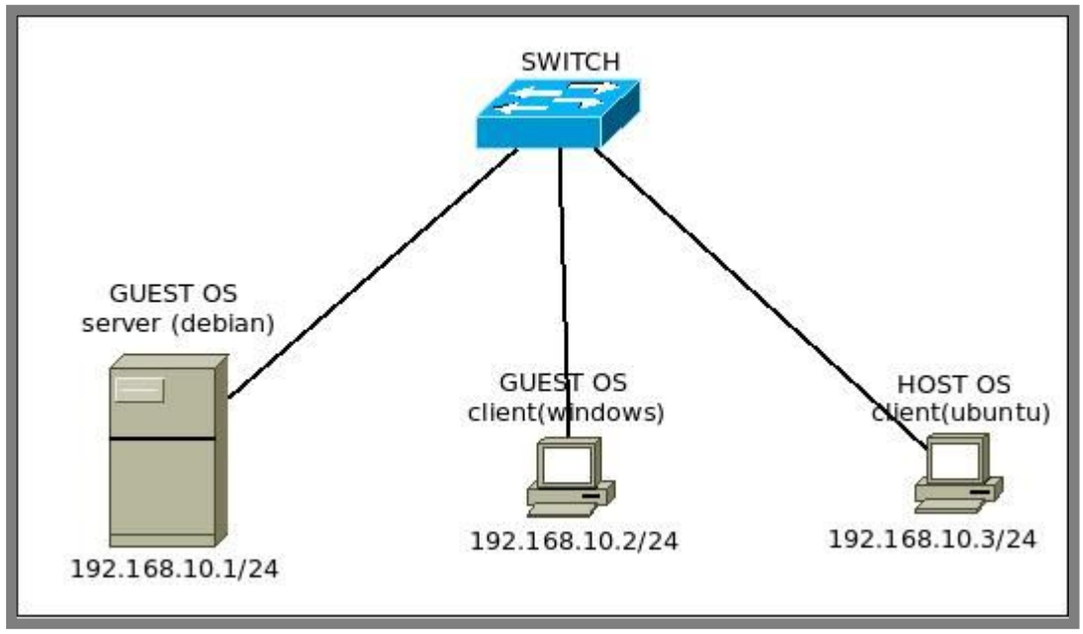
---END OF CHAPTER---

Bab 11
File Sharing Server

File sharing server digunakan untuk melakukan sharing file atau data dalam jaringan skala kecil, atau sering disebut Local Area Coneection (LAN). Dengan memanfaatkan file sharing server, kita bisa melakukan pengolohan data secara terpusat dalam suatu jaringan.

Ada beberapa protocol yang bisa kita manfaatkan untuk membuat file sharing server. Diantaranya yang paling populer adalah samba dan nfs. Pada bab ini kita akan membahas konfigurasi file sharing server menggunakan kedua protocol tersebut.

Pada bab ini, kita akan menggunakan topologi sebagai berikut



Gambar 11.1 Topologi jaringan untuk praktik file sharing server

Perhatikan topologi diatas, terlihat bahwa kita menggunakan dua komputer client yang memiliki sistem operasi berbeda, yaitu windows dan ubuntu. Hal ini dikarenakan ada perbedaan yang cukup jauh antara konfigurasi client di windows dan linux.

Diasumsikan bahwa server dan client sudah dikonfigurasi ip address sesuai topologi ditas dan sudah bisa saling berkomunikasi.

Konfigurasi Samba dengan User Authentication

Samba merupakan sebuah protocol file sharing server yang bisa digunakan pada sebuah jaringan dengan sistem operasi beragam. Dalam artian, protocol ini bisa kita gunakan pada sistem operasi linux maupun windows.

Pada sub bab ini kita akan membahas konfigurasi samba dengan menggunakan user authentication. Selanjutnya akan dibahas konfigurasi samba dengan anonymous login pada sub bab berikutnya.

Hal pertama yang bisa kita lakukan adalah membuat direktori atau folder yang akan di share. Sebagai contoh kasus, kita ditugaskan untuk membuat file sharing dengan nama internal yang berada didalam direktori /samba

```
root@forkits:~# mkdir /samba
root@forkits:~# mkdir /samba/internal
root@forkits:~# ls -l /samba/
total 4
drwxr-xr-x 2 root root 4096 Apr 23 17:11 internal
root@forkits:~# chmod 777 /samba/internal/
root@forkits:~# ls -l /samba/
total 4
drwxrwxrwx 2 root root 4096 Apr 23 17:11 internal
root@forkits:~#
```

Gambar 11.2 Membuat direktory yang akan disharing

Berikut penjelasan dari masing-masing perintah diatas

Syntak	Deskripsi
mkdir /samba	Digunakan untuk membuat direktori samba
mkdir /samba/internal	Digunakan untuk membuat direktori yang akan digunakan untuk sharing dengan nama internal
ls -l /samba	Digunakan untuk melihat isi direktori /samba dengan detail. Perhatikan bahwa terdapat direktori yang baru saja kita buat. Perhatikan bahwa hak akses direktori yang baru kita buat tersebut adalah read & write bagi user root saja, sedangkan untuk user selain root hanya memiliki hak akses read only
chmod 777 /samba/internal	Digunakan untuk merubah hak akses direktori internal menjadi read write untuk siapapun. Perintah ini opsional, artinya bisa kita gunakan hanya jika kita menginginkannya
ls -l	Perhatikan bahwa saat ini direktori internal sudah memiliki hak akses read & write untuk semua user

Langkah selanjutnya yang harus kita lakukan adalah menginstall aplikasi samba. Berikut perintah yang bisa kita gunakan

```
root@forkits:~# apt-get install samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2
  libfile-copy-recursive-perl libtalloc2 libtdb1 libwbclient0 samba-common
  samba-common-bin tdb-tools update-inetd
Suggested packages:
  cups-common openbsd-inetd inet-superserver smbldap-tools ldb-tools ctdb
The following NEW packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2
  libfile-copy-recursive-perl libtalloc2 libtdb1 libwbclient0 samba
  samba-common samba-common-bin tdb-tools update-inetd
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/9028 kB of archives.
After this operation, 44.9 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 11.3 Installasi aplikasi samba untuk file server

Selanjutnya tambahkan user yang akan digunakan untuk login ke file sharing samba

```
root@forkits:~# useradd sharing
root@forkits:~# smbpasswd -a sharing
New SMB password: (tidak terlihat)
Retype new SMB password: (tidak terlihat)
Added user sharing.
root@forkits:~#
```

Gambar 11.4 Menambahkan user untuk samba

Perhatikan gambar diatas, terlihat bahwa kita menambahkan user dengan nama sharing. Dilanjutkan dengan sebuah perintah untuk memberikan password samba pada user tersebut. Password ini bisa berbeda dengan password yang digunakan user untuk login ke sistem operasi.

Selanjutnya kita harus melakukan beberapa konfigurasi pada samba. Berikut konfigurasi yang perlu dilakukan untuk membuat file sharing samba menggunakan user authentication

```
root@forkits:~# nano /etc/samba/smb.conf
.....
.....
.....
######## Authentication ########

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server. See
# /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/ServerType.html
# in the samba-doc package for details.
security = user
.....
.....
.....
#======================= Share Definitions =======================
[internal]
    path           = /samba/internal
    browseable     = yes
    writeable      = yes
    valid users    = sharing
    admin users    = root
.....
.....
.....
```

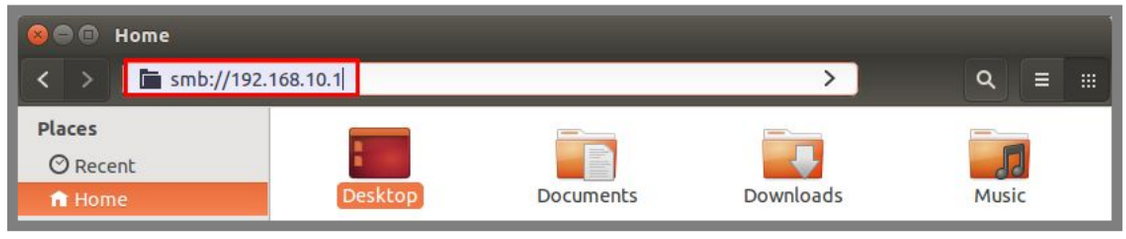
Gambar 11.5 Konfigurasi samba

Rubah atau tambahkan konfigurasi pada bagian yang ditandai teks warna hijau. Selanjutnya restart service samba

```
root@forkits:~# service samba restart
[ ok ] Stopping Samba daemons: nmbd smbd.
[ ok ] Starting Samba daemons: nmbd smbd.
root@forkits:~#
```

Gambar 11.6 Restart service samba

Sebelum melakukan pengujian, pastikan client ubuntu sudah terinstall aplikasi *smbclient*. Untuk mengakses samba file sharing, gunakan tombol kombinasi ctrl+l pada file explorer kemudian ketikkan smb://ip_server pada address bar.

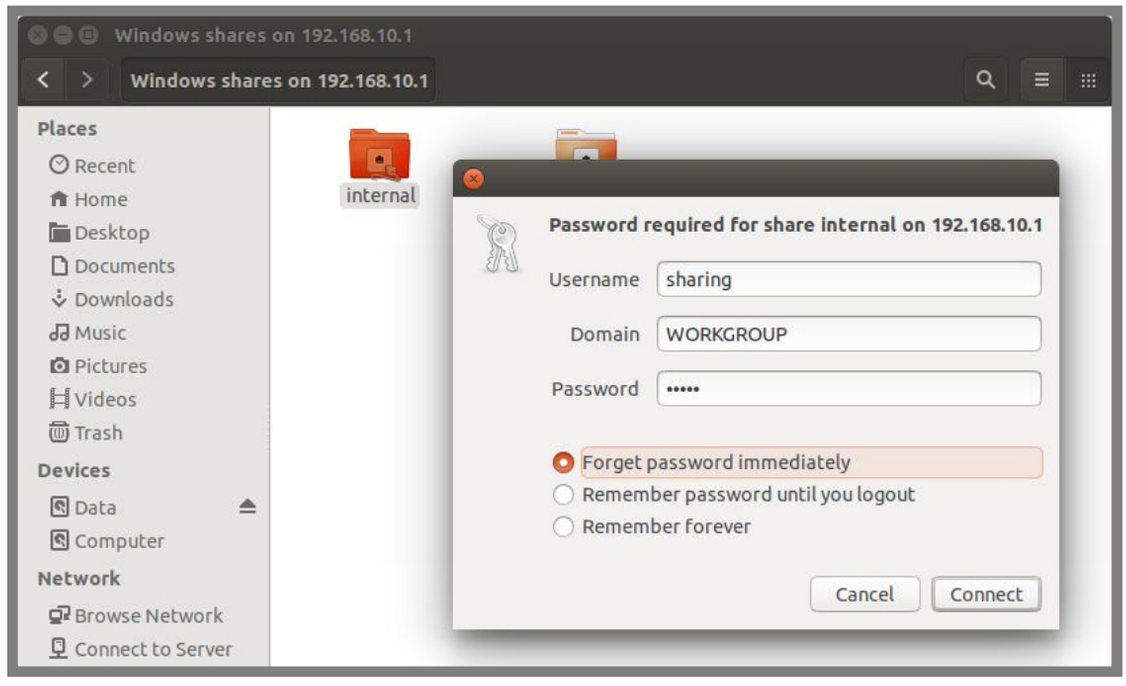


Gambar 11.7 Mengakses file server dari client ubuntu

Berikut tampilan setelah kita melakukan langkah diatas

Gambar 11.8 Mengakses file server dari client ubuntu

Saat kita mencoba mengakses folder atau direktori internal, akan diminta untuk memasukkan username dan password

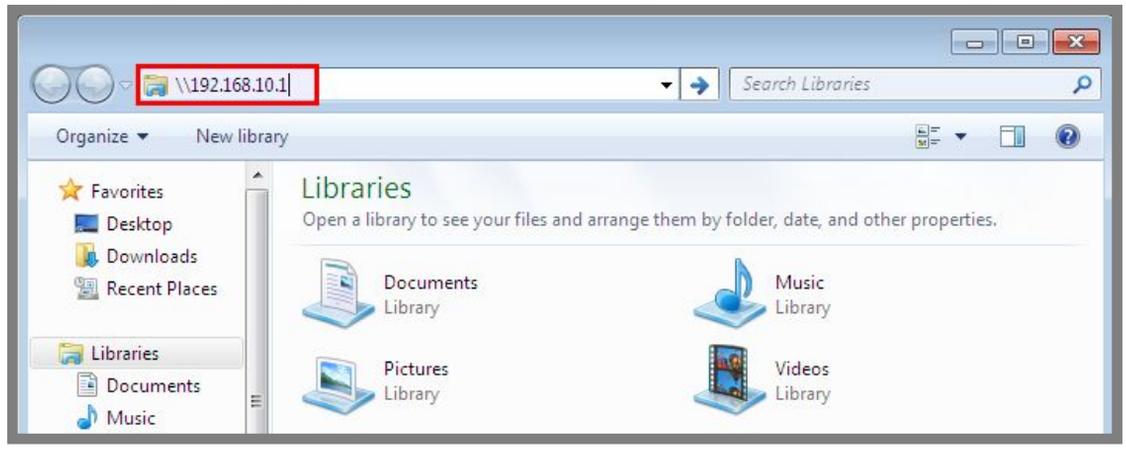


Gambar 11.9 Mengakses file server dari client ubuntu

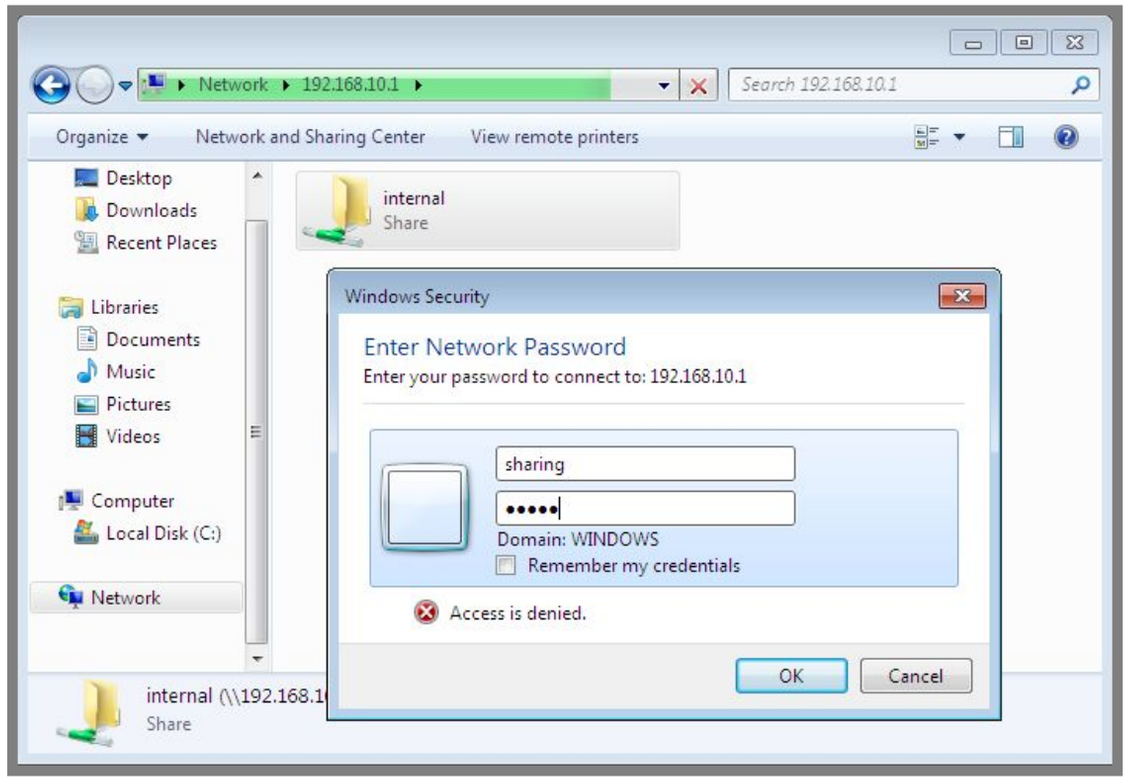
Karena permission yang ada pada direktori internal adalah read write untuk semua user, seharusnya kita bisa membuat sebuah direktori didalamnya

Gambar 11.10 Mencoba membuat direktori pada share internal

Jika kita menggunakan sistem operasi windows sebagai client, maka berikut langkah-langkah yang perlu dilakukan untuk mengakses samba file sharing

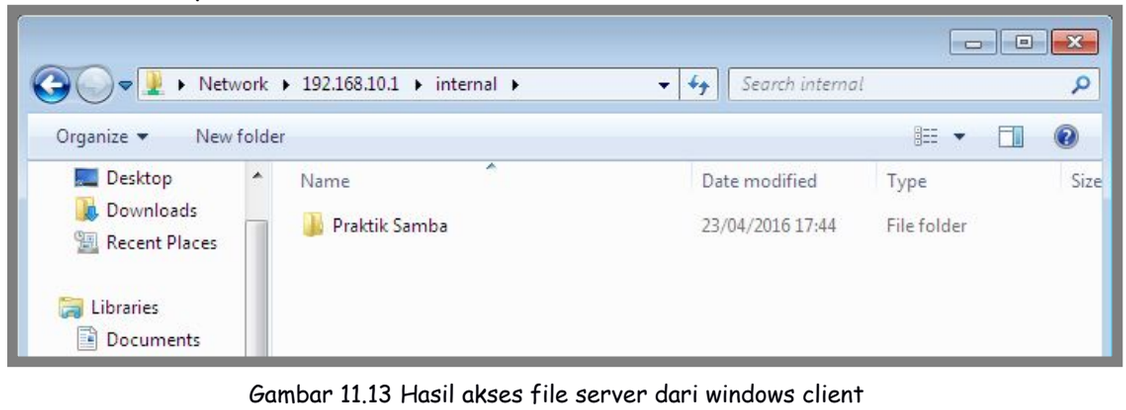


Gambar 11.11 Mengakses file server dari cleint windows



Gambar 11.12 Memasukkan username dan password samba

Berikut hasilnya



Gambar 11.13 Hasil akses file server dari windows client

Konfigurasi Samba dengan Anonymous Login

Pada sub bab sebelumnya telah dibahas konfigurasi samba dengan menggunakan authentication login, pada sub bab ini akan dibahas konfigurasi samba dengan anonymous login (tanpa user).

Sebagai contoh kasus kita akan membuat direktori public dan melakukan share direktori tersebut dengan anonymous login

```
root@forkits:~# mkdir /samba/public
root@forkits:~# mkdir /samba/public/Software
root@forkits:~# mkdir /samba/public/Game
root@forkits:~# mkdir /samba/public/Video
root@forkits:~# ls -l /samba/
total 8
drwxrwxrwx 3 root root 4096 Apr 23 17:44 internal
drwxr-xr-x 5 root root 4096 Apr 23 18:05 public
root@forkits:~#
```

Gambar 11.14 Membuat direktori untuk samba

Perhatikan gambar diatas, terlihat bahwa kita membuat direktori public dan beberapa direltori lain didalamnya. Direktori-direktori tambahan (Software, Game, dan Video) hanya digunakan untuk keperluan pengujian saja. Selanjutnya perhatikan perintah terhari, terlihat bahwa direktori public mempunyai permission read only untuk user selain root. Jadi kita nanti tidak mungkin bisa membuat atau merubah isi direktori public pada file sharing.

Selanjutnya lakukan konfigurasi pada samba

```
root@forkits:~# nano /etc/samba/smb.conf
.....
.....
######## Authentication ########
security = share
.....
.....
#======================= Share Definitions =======================
[public]
    path          = /samba/public
    browseable    = yes
    guest ok      = yes
.....
.....
```

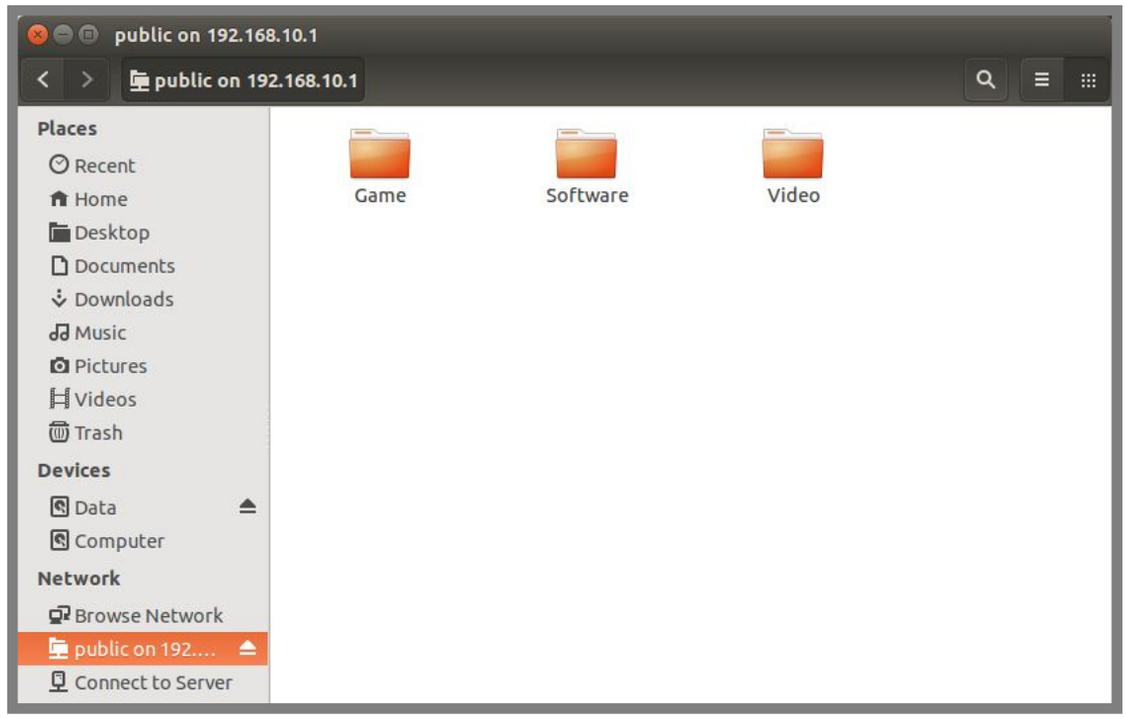
Gambar 11.15 Konfigurasi anonymous login samba

Selanjutnya restart service samba

```
root@forkits:~# service samba restart
[ ok ] Stopping Samba daemons: nmbd smbd.
[ ok ] Starting Samba daemons: nmbd smbd.
root@forkits:~#
```

Gambar 11.16 Restart service samba

Berikut pengujian yang dilakukan pada client ubuntu



Gambar 11.17 Pengujian dari client

Untuk melakukan pengujian di client windows, langkah-langkah yang perlu dilakukan sama saja dengan pengujian sebelumnya.

Kombinasi Authentication & Anonymous Login Samba

Pada sub bab sebelumnya, kita telah membahas konfigurasi samba dengan authentication maupun anonymous login. Pada sub bab ini, kita akan konfigurasi samba agar sharing dua direktori, direktori pertama akan menggunakan authentication login, sedangkan direktori kedua akan menggunakan anonymous login. Dalam arti lain bab ini akan membahas konfigurasi samba dengan kombinasi antara authentication dan anonymous login.

Sebagai contoh kasus, kita akan melakukan share direktori internal dan public yang telah kita buat sebelumnya. Dengan ketentuan direktori internal akan menggunakan authentication login dan direktori public akan menggunakan anonymous login.

Berikut konfigurasi yang perlu dilakukan

```
root@forkits:~# nano /etc/samba/smb.conf
.....
.....
.....
######## Authentication ########

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server. See
# /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/ServerType.html
# in the samba-doc package for details.
security = user
.....
.....
.....
#======================= Share Definitions =======================
[public]
    path            = /samba/public
    browseable      = yes
    guest ok        = yes

[internal]
    path            = /samba/internal
    browseable      = yes
    writeable       = yes
    valid users     = sharing
    admin users     = root
.....
.....
.....
```

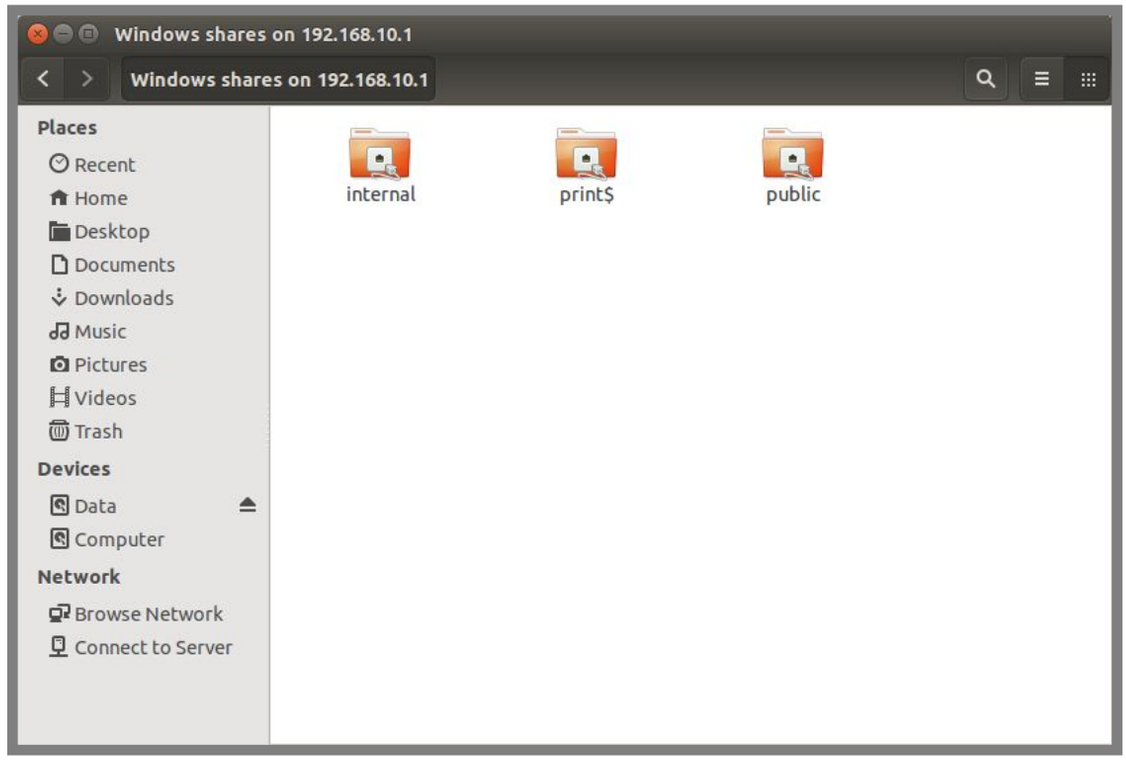
Gambar 11.18 Konfigurasi samba

Selanjutnya restart service samba

```
root@forkits:~# service samba restart
[ ok ] Stopping Samba daemons: nmbd smbd.
[ ok ] Starting Samba daemons: nmbd smbd.
root@forkits:~#
```

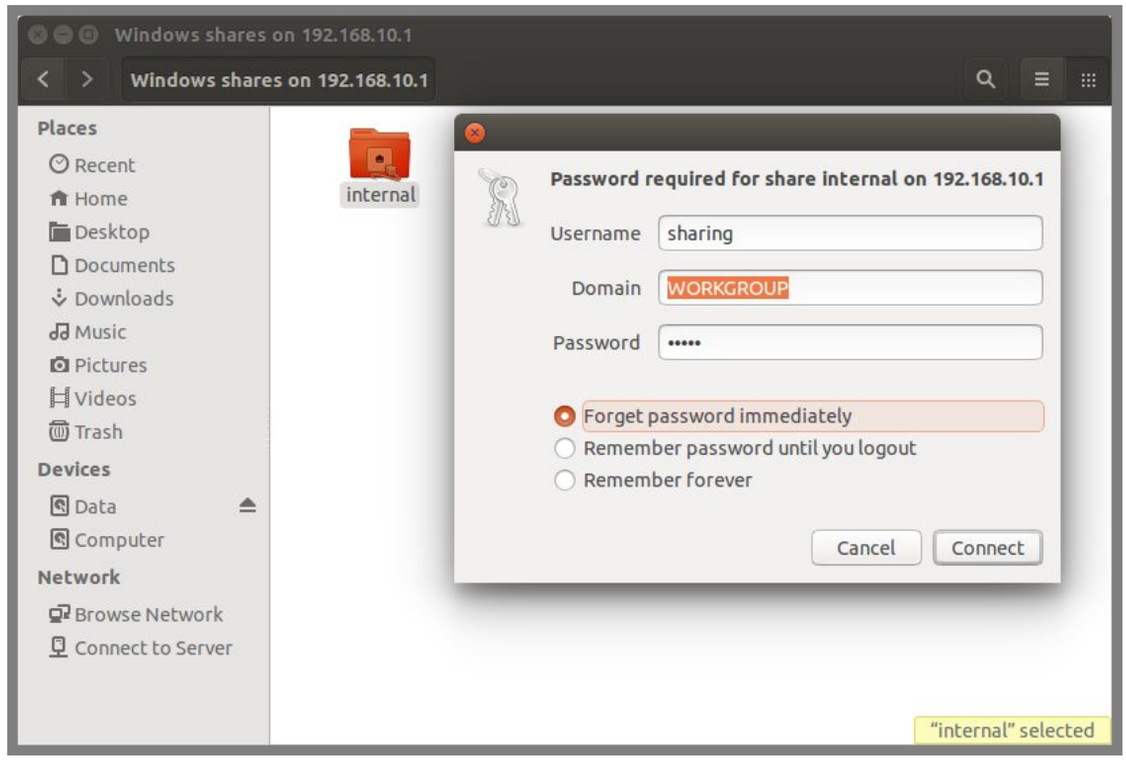
Gambar 11.19 Restart service samba

Berikut pengujian yang dilakukan pada client ubuntu



Gambar 11.20 Mengakses file server dari client ubuntu

Perhatikan gambar diatas, terlihat bahwa ada dua direktori yang disharing, yaitu internal dan public. Jika kita mencoba mengakses direktori public, maka akan diminta memasukkan username dan password seperti berikut



Gambar 11.21 Memasukkan username dan password untuk samba

Namun jika kita mengakses direktori public, kita tidak akan diminta untuk memasukkan username dan password

Gambar 11.22 List direktori yang dishare oleh samba

Mounting Samba Folder on Boot di Ubuntu

Ada suatu keadaan dimana kita diharuskan untuk selalu mengakses folder yang disharing oleh server. Tentu sangat merepotkan jika harus melakukan pekerjaan yang sama berulang-ulang untuk mengakses folder yang disharing oleh server. Oleh karena itu, kita bisa melakukan sebuah konfigurasi pada client yang tujuannya adalah untuk mount folder yang disharing oleh server secara otomatis setiap kali client dinyalakan.

Untuk melakukan hal seperti yang dijelaskan diatas, kita perlu menginstall sebuah aplikasi untuk melakukan mount di ubuntu. Berikut perintah yang bisa kita gunakan untuk menginstall aplikasi tersebut

```
admin@ubuntu:~$ sudo apt-get install cifs-utils
[sudo] password for admin: (tidak terlihat)
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  keyutils libnss-winbind libpam-winbind winbind
The following NEW packages will be installed:
  cifs-utils keyutils libnss-winbind libpam-winbind winbind
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/3358 kB of archives.
After this operation, 15.5 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 11.23 Installasi palikasi cifs untuk mounting di client

Selanjutnya, untuk melihat folder atau direktori yang dishare oleh server, kita bisa menggunakan perintah seperti berikut

```
admin@ubuntu:~$ smbclient -L 192.168.10.1
Enter admin's password: (enter)
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.6.6]

    Sharename       Type      Comment
    ---------       ----      -------
    public          Disk
    internal        Disk
    print$          Disk      Printer Drivers
    IPC$            IPC       IPC Service (forkits server)
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.6.6]

    Server               Comment
    ---------            -------
    FORKITS              forkits server

    Workgroup            Master
    ---------            -------
    WORKGROUP            FORKITS
admin@ubuntu:~$
```

Gambar 11.24 Melihat direktori yang dishare oleh server

Perhatikan gambar diatas, terlihat bahwa saat diminta untuk memasukkan password, kita langsung menekan enter. Hal ini dikarenakan untuk melihat folder atau direktori yang disharing oleh server kita cukup menggunakan user anonymous.

Selanjutnya berikut langkah yang perlu dilakukan untuk mount folder atau direktori yang disharing oleh server secara otomatis

```
admin@ubuntu:~$ sudo mkdir /mnt/pub
admin@ubuntu:~$ sudo mkdir /mnt/int
admin@ubuntu:~$ sudo nano /etc/fstab
.....
.....
.....
//192.168.10.1/public    /mnt/pub    cifs   auto,noexec    0   0
//192.168.10.1/internal  /mnt/int    cifs   auto,noexec,username=sharing,passwrord=** 0   0
```

Gambar 11.25 Konfigurasi fstab pada client

Perhatikan gambar diatas, terlihat bahwa kita membuat dua direktori di /mnt untuk tempat mounting folder yang dishrare oleh server. Selanjutnya kita menambahkan dua baris konfigurasi didalam fstab. Sesuaikan ip address, username, dan password dengan konfigurasi yang telah dilakukan pada samba

Selanjutnya untuk melakukan pengujian kita harus merestart client. Namun karena client yang kita gunakan adalah host os, dan jika kita restart maka server kita (guest os) juga akan mati, maka kita hanya akan menggunakan perintah remount berikut.

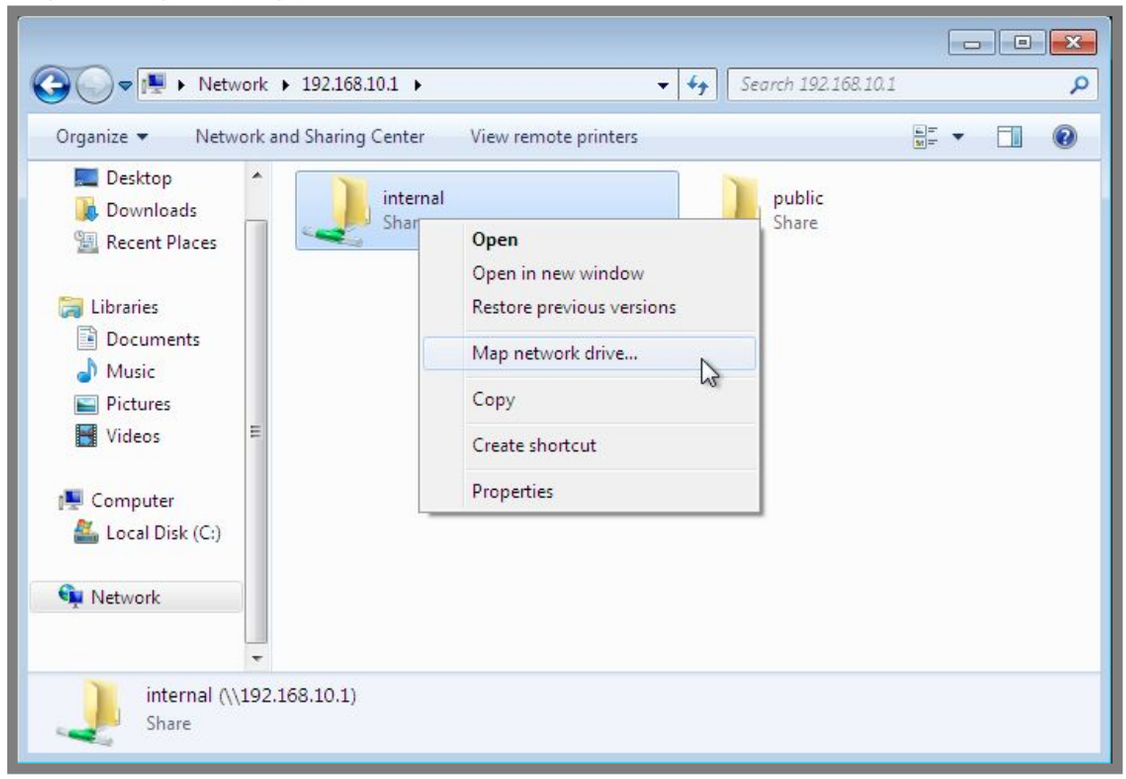
```
admin@ubuntu:~$ sudo mount -a
Password: (tidak terlihat)
admin@ubuntu:~$ sudo ls /mnt/int/
Praktik Samba
admin@ubuntu:~$ sudo ls /mnt/pub/
Game  Software  Video
admin@ubuntu:~$
```

Gambar 11.26 Melihat hasil konfigurasi mounting di client

Perhatikan gambar diatas, terlihat bahwa setelah kita mengetikkan perintah pertama, maka folder yang disharing oleh server sudah termount pada direktori yang ditentukan.

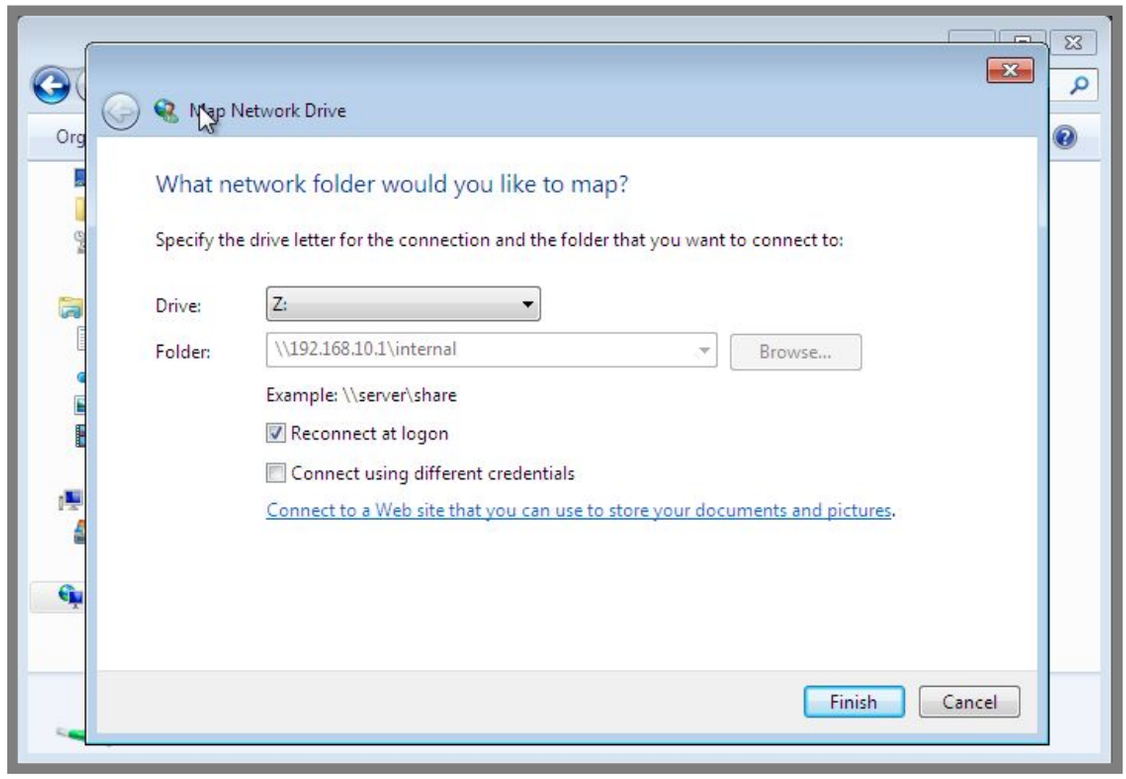
Mounting Samba Folder on Boot di Windows

Untuk mounting folder yang disharing oleh server di windows secara otomatis, kita bisa memanfaatkan fitur map network drive yang ada diwindows. Berikut langkah-langkah yang perlu dilakukan



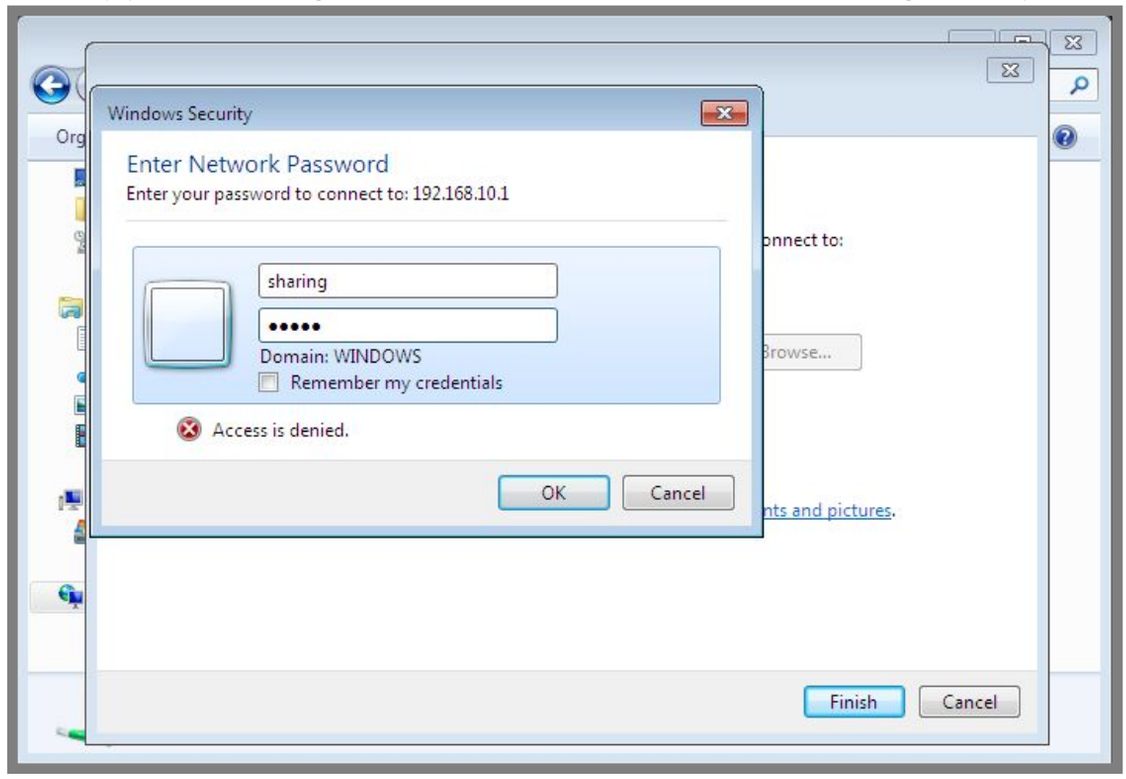
Gambar 11.27 Konfigurasi mounting otomatis di windows client

Selanjutnya kita hanya perlu mengikuti langkah-langkah yang telah ditunjukkan saja



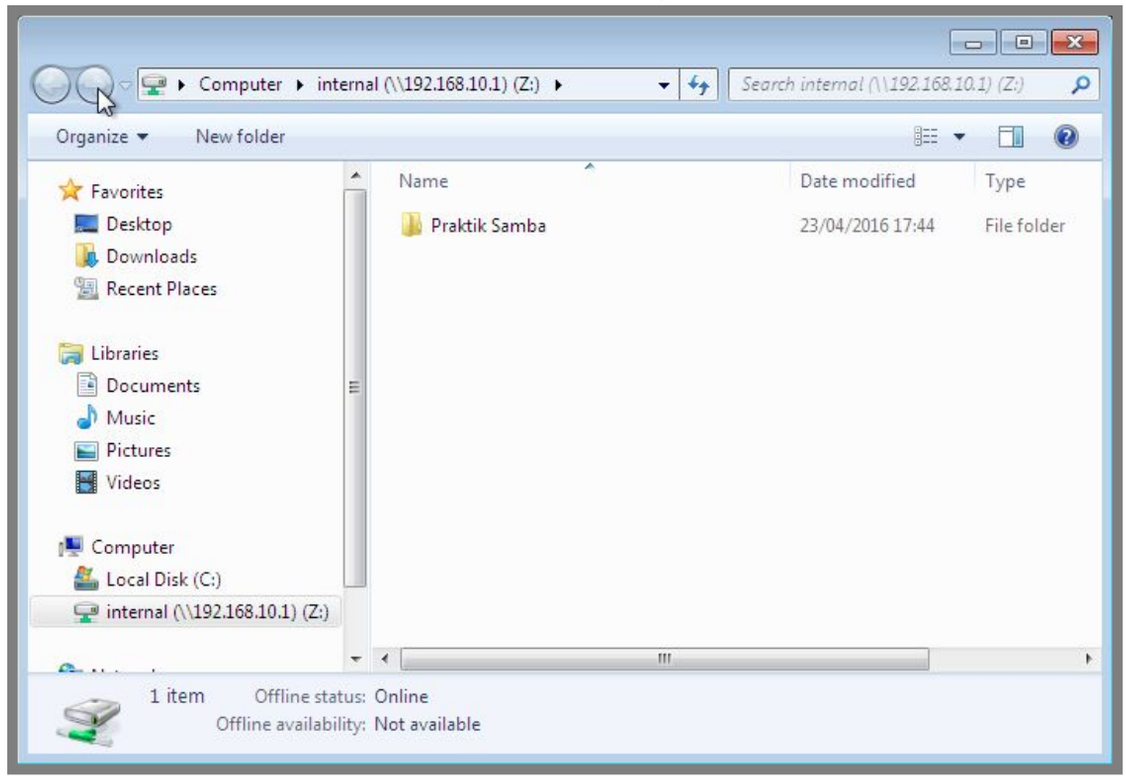
Gambar 11.28 Konfigurasi mounting otomatis di windows client

Masukkan password, hal ini karena folder internal disharing menggunakan metode security password. Langkah ini tidak akan muncul saat kita mounting folder public



Gambar 11.29 Memasukkan username dan password untuk samba

Berikut tampilan saat kita sudah selesai mounting folder internal



Gambar 11.30 Hasil konfigurasi mounting otomatis di client windows

Saat ini begitu client windows dinyalakan, maka folder internal yang disharing oleh server sudah muncul di bagian computer seperti gambar diatas.

Untuk mounting folder public, langkah-langkah yang harus dilakukan sama saja. Tidak akan dijelaskan lagi pada buku ini.

Konfigurasi File Server dengan NFS

Selain menggunakan samba untuk file sharing server seperti yang telah kita bahas pada sub bab sebelumnya, kita bisa juga memanfaatkan Network File System (NFS). Secara keseluruhan, fungsi dari samba dan NFS sama saja, yaitu untuk keperlua file sharing server.

Topologi yang akan kita gunakan pada sub bab ini sama dengan topologi yang telah kita gunakann sebelumnya, yaitu pada gambar 11.1.

Berikut langkah-langkah yang perlu kita lakukan untuk menginstall dan mengkonfigurasi file sharing server menggunakan NFS

```
root@forkits:~# apt-get install nfs-kernel-server nfs-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-common is already the newest version.
The following NEW packages will be installed:
  nfs-kernel-server
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/154 kB of archives.
After this operation, 502 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 11.31 Installasi aplikasi nfs server

Perhatikan perintah yang kita gunakan diatas berfungsi untuk menginstall aplikasi-aplikasi yang diperlukan untuk membuat sebuah file sharing server dengan NFS.

Selanjutnya lakukan beberapa konfigurasi seperti berikut

```
root@forkits:~# mkdir /nfs
root@forkits:~# chmod 777 /nfs/
root@forkits:~# nano /etc/exports
.....
.....
.....
# Example for NFSv4:
# /srv/nfs4          gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes   gss/krb5i(rw,sync,no_subtree_check)
#
/nfs/    192.168.10.0/24(rw,sync,no_subtree_check,no_root_squash)
```

Gambar 11.32 Konfigurasi file server dengan nfs

Berikut penjelasan dari masing-masing perintah dan script yang digunakan pada gambar diatas

Syntak	Deskripsi
mkdir /nfs	Digunakan untuk membuat direktori yang akan kita sharing menggunakan NFS
chmod 777 /nfs/	Digunakan untuk merubah hak akses direktori /nfs menjadi full akses untuk semua orang
nano /etc/exports	Digunakan untuk melakukan konfigurasi NFS
/nfs/	Menunjukkan bahwa direktori yang dishare menggunakan NFS adalah /nfs/
192.168.10.0/24	Menunjukkan ip prefix yang bisa mengakses NFS

rw	Menunjukkan bahwa hak akses untuk direktori yang disharing adalah read and write
sync	Digunakan agar isi direktori pada server maupun pada client selalu sinkron (jika ada perubahan di server, maka diclient akan langsung update, begitu juga sebaliknya)
no_subtree_check	Menunjukkan bahwa client tidak bisa melihat susunan direktori yang disharing
no_root_squash	Menunjukkan bahwa siapa saja yang mengakses direktori yang disharing mempunyai hak akses setara dengan root terhadap direktori tersebut

Selanjutnya restart service NFS di server

```
root@forkits:~# mkdir /nfs/folder1
root@forkits:~# mkdir /nfs/folder2
root@forkits:~# service nfs-kernel-server restart
[ ok ] Stopping NFS kernel daemon: mountd nfsd.
[ ok ] Unexporting directories for NFS kernel daemon....
[ ok ] Exporting directories for NFS kernel daemon....
[ ok ] Starting NFS kernel daemon: nfsd mountd.
root@forkits:~#
```

Gambar 11.33 Restart service nfs

Perhatikan gambar diatas, terlihat bahwa sebelum merestart service NFS, kita membuat beberapa direktori didalam /nfs/ untuk keperluan pengujian.

Selanjutnya berikut perintah yang dapat kita gunakan untuk mount direktori yang disharing oleh NFS server di client ubuntu

```
admin@ubuntu:~$ sudo mkdir /media/nfs
[sudo] password for admin: (tidak terlihat)
admin@ubuntu:~$ sudo mount -t nfs 192.168.10.1:/nfs/ /media/nfs/
admin@ubuntu:~$ ls /media/nfs/
folder1  folder2
admin@ubuntu:~$
```

Gambar 11.34 Mount nfs pada client ubuntu

Perhatikan gambar diatas, terlihat bahwa pertama kita membuat sebuah direktori untuk memount direktori yang disharing oleh server, selanjutnya kita menggunakan perintah mount untuk memount direktori yang dishare oleh server ke direktori yang baru saja kita buat. Perintah terahir menunjukkan bahwa kita berhasil memount direktori yang disharing oleh server, terbukti dengan adanya direktori folder1 dan folder2 yang telah kita buat di server pada direktori /media/nfs/.

Perintah diatas digunakan untuk mount direktori yang dishare oleh server secara sementara. Dalam artian jika komputer client kita restart, maka direktori tersebut tidak akan termount secara otomatis.

Dalam beberapa kasus, kita diharuskan untuk memount direktori yang dishare oleh server tersebut secara otomatis dikomputer client. Berikut langkah-langkah yang bisa kita lakukan

```
admin@ubuntu:~$ sudo nano /etc/fstab
.....
.....
.....
192.168.10.1:/nfs/   /media/nfs      nfs      rw      0   0
admin@ubuntu:~$ sudo reboot
```

Gambar 11.35 Mount nfs secara permanen di client ubuntu

Perhatikan gambar diatas, terlihat bahwa kita menambahkan sebuah baris konfigurasi pada ahir file fstab. Dialnjutkan perintah untuk merestart komputer client.

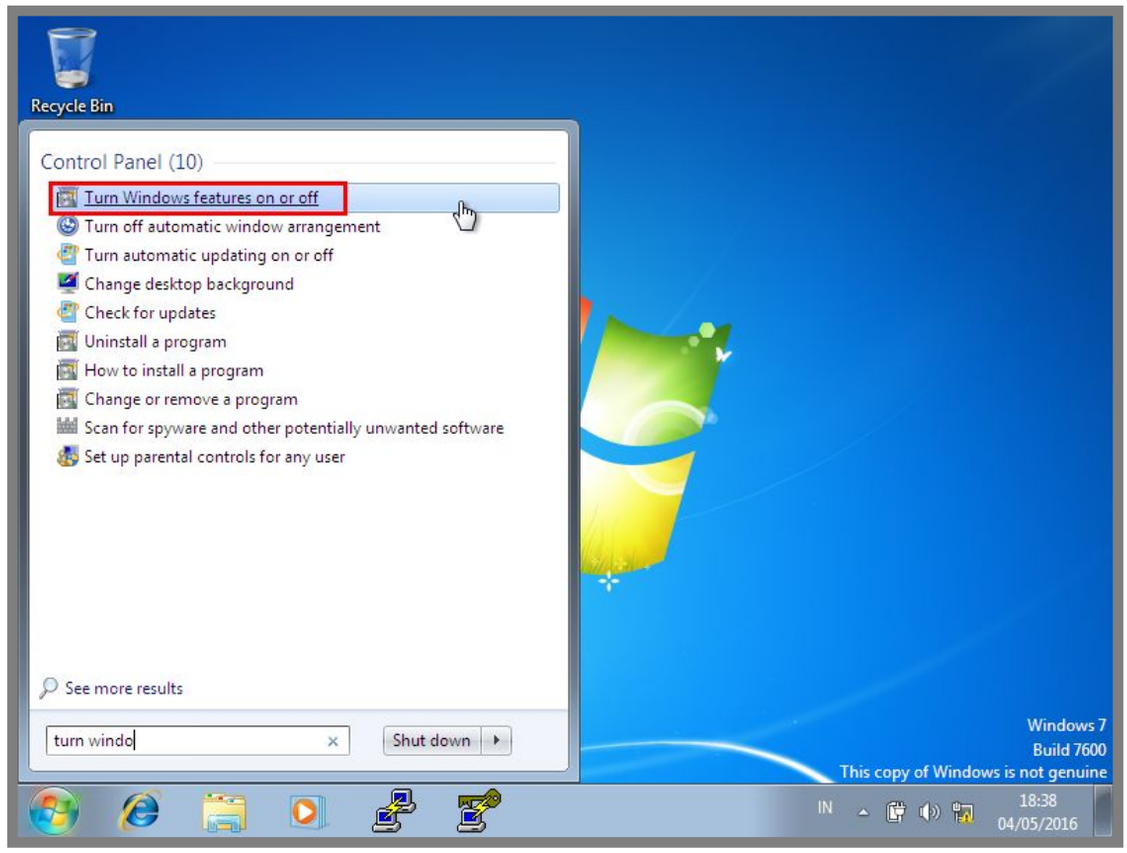
Setelah komputer client selesai restart, pastikan bahwa direktori yang disharing oleh server telah termount secara otomatis di /media/nfs

```
admin@ubuntu:~$ ls /media/nfs/
folder1  folder2
admin@ubuntu:~$
```

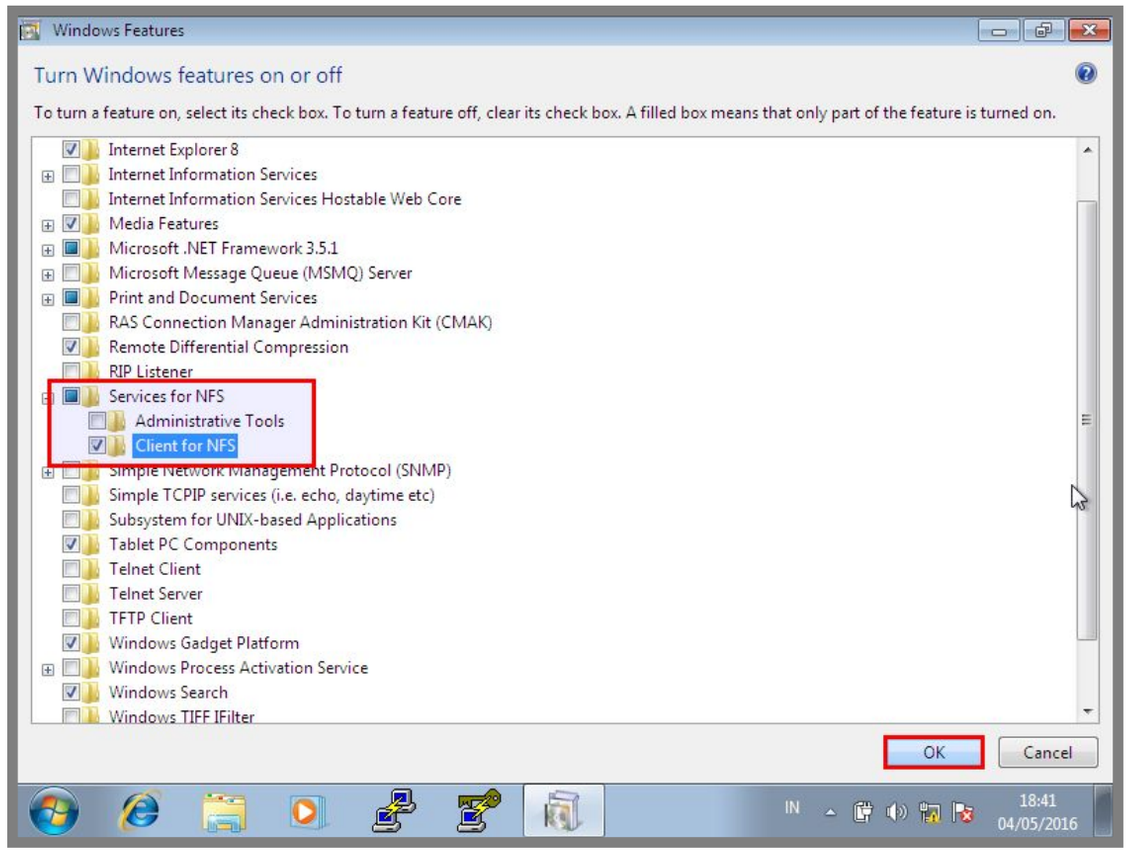
Gambar 11.36 Hasil mounting nfs secara permanen

Perhatikan gambar diatas, terlihat bahwa setelah client direstart, maka direktori yang disharing oleh server otomatis dimount oleh client.

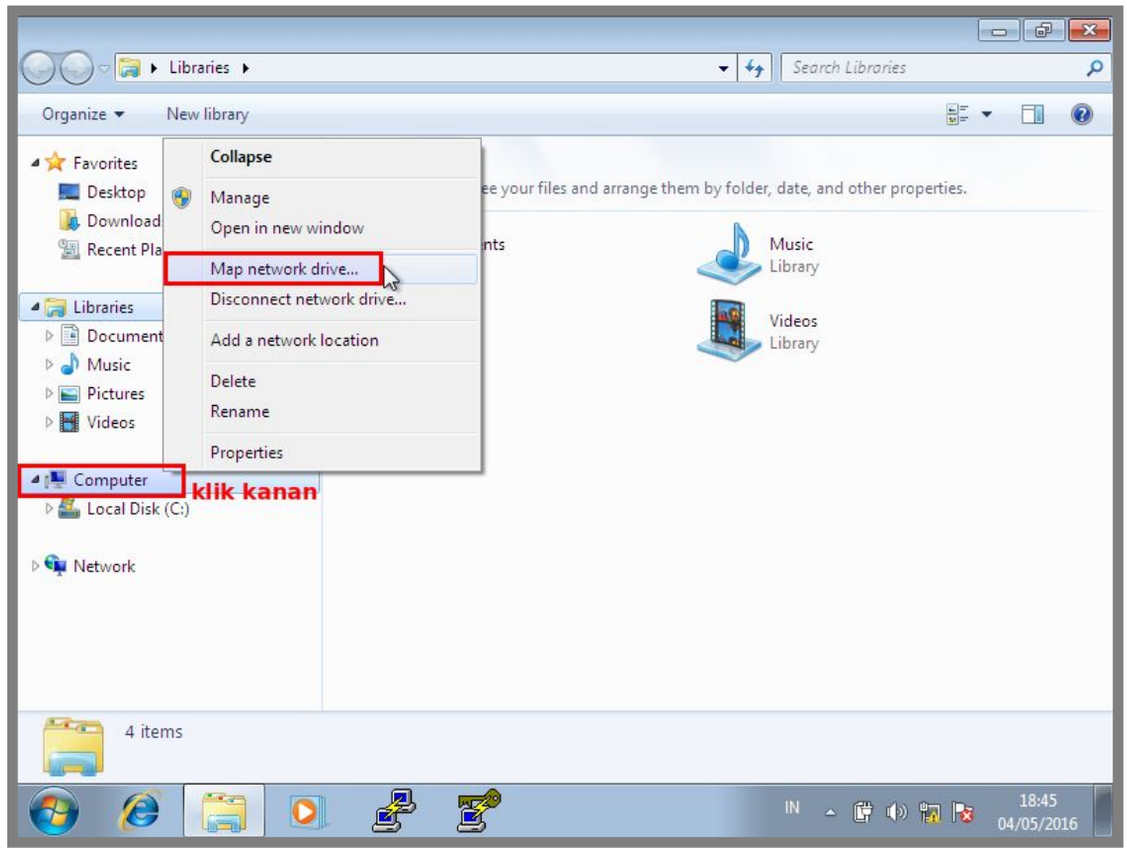
Selanjutnya untuk mounting direktori yang disharing oleh server menggunakan windows, bisa mengikuti langkah-langkah berikut



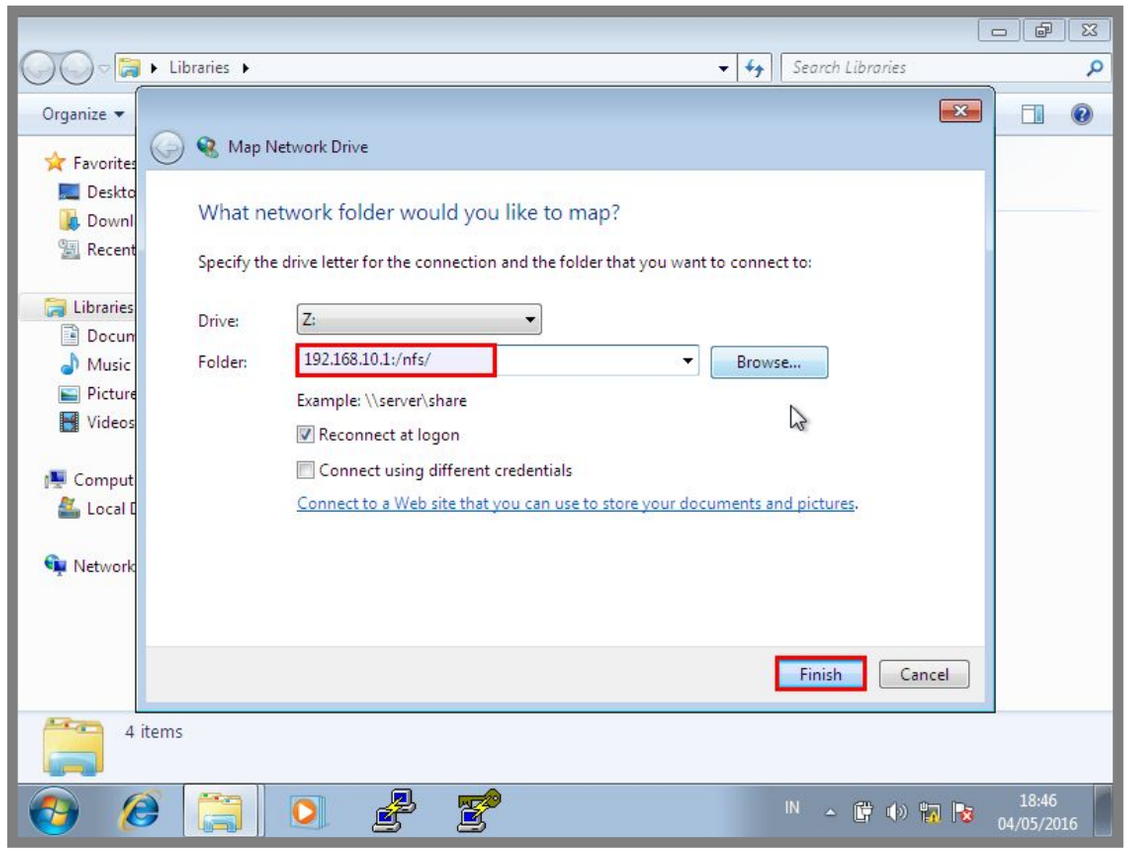
Gambar 11.37 Mengaktifkan fitur nfs client di windows



Gambar 11.38 Mengaktifkan fitur nfs client di windows

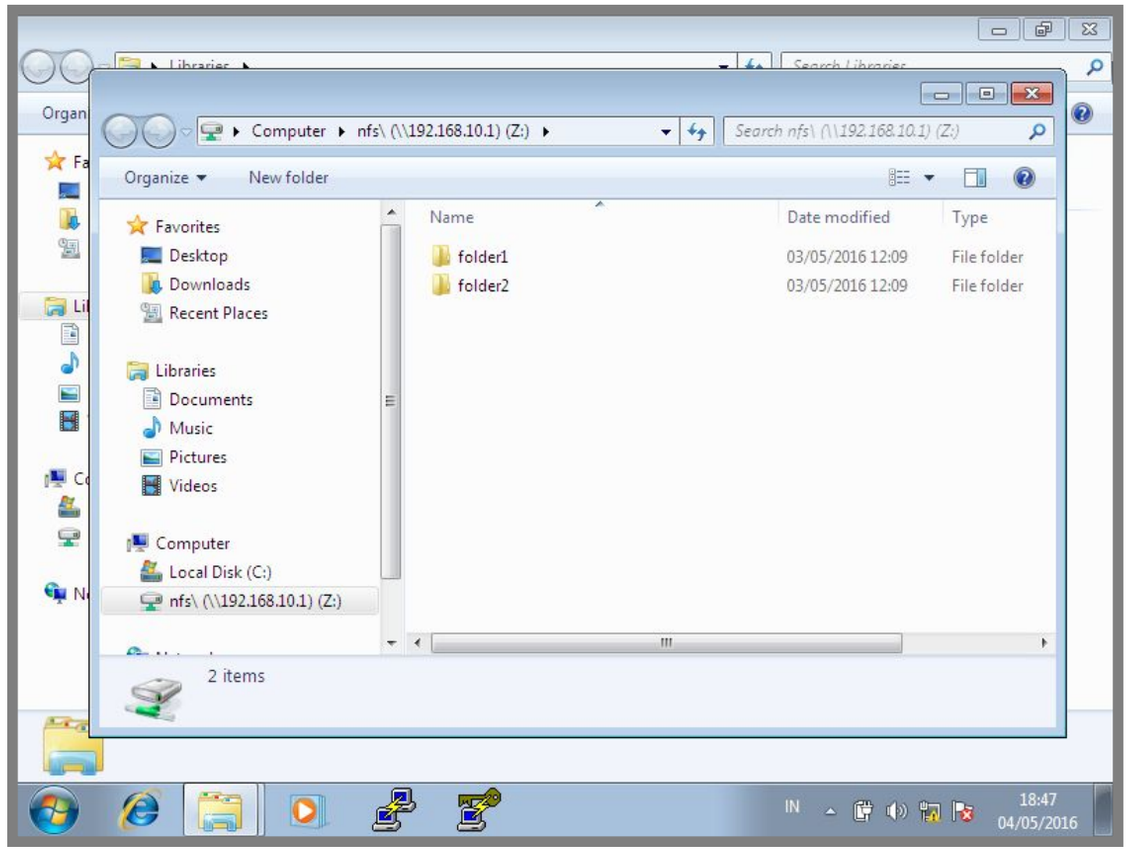


Gambar 11.39 Memount nfs pada client windows



Gambar 11.40 Memount nfs pada client windows

Berikut hasil saat kita telah selesai memount direktori yang dishare oleh server di windows



Gambar 11.41 Hasil mounting nfs pada client windows

---END OF CHAPTER---

Bab 12
Network Time Protocol Server

Network Time Protocol (NTP) adalah sebuah protocol yang bertugas untuk melakukan sinkorniasis waktu antar komputer dalam suatu jaringan, entah itu pada jaringan local ataupun jaringan publik.

Sehingga dapat kita simpulkan bahwa NTP Server adalah sebuah server yang bertugas untuk menyamakan waktu pada seluruh perangkat jaringan yang ada, entah itu komputer, router, switch, access point, printer, cctv, dll. Hal ini sangat diperlukan karena tidak mungkin jika seluruh client yang ada dalam jaringan mempunyai waktu yang sama persis kecuali ada sebuah server yang mengaturnya.

Konfigurasi Local Time

Saat komputer client ingin menyamakan waktu dengan komputer server, tentunya komputer server harus mempunyai waktu yang benar. Bagaimana jadinya jika waktu pada komputer server saja sudah salah? Tentu waktu pada seluruh client akan salah juga.

Oleh sebab itu, hal pertama sebelum membuat sebuah NTP server, kita harus mengkonfigurasi waktu (tanggal & jam) pada komputer server.

Berikut perintah yang bisa kita gunakan untuk melihat waktu pada komputer server

```
root@forkits:~# date
Sel Agu 26 02:57:30 WIB 2014
root@forkits:~#
```

Gambar 12.1 Melihat waktu pada server

Perhatikan gambar diatas, terlihat bahwa waktu di komputer server menunjukkan tanggal 26 Agustus 2014. Padahal saat ini sudah 2016, oleh sebab itu kita akan

mengkonfigurasi waktu pada server. Berikut perintah yang dapat kita gunakan

```
root@forkits:~# date --set 2016-05-14
Sat May 14 00:00:00 WIB 2016
root@forkits:~#
```

Gambar 12.2 Merubah konfigurasi tanggal pada server

Perhatikan gambar diatas, terlihat bahwa kita mengkonfigurasi waktu pada server menjadi tanggal 14 Mei 2016. Selanjutnya kita harus mengkonfigurasi jam pada server, berikut perintah yang dapat kita gunakan

```
root@forkits:~# date --set 09:28:43
Sat May 14 09:28:43 WIB 2016
root@forkits:~#
```

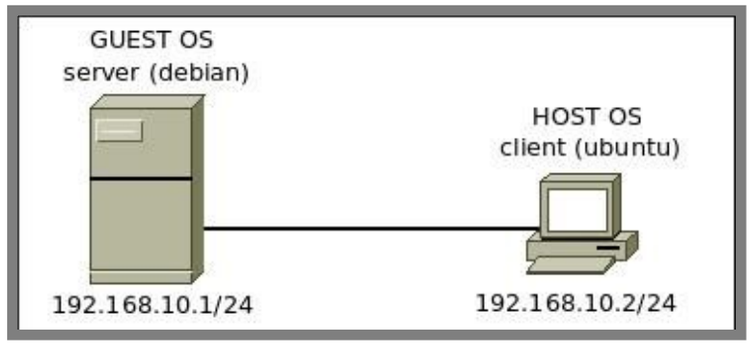
Gambar 12.3 Merubah konfigurasi jam pada server

Perintah diatas digunakan untuk mengkonfigurasi jam pada server menjadi pukul 9 lebih 28 menit 43 detik. Format penulisan jam yang digunakan adalah 24 jam, sehingga jika kita ingin mengkonfigurasi jam 9 malam, maka kita harus menulisnya jam 21.

Konfigurasi NTP Server

Setelah memastikan konfigurasi waktu pada server sudah benar, kita bisa membuat NTP server sehingga nantinya client dapat mencocokkan waktunya dengan waktu pada komputer server.

Berikut topologi jaringan yang akan kita praktikkan



Gambar 12.2 Topologi jaringan untuk praktik ntp server

Diasumsikan bahwa antara komputer server dan komputer client telah dikonfigurasi ip address sesuai topologi diatas dan telah bisa saling berkomunikasi.

Aplikasi yang bisa kita gunakan untuk membuat sebuah NTP server di debian adalah ntp. Berikut perintah yang dapat kita gunakan untuk menginstall aplikasi tersebut

```
root@forkits:~# apt-get install ntp
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libopts25
Suggested packages:
  ntp-doc
The following NEW packages will be installed:
  libopts25 ntp
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/614 kB of archives.
After this operation, 1394 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 12.3 Install aplikasi ntp server

Selanjutnya untuk membuat sebuah server stratum 1, kita perlu melakukan sedikit perubahan konfigurasi pada ntp server. Berikut konfigurasi yang perlu dilakukan

```
root@forkits:~# nano /etc/ntp.conf
.....
.....
.....
# pool.ntp.org maps to about 1000 low-stratum NTP servers.  Your server will
# pick a different set every time it starts up.  Please consider joining the
# pool: <http://www.pool.ntp.org/join.html>
#server 0.debian.pool.ntp.org iburst
#server 1.debian.pool.ntp.org iburst
#server 2.debian.pool.ntp.org iburst
#server 3.debian.pool.ntp.org iburst
server 127.127.1.0
fudge 127.127.1.0 stratum 1
.....
.....
.....
# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
restrict 192.168.10.0 mask 255.255.255.0 nomodify notrap
.....
.....
.....
```

Gambar 12.4 Konfigurasi ntp server

Perhatikan gambar diatas, terlihat bahwa kita melakukan perubahan pada teks warna hijau. Selanjutnya restart service ntp dengan perintah berikut

```
root@forkits:~# service ntp restart
[ ok ] Stopping NTP server: ntpd.
[ ok ] Starting NTP server: ntpd.
root@forkits:~#
```

Gambar 12.5 Restart service ntp server

Selanjutnya untuk melakukan pengujian di client ubuntu, kita membutuhkan sebuah aplikasi yang bernama *ntpdate*. Berikut perintah yang dapat kita gunakan untuk menginstall aplikasi tersebut

```
admin@ubuntu:~$ sudo apt-get install ntpdate
Password: (tidak terlihat)
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  lockfile-progs
The following NEW packages will be installed:
  lockfile-progs ntpdate
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/91.0 kB of archives.
After this operation, 291 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 12.6 Installasi aplikasi ntp client di client ubuntu

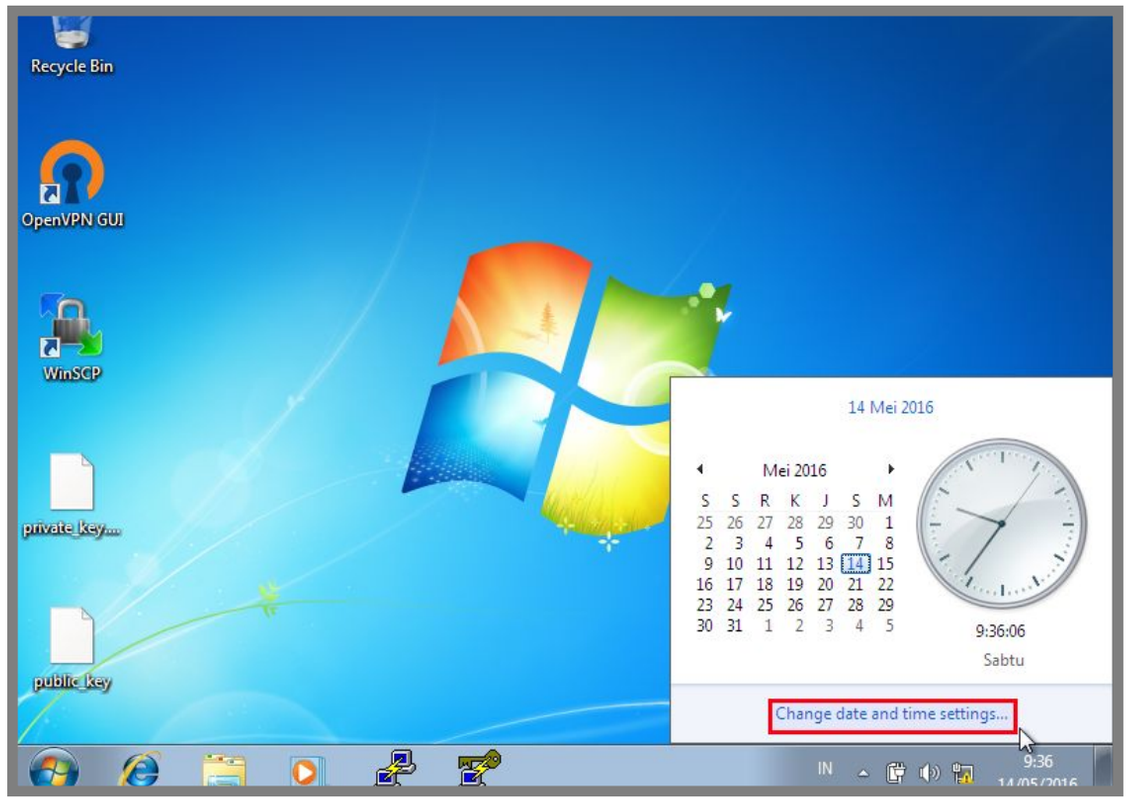
Berikut perintah yang dapat kita gunakan untuk melakukan sinkronisasi waktu di client ubuntu.

```
admin@ubuntu:~$ sudo ntpdate 192.168.10.1
14 May 09:27:39 ntpdate[7303]: step time server 192.168.10.1 offset
-1.520252 sec
admin@ubuntu:~$
```

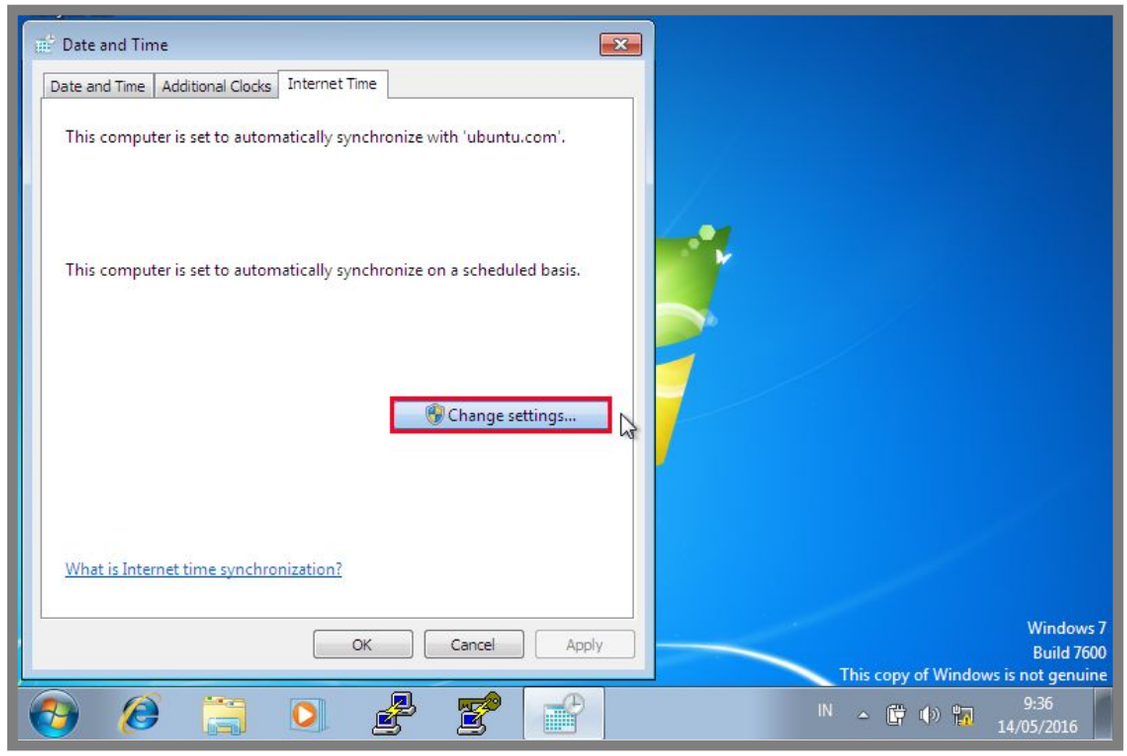
Gambar 12.7 Menyesuaikan waktu pada client ubuntu dengan waktu pada server

Perhatikan bahwa saat ini waktu pada komputer client sudah tersinkron dengan waktu pada server. Dalam artian waktunya sama persis.

Selanjutnya jika kita menggunakan windows sebagai client, berikut langkah-langkah yang perlu dilakukan untuk sinkronisasi waktu dengan NTP server

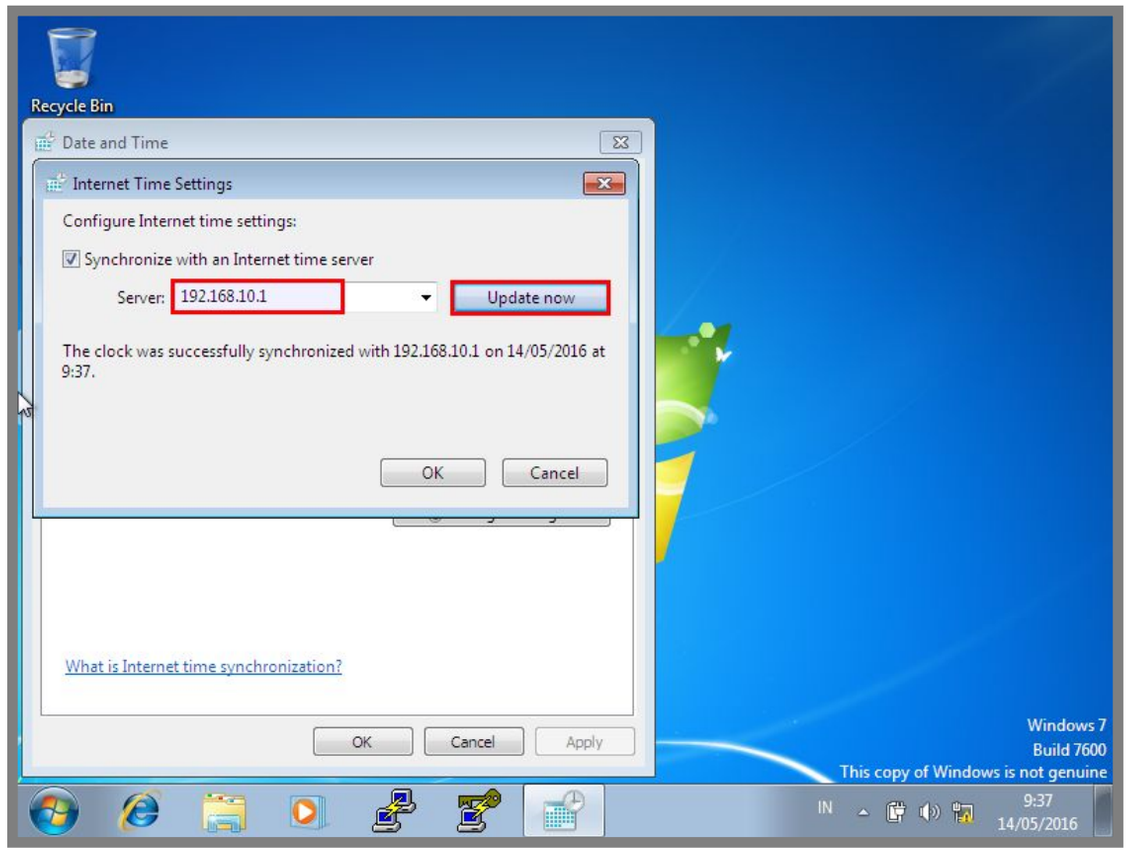


Gambar 12.8 Konfigurasi ntp client di windows



Gambar 12.9 Konfigurasi ntp client di windows

Selanjutnya masukkan ip address dari NTP server kemudian lakukan update. Perlu diketahui bahwa biasanya pada percobaan pertama, update sering gagal. Namun kita tidak perlu hawatir, kita hanya tinggal mengulangi update hingga berhasil.



Gambar 12.10 Konfigurasi ntp client di windows

Perhatikan gambar diatas, terlihat bahwa ada sebuah peringatan bahwa kita telah berhasil melakukan sinkronisasi waktu dengan NTP server.

---END OF CHAPTER---

Bab 13
Monitoring Server

Seorang administrator jaringan tidak hanya dituntut untuk bisa mengkonfigurasi jaringan dengan baik dan efisien. Namun juga dituntut untuk bisa melakukan monitoring dan maintenance jaringan dengan baik.

Monitoring jaringan merupakan sebuah kegiatan yang sangat penting untuk dilakukan. Dengan melakukan monitoring jaringan, kita bisa mengetahui kondisi jaringan, apakah jaringan dalam keadaan up ataupun down.

Tujuan utama dari pekerjaan monitoring jaringan adalah, seorang administrator jaringan bisa mengetahui jika suatu saat terjadi kesalahan pada jaringan (down). Sehingga administrator jaringan bisa menangani masalah tersebut sedini mungkin.

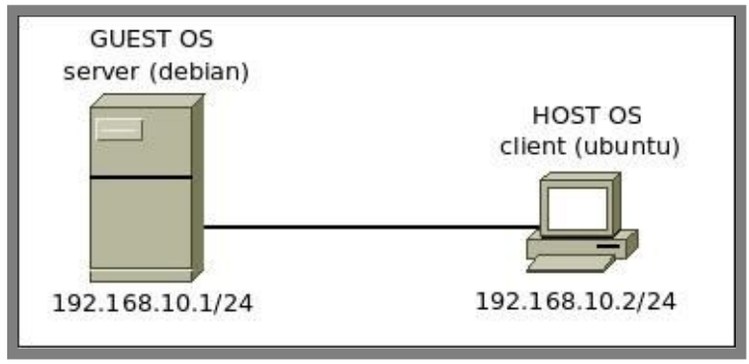
Untuk melakukan monitoring jaringan, pada dasarnya kita bisa memanfaakan dua hal, yaitu utilitas ping dan SNMP. Jika kita memanfaatkan utilitas ping, maka kita hanya akan mengetahui kondisi dari sebuah perangkat jaringan apakah dalam kondisi up atau down.

Namun jika kita menggunakan protocol SNMP, selain dapat mengetahui kondisi perangkat jaringan apakah up atau down, kita juga bisa mengetahui beberapa informasi lain yang penting. Seperti utilitas dari cpu, memory, merek perangkat, tipe perangkat, trafik yang terbaca pada setiap port, versi sistem operasi yang digunakan, dll.

Selanjutnya, untuk melakukan monitoring jaringan tentunya kita akan lebih mudah jika dihadapkan dengan antarmuka grafis yang menarik (GUI). Oleh sebab itu, kita akan memanfaatkan aplikasi cacti untuk melakukan monitoring jaringan.

Cacti adalah salah satu aplikasi yang digunakan untuk monitoring jaringan yang memanfaatkan protocol SNMP. Aplikasi ini berbasis web application, sehingga untuk menjalankannya, kita hanya membutuhkan sebuah web browser.

Untuk praktik pada bab ini, kita akan menggunakan topologi jaringan seperti yang ditunjukkan pada gambar 13.1. Diasumsikan antara komputer server dan client sudah dikonfigurasi ip address sesuai topologi dan telah bisa saling berkomunikasi.



Gambar 13.1 Topologi jaringan untuk praktik monitoring server

Konfigurasi SNMP

Telah disebutkan sebelumnya bahwa SNMP merupakan sebuah protocol yang bertugas untuk melakukan monitoring jaringan. Oleh sebab itu, kita wajib menginstall dan mengkonfigurasi protocol ini jika ingin memonitoring jaringan.

Berikut perintah yang dapat kita gunakan untuk menginstall SNMP di Debian

```
root@forkits:~# apt-get install snmp snmpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libsensors4 libsnmp-base libsnmp15
Suggested packages:
  lm-sensors snmp-mibs-downloader
The following NEW packages will be installed:
  libsensors4 libsnmp-base libsnmp15 snmp snmpd
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/5346 kB of archives.
After this operation, 8278 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 13.2 Installasi aplikasi snmp

Selanjutnya kita harus melakukan konfigurasi pada SNMP. Ikuti langkah-langkah yang ditunjukkan pada gambar 13.3

```
root@forkits:~# nano /etc/snmp/snmpd.conf
.....
.....
.....
#   Listen for connections from the local system only
#agentAddress  udp:127.0.0.1:161
#   Listen for connections on all interfaces (both IPv4 *and* IPv6)
agentAddress udp:161,udp6:[::1]:161
.....
.....
.....
rocommunity public 127.0.0.1
rocommunity public 0.0.0.0/0
.....
.....
.....
sysLocation     Blitar
sysContact      admin@forkits.com
.....
.....
.....
```

Gambar 13.3 Konfigurasi snmp

Perhatikan gambar diatas, terlihat bahwa kita melakukan perubahan pada teks berwarna hijau. Selanjutnya restart service snmp

```
root@forkits:~# service snmpd restart
Restarting network management services: snmpd.
root@forkits:~#
```

Gambar 13.4 Restart service snmp

Selanjutnya untuk melakukan pengujian, kita bisa memanfaatkan perintah *snmpwalk*. Perhatikan hasil pengujian yang ditunjukkan gambar 13.5

```
root@forkits:~# snmpwalk -c public -v1 127.0.0.1
iso.3.6.1.2.1.1.1.0 = STRING: "Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3
i686"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (3855) 0:00:38.55
iso.3.6.1.2.1.1.4.0 = STRING: "admin@forkits.com"
iso.3.6.1.2.1.1.5.0 = STRING: "forkits"
iso.3.6.1.2.1.1.6.0 = STRING: "Blitar"
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (2) 0:00:00.02
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The MIB for Message Processing and
Dispatching."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The management information definitions for
the SNMP User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "The MIB module for managing TCP
implementations"
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing IP and ICMP
implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP
implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "View-based Access Control Model for
SNMP."
```

Gambar 13.5 Pengujian snmp

Intinya, jika saat kita melakukan pengujian menggunakan snmpwalk dan telah muncul keluaran yang sangat benyak seperti diatas, itu tandanya kita sudah selesai dan berhasil melakukan konfigurasi SNMP.

Selanjutnya kita tinggal melakukan installasi dan konfigurasi cacti untuk melakukan monitoring jaringan menggunakan antarmuka yang menarik dan interactive (GUI).

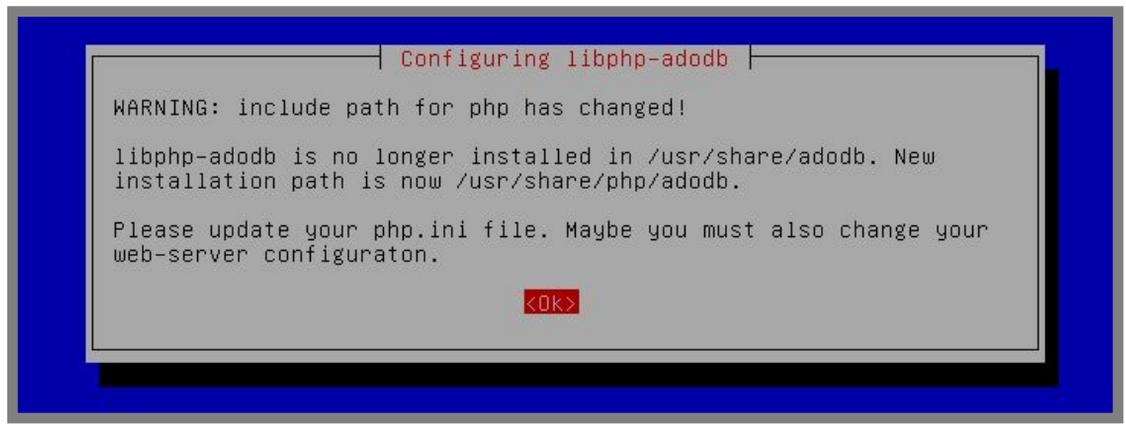
Installasii Cacti

Untuk melakukan installasi cacti, sebelumnya server kita sudah harus terinstall web server (apache) dan database server (mysql-server). Untuk melakukan installasi web server dan database server, silahkan merujuk ke bab sebelumnya.

Diasumsikan bahwa server telah terinstall web server dan database server dengan baik, maka kita bisa langsung install cacti. Berikut perintah yang dapat kita gunakan

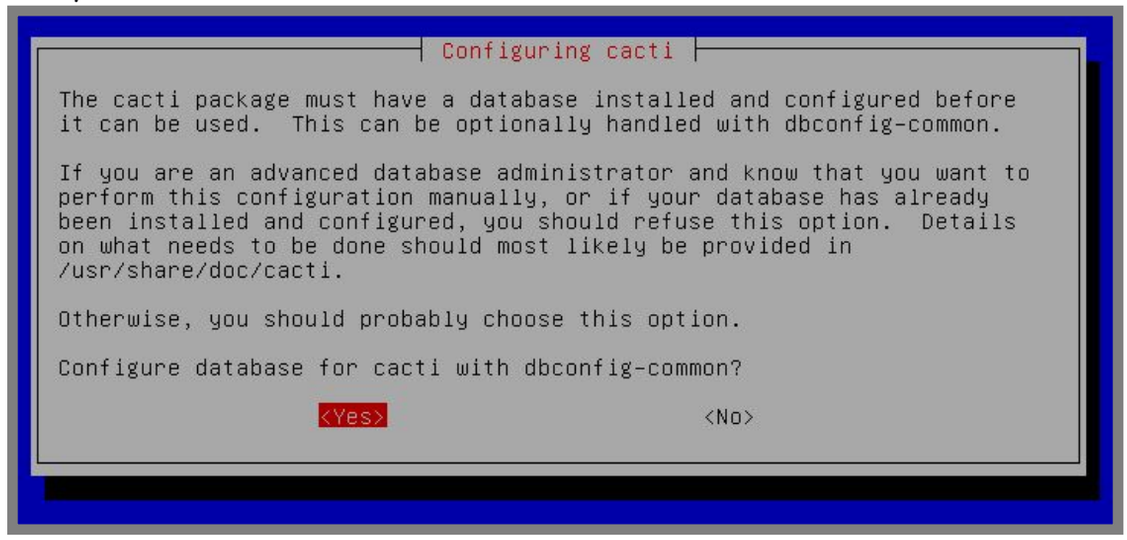
```
root@forkits:~# apt-get install cacti
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  fontconfig libcairo2 libdatrie1 libdbi1 libffi5 libglib2.0-0 libglib2.0-data
libjs-jquery-cookie libpango1.0-0 libphp-adodb
  libpixman-1-0 librrd4 libthai-data libthai0 libxcb-render0 libxcb-shm0
libxft2 libxrender1 php5-snmp rrdtool shared-mime-info
  ttf-dejavu ttf-dejavu-extra
Suggested packages:
  moreutils ttf-baekmuk ttf-arphic-gbsn00lp ttf-arphic-bsmi00lp
ttf-arphic-gkai00mp ttf-arphic-bkai00mp php5-adodb librrds-perl
The following NEW packages will be installed:
  cacti fontconfig libcairo2 libdatrie1 libdbi1 libffi5 libglib2.0-0
libglib2.0-data libjs-jquery-cookie libpango1.0-0 libphp-adodb
  libpixman-1-0 librrd4 libthai-data libthai0 libxcb-render0 libxcb-shm0
0 upgraded, 24 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/12.0 MB of archives.
After this operation, 36.9 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 13.6 Installasi cacti pada server



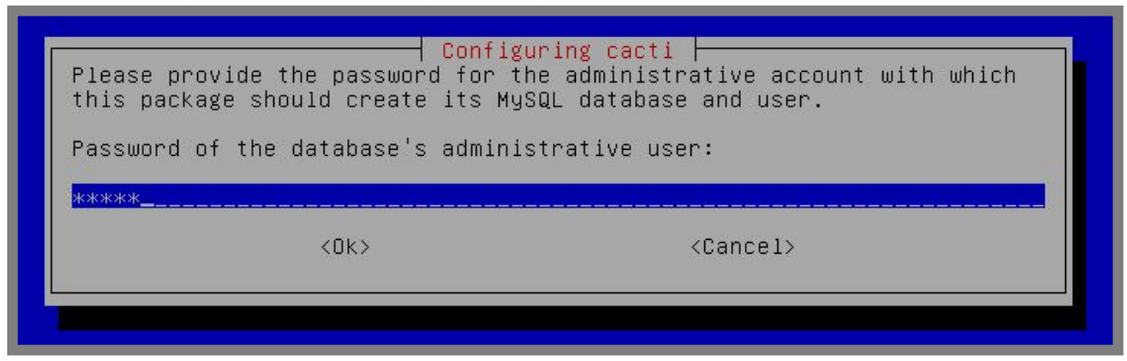
Gambar 13.7 Proses installasi cacti

Ada pertanyaan apakah kita ingin membuat database yang dibutuhkan oleh cacti? Pilih yes



Gambar 13.8 Proses installasi cacti

Untuk membuat database yang dibutuhkan cact, kita diharuskan untuk memasukkan password untuk login ke mysql

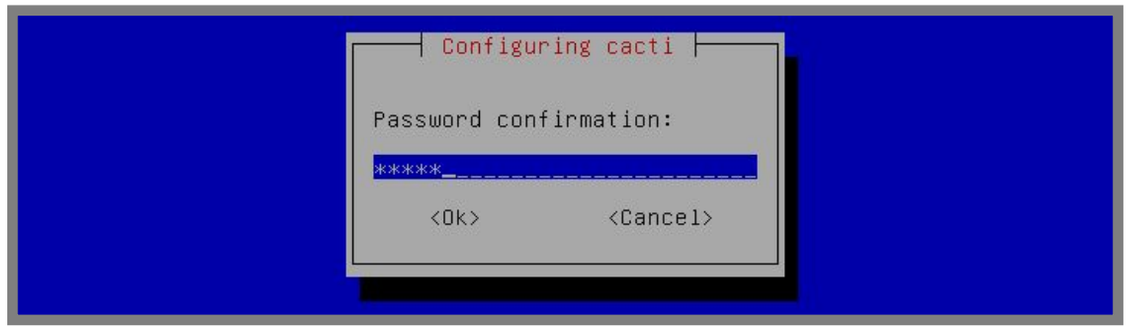


Gambar 13.9 Proses installasi cacti

Selanjutnya kita diminta untuk memasukkan password yang nantinya akan digunakan untuk database cacti

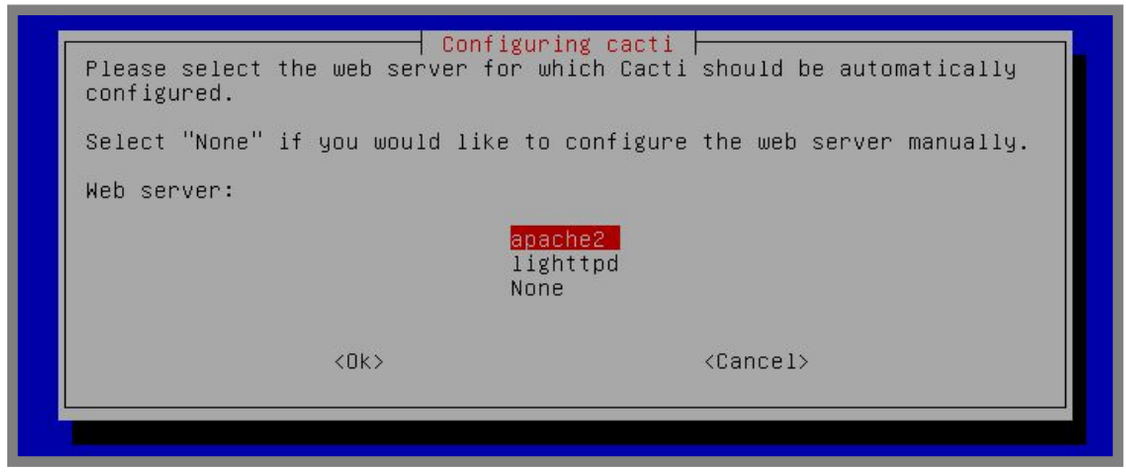
Gambar 13.10 Proses installasi cacti

Masukkan password untuk cacti sekali lagi



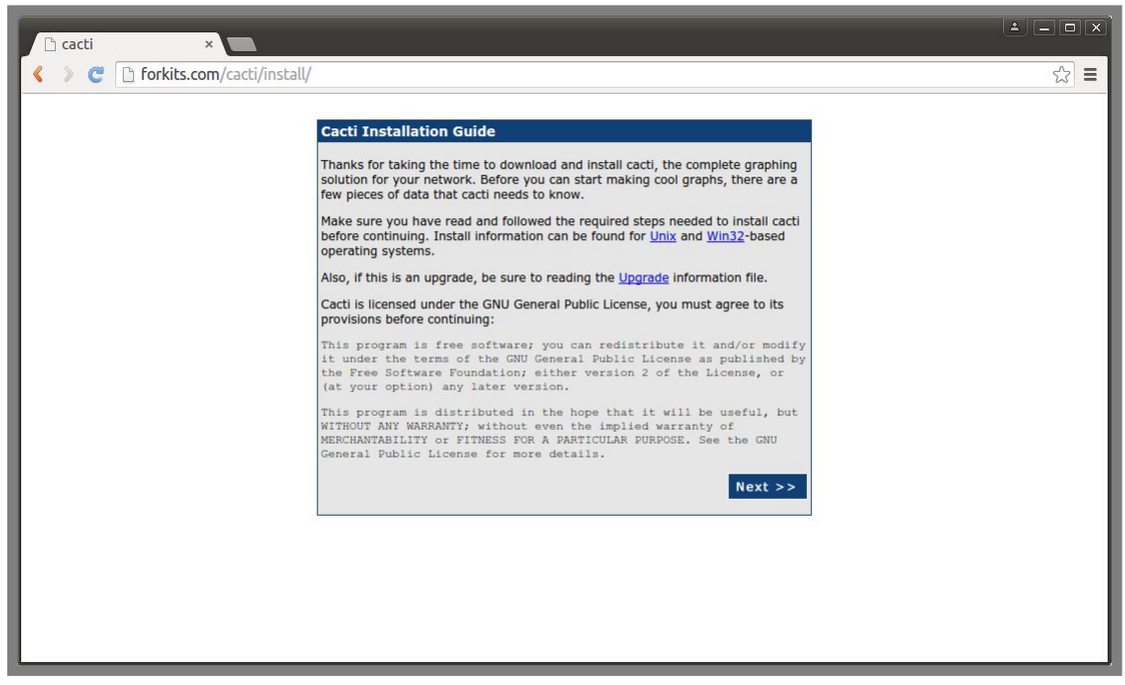
Gambar 13.11 Proses installasi cacti

Pilih web server yang digunakan pada server

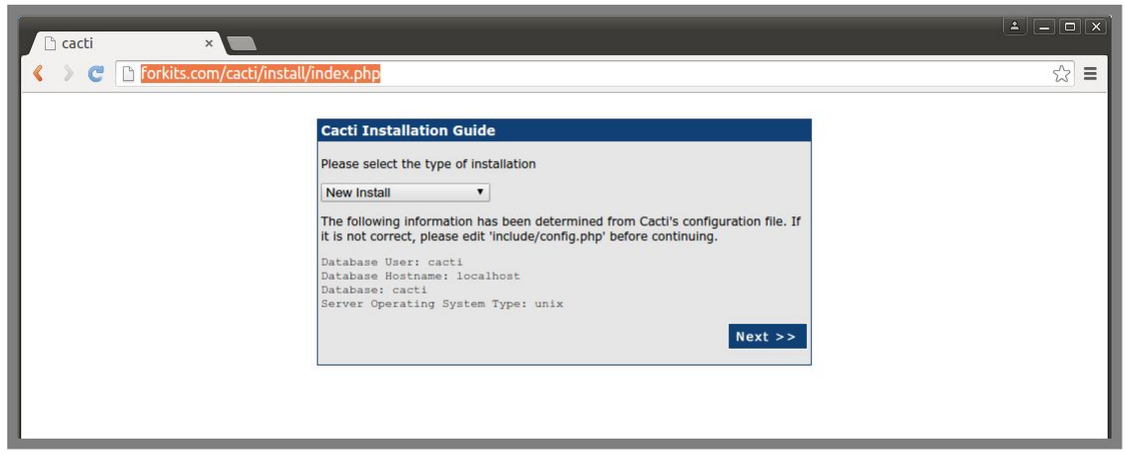


Gambar 13.12 Proses installasi cacti

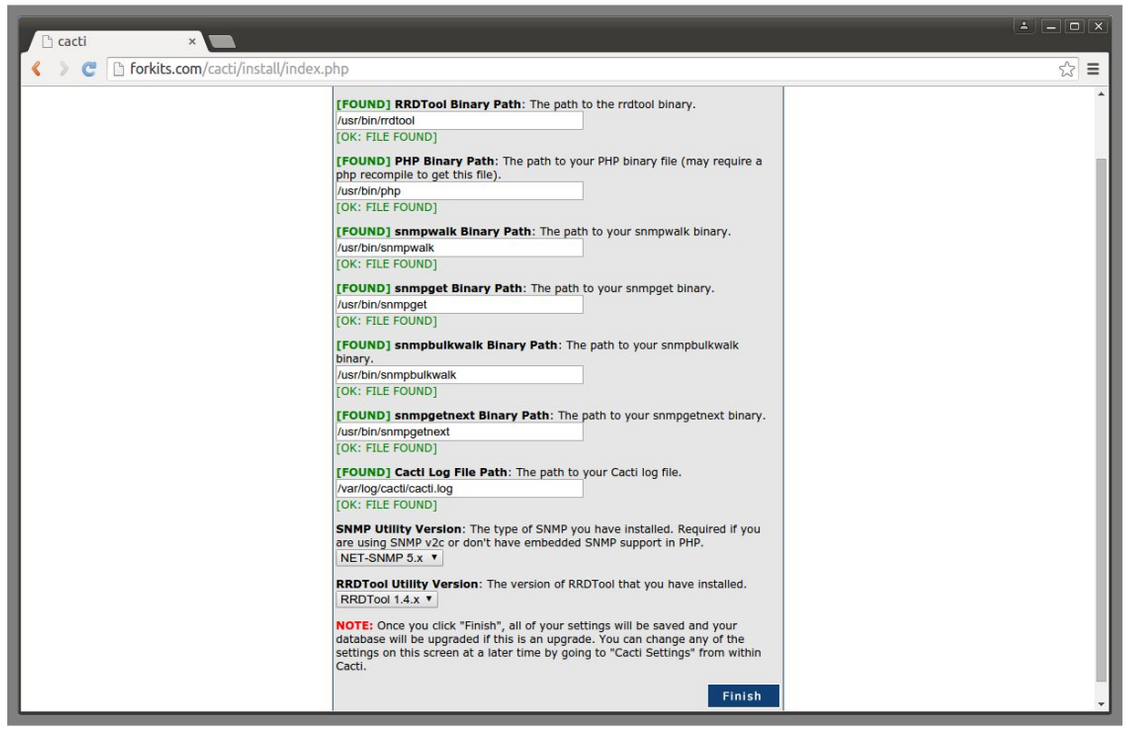
Sampai saat ini kita telah selesai install paket-paket yang dibutuhkan oleh cacti. Selanjutnya kita harus melanjutkan proses installasi cacti menggunakan web browser, kita bisa melakukan proses installasi ini dari client menggunakan url *domain_server/cacti*



Gambar 13.13 Proses installasi cacti

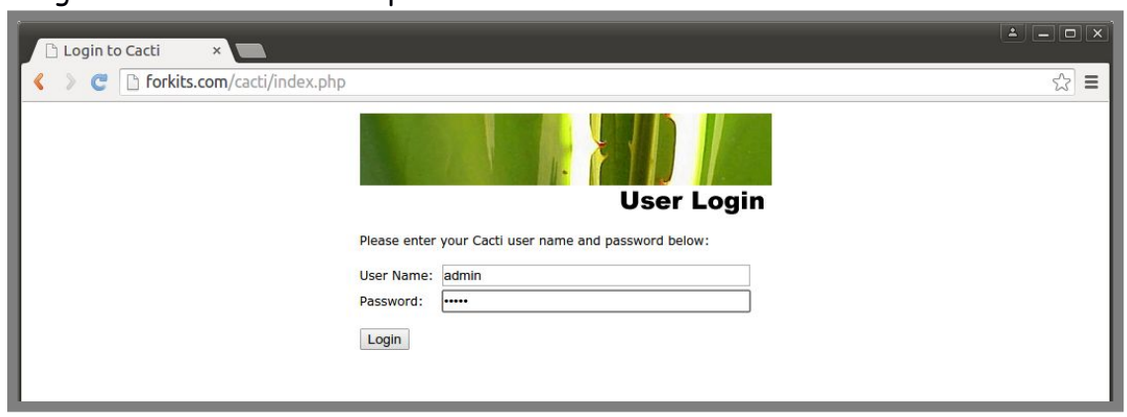


Gambar 13.14 Proses installasi cacti



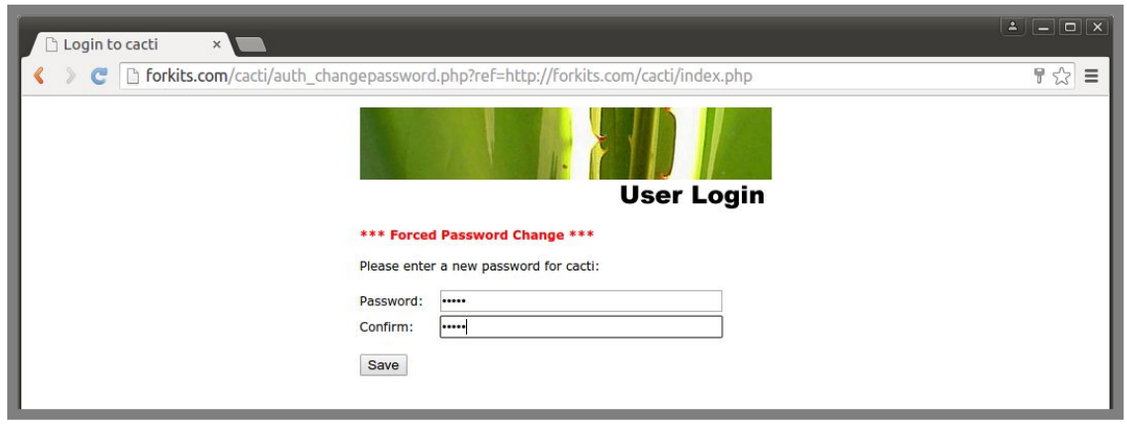
Gambar 13.15 Proses installasi cacti

Sampai saat ini kita sudah selesai install cacti. Selanjutnya kita bisa login ke cacti dengan username admin dan password admin



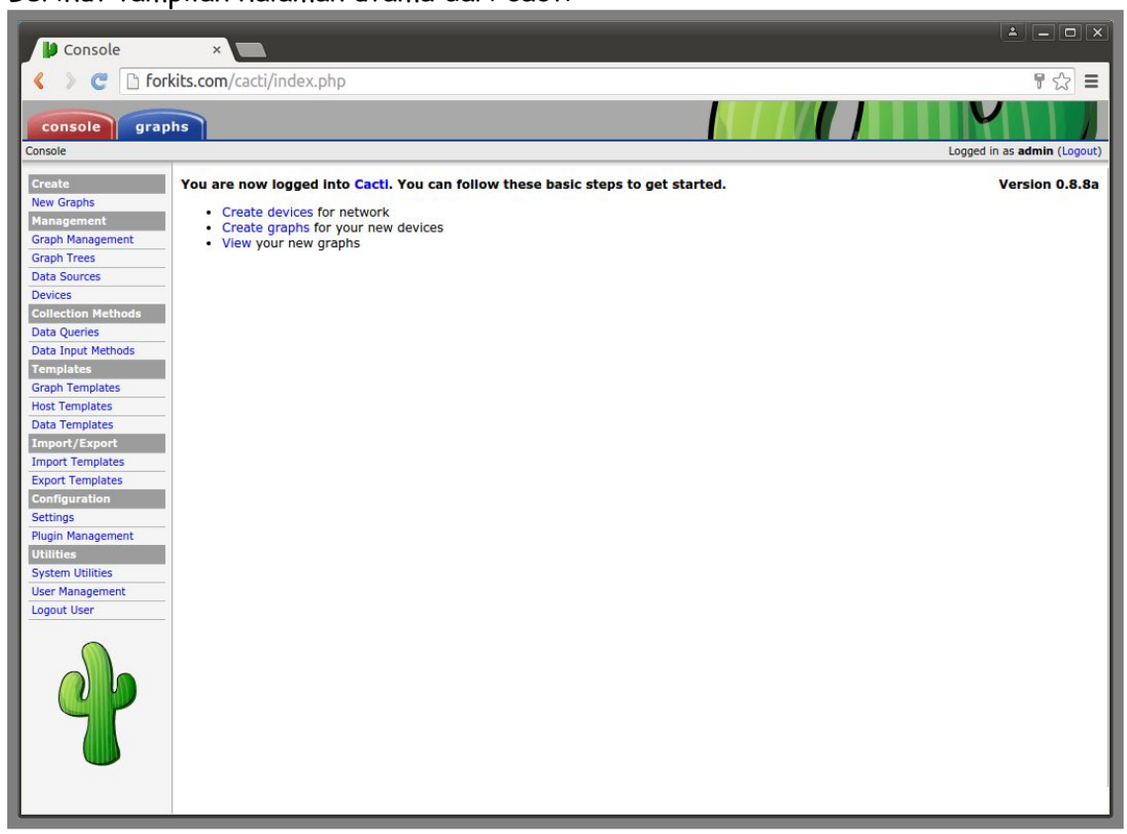
Gambar 13.16 Halaman login cacti

Setelah login, maka selanjutnya kita diharuskan untuk merubah password default admin menjadi sesuai dengan keinginan kita untuk menjaga keamanan cacti



Gambar 13.17 Mengganti password admin cacti

Berikut tampilan halaman utama dari cacti

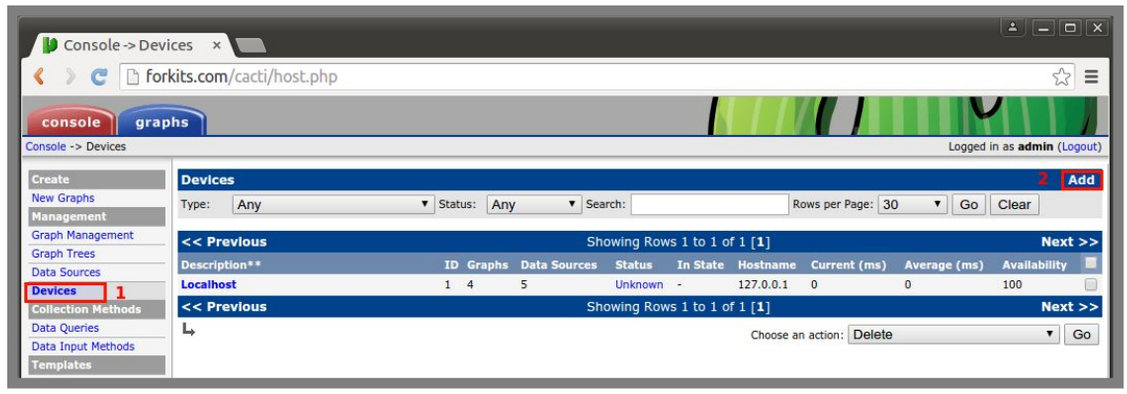


Gambar 13.18 Halaman utama cacti

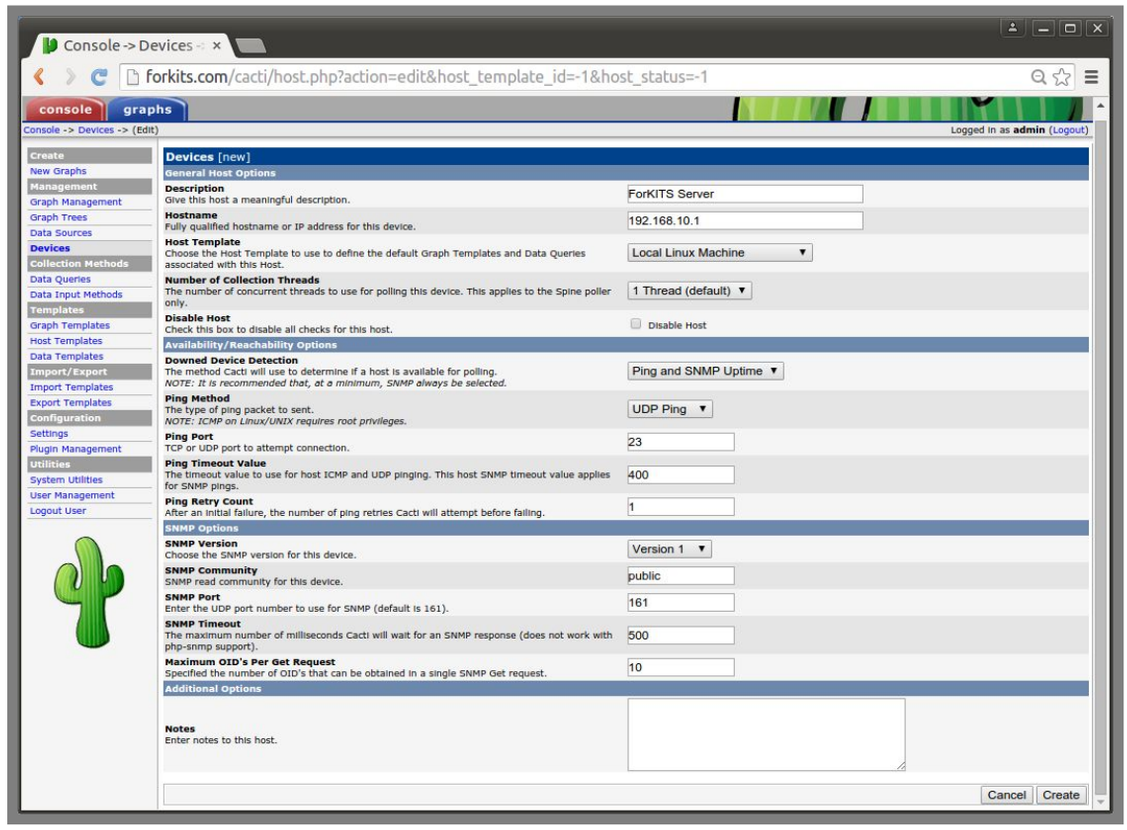
Administrasi Cacti

Untuk memonitoring jaringan, kita tidak cukup jika hanya menginstall SNMP dan cacti saja. Kita harus melakukan administrasi pada cacti, yaitu menambahkan device apa saja yang ingin dimonitoring, kemudian membuat graph untuk device tersebut.

Untuk lebih mudahnya, kita akan langsung praktik saja. Pada bab ini, kita akan mangadministrasi cacti agar bisa memonitoring komputer server. Berikut langkah-langkah yang perlu dilakukan (sebelum klik add, hapus dulu device localhost yang sudah ada dengan cara centang pada localhost, kemudian chose action = delete, kemudian klik go).

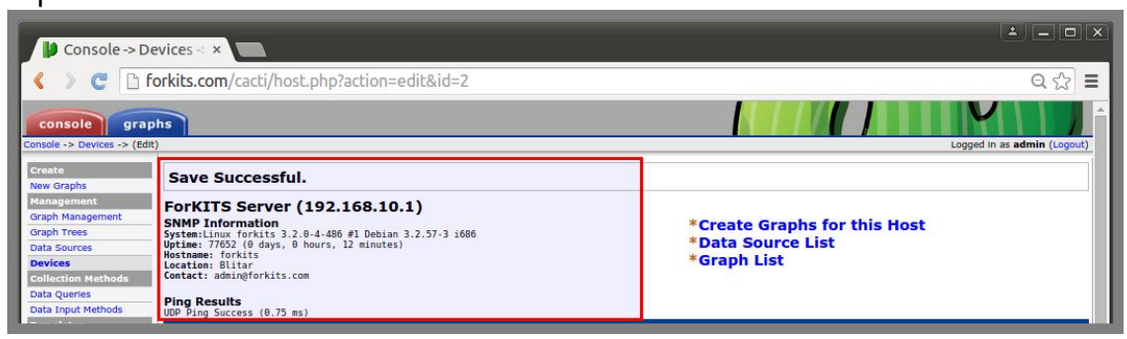


Gambar 13.19 Menghapus dan menambahkan device pada cacti



Gambar 13.20 Konfigurasi device baru pada cacti

Perhatikan gambar diatas, kita memasukkan data-data tentang server, seperti deskripsi, ip address, dll. Kemudian klik create.
Jika kita berhasil menambahkan device di cacti, maka akan muncul tampilan seperti berikut



Gambar 13.21 Hasil penambahan device baru pada cacti

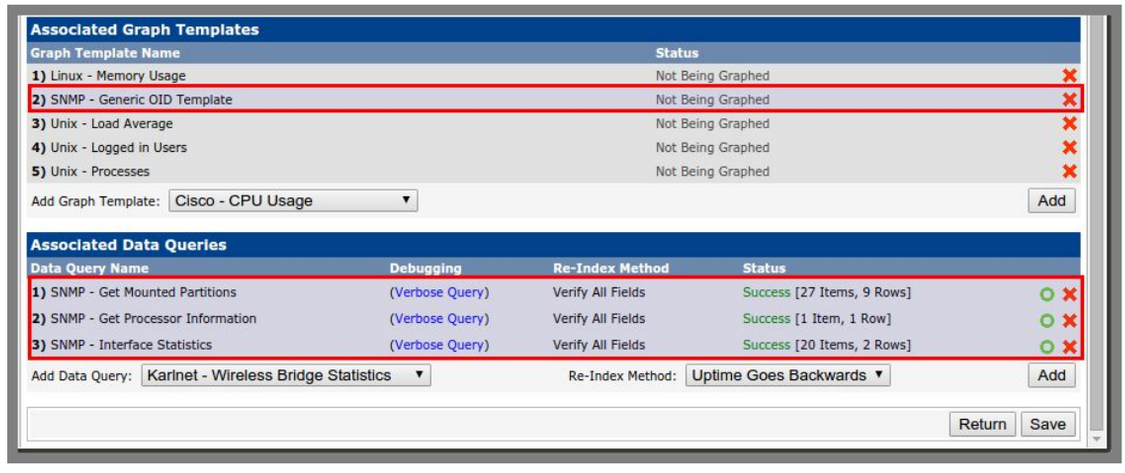
Selanjutnya scroll ke paling bawah. Pada bagian *Associated Data Queries*, hapus *Unix Get Mounted partition* dengan klik tombol silan warna merah (x).

Kemudian *Add Data Query : SNMP - Get Mounted Partitions, Re-Index Method : Verify All Fields* , kemudian klik *Add.*

Ulangi langkah diatas untuk menambahkan *SNMP - Get Processor Information* dan *SNMP - Interface Statistic.*

Selanjutnya pada bagian *Associated Graph Template* tambahkan *SNMP - Generic OID Template.*

Berikut tampilan ahir setelah melakukan langkah-langkah diatas



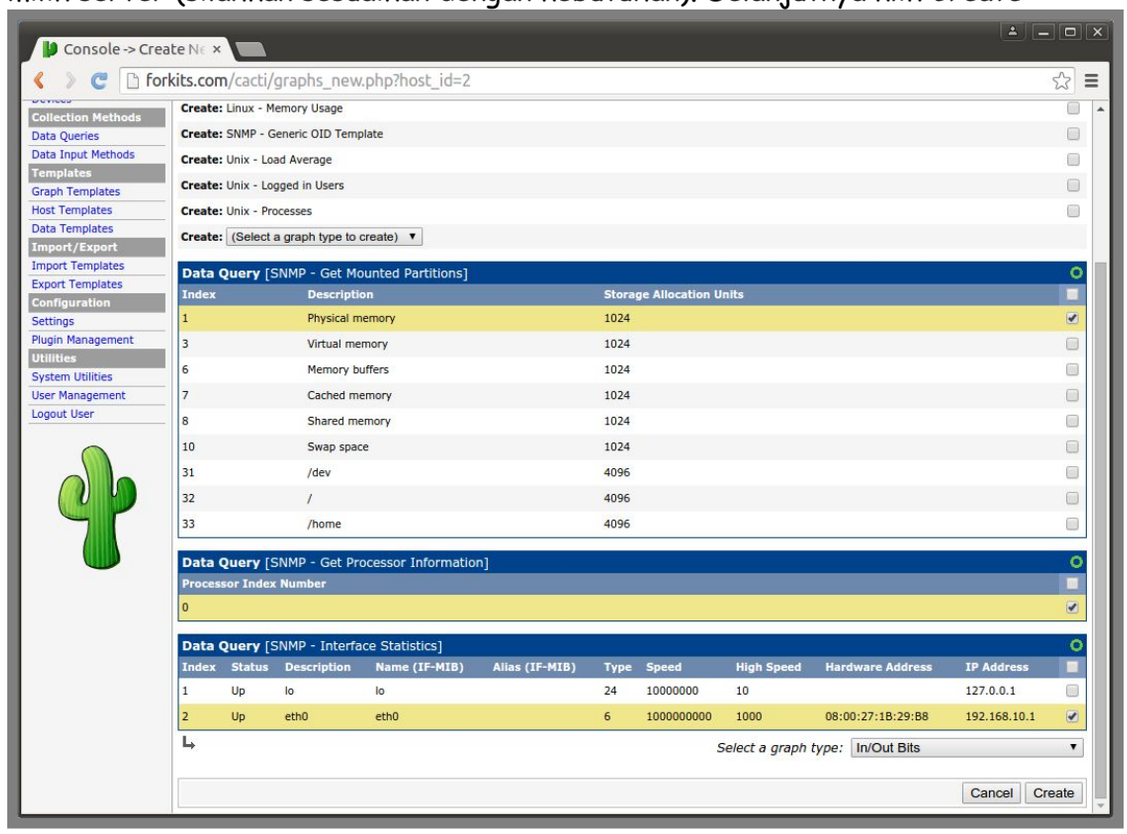
Gambar 13.22 Konfigurasi device baru pada cacti

Jangan lupa untuk save perubahan-perubahan yang telah kita lakukan diatas dengan cara klik tombol *Save.*

Selanjutnya agar kita bisa memonitoring server dengan milhat grafik-grafik, kita harus membuat sebuah graph untuk server tersebut pada cacti. Klik *Create Graphs for this Host*

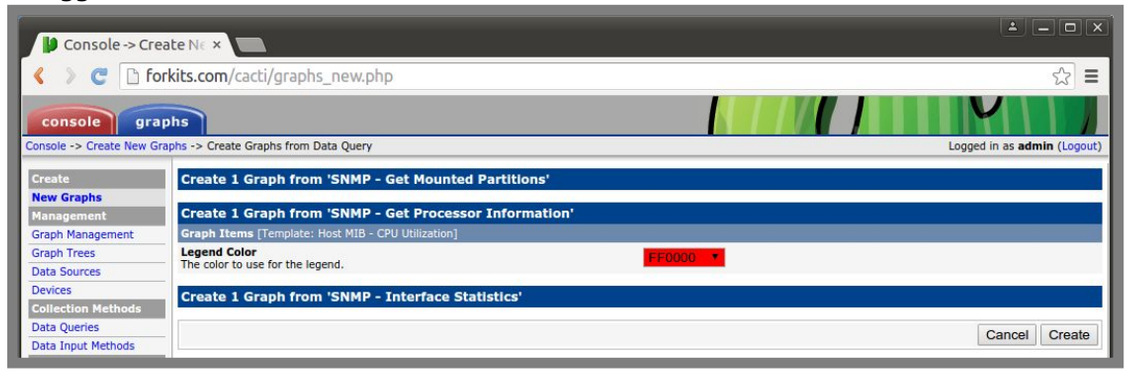
Gambar 13.23 Membuat grafik untuk device baru

Selanjutnya tentukan apa saja yang ingin kita monitoring dari server. Pada tahap ini, saya hanya menginginkan memonitoring RAM, Processor, dan interface eth0 milik server (silahkan sesuaikan dengan kebutuhan). Selanjutnya klik *Create*



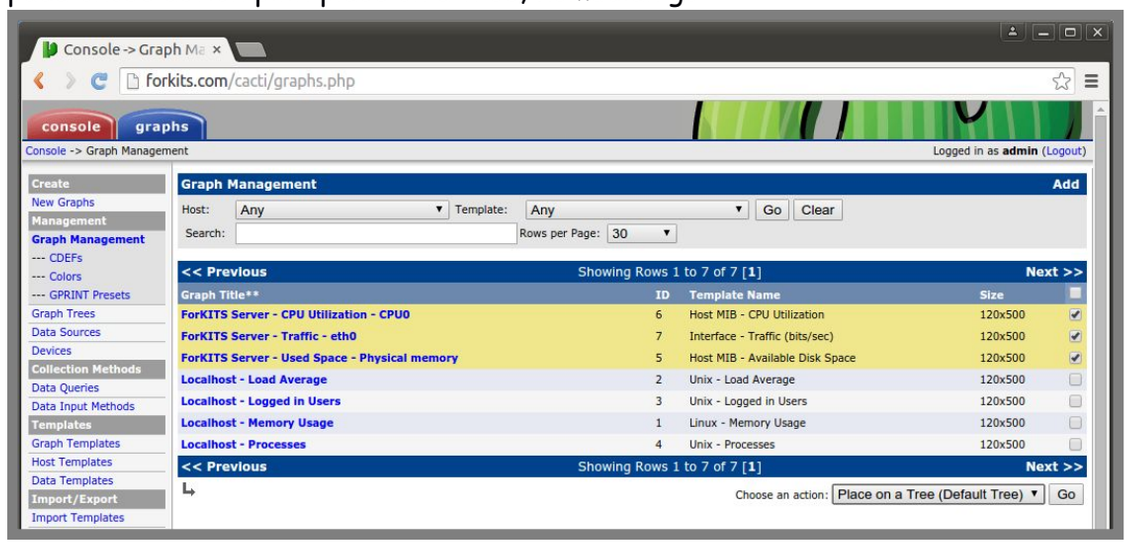
Gambar 13.24 Memilih apa saja yang akan dibuatkan grafik

Silahkan pilih warna sesuai dengan selera, atau langsung klik create saja untuk menggunakan warna default



Gambar 13.25 Konfigurasi grafik yang akan dibuat

Agar grafik-grafik dari komputer server tadi muncul dihalaman utama cacti, kita harus melakukan beberapa konfigurasi tambahan. Masuk pada bagian *graph management*, centang pada bagian yang ingin diletakkan dihalaman utama cacti, pada chose action pilih place on a tree, kemudian go



Gambar 13.26 Menambahkan grafik ke halaman utama cacti

Pada beberapa kasus, akan muncul pesan error seperti yang ditunjukkan gambar 13.27. Ini adalah bug dari cacti versi tersebut.

Gambar 13.27 Error saat managemen grafik pada cacti

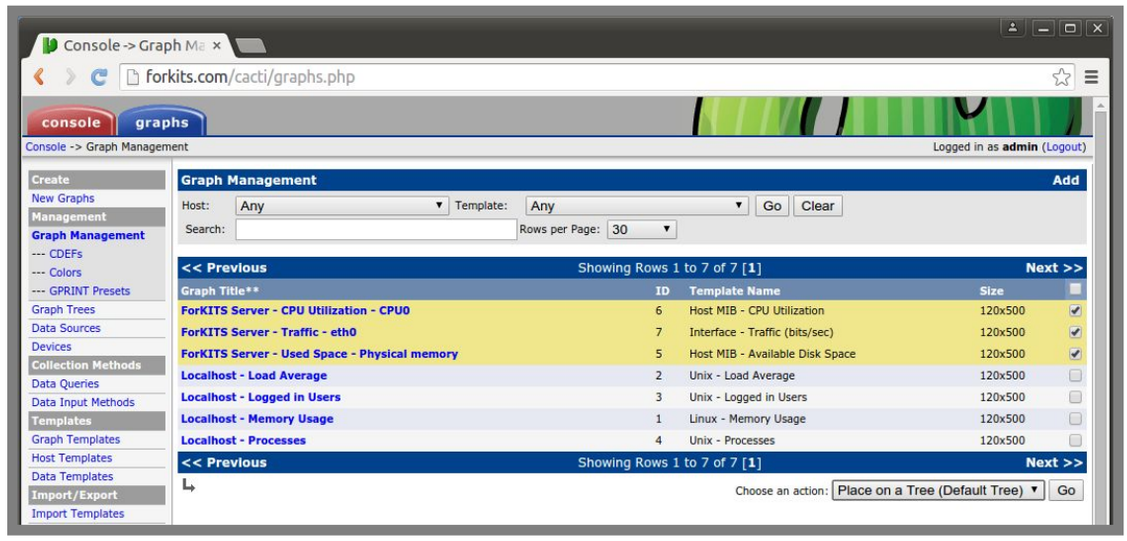
Untuk mengatasi hal tersebut, lakukan langkah-langkah sebagai berikut

```
root@forkits:~# nano /usr/share/cacti/site/graphs.php
.....
.....
.....
    /* ================= input validation ================= */
    //input_validate_input_number(get_request_var_post('drp_action'));
    /* ==================================================== */
.....
.....
.....
```

Gambar 13.28 Konfigurasi cacti untuk mengatasi error sebelumnya

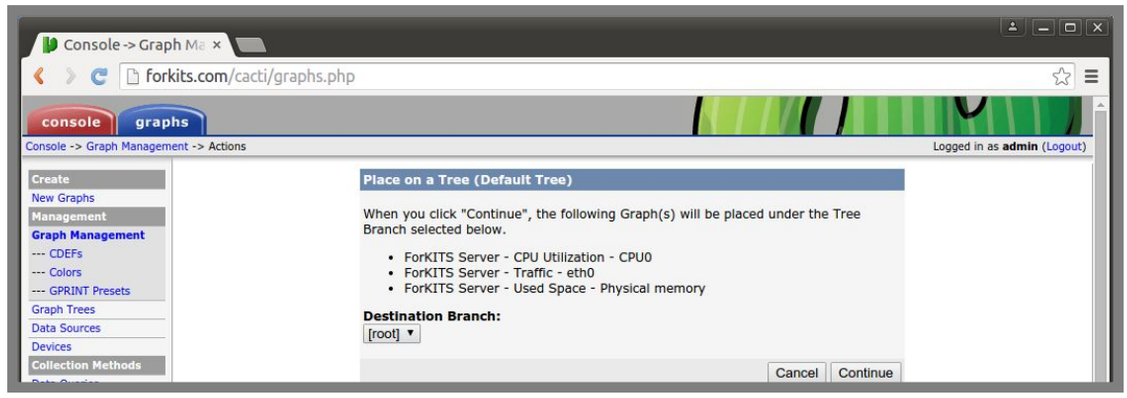
Perhatikan gambar diatas, terlihat bahwa kita melakukan perubahan pada teks warna hijau. Kita bisa mencari baris tersebut dengan kata kunci *drp_action*

Selanjutnya, ulangi langkah untuk menambahkan grafik dari komputer server ke halaman utama cacti



Gambar 13.29 Menambahkan garfik ke halaman utama cacti

Langsung saja klik continue



Gambar 13.30 Menambahkan garfik ke halaman utama cacti

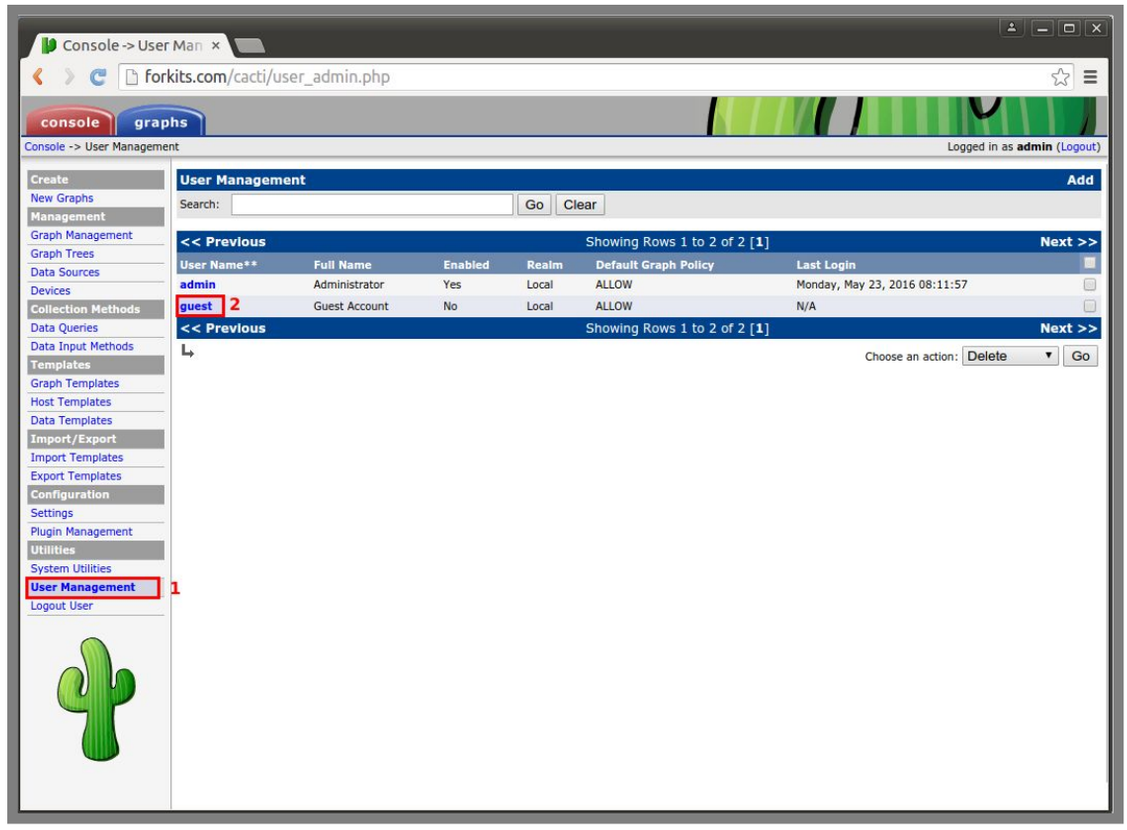
Sampai saat ini kita sudah berhasil membuat grafik untuk server, kemudian meletakkan grafik tersebut ke halaman utama cacti. Selanjutnya jangan lupa untuk mengembalikan konfigurasi pada *graphs.php* yang kita rubah tadi

```
root@forkits:~# nano /usr/share/cacti/site/graphs.php
    /* ================ input validation ================ */
    input_validate_input_number(get_request_var_post('drp_action'));
    /* ================================================== */
```

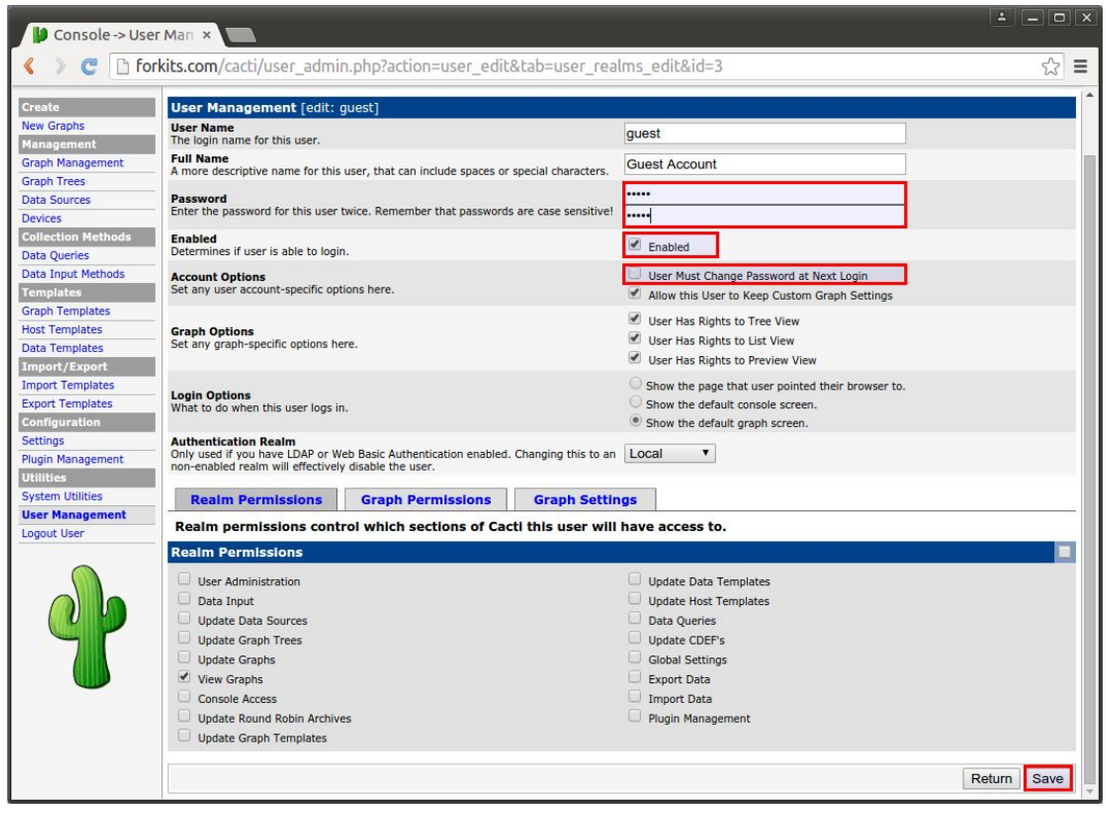
Gambar 13.31 Mengembalikan perubahan konfigurasi pada cacti

Selanjutnya, untuk alasan kemanan, kita tentunya tidak akan memberikan username dan password admin kepada orang lain. Namun demikian, mungkin saja kita membutuhkan bantuan orang lain untuk memonitoring server.

Oleh sebab itu, kita harus membuat user guest yang tujuannya hanya untuk memonitoring (tidak bisa melakukan perubahan konfigurasi pada cacti). Berikut langkah-langkah yang dapat kita lakukan

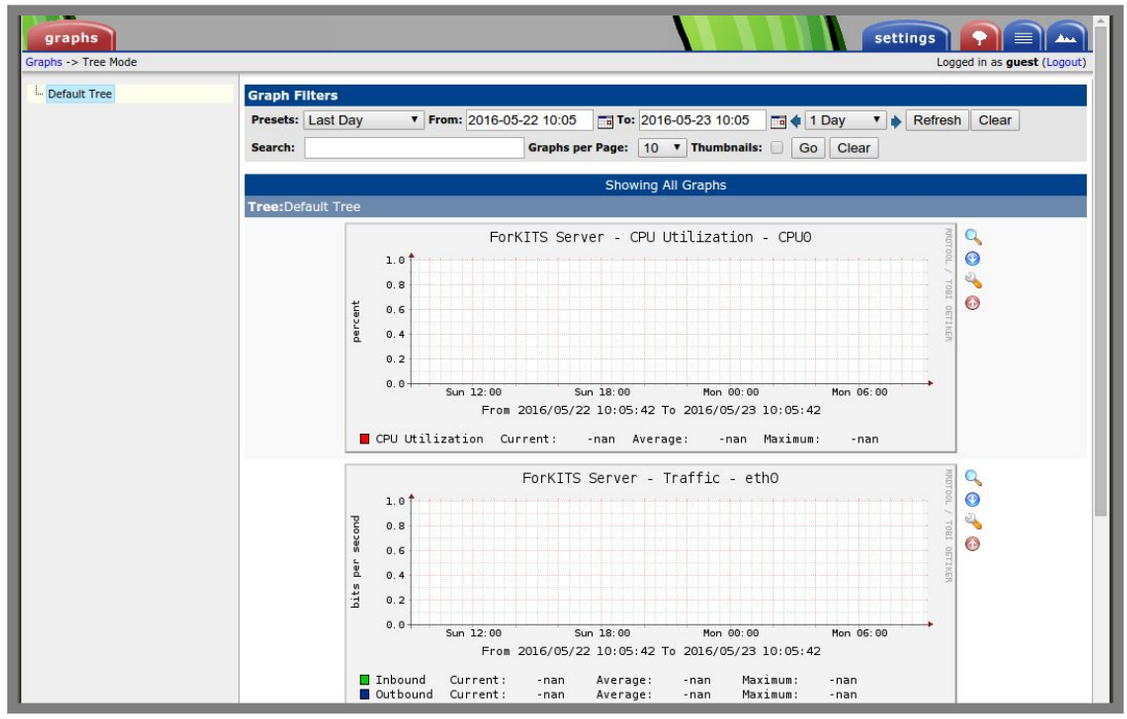


Gambar 13.32 Managemen user



Gambar 13.33 Konfigurasi user guest

Untuk percobaan, silahkan logout kemudian login kembail menggunakan user guest. Berikut tampilan halaman utama saat login menggunakan user guest



Gambar 13.33 Halaman utama cacti (belum ada grafik)

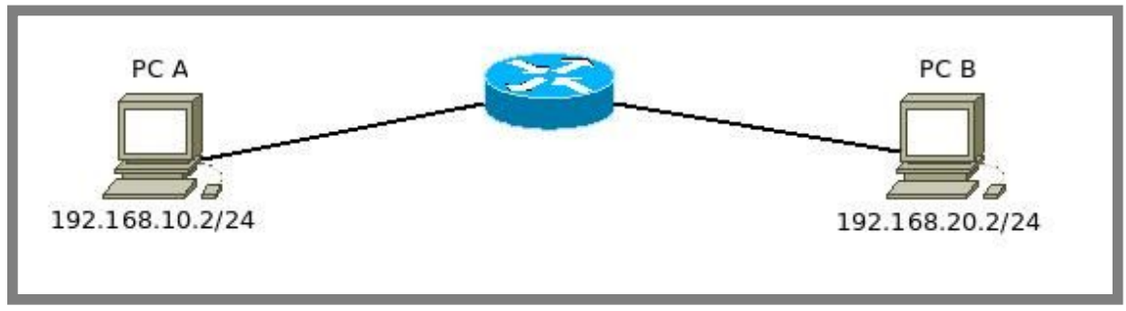
Perhatikan gambar diatas, terlihat bahwa pada grafik belum ada trafik sama sekali. Untuk melihat hasilnya, kita harus menungu sekitar 15-30 menit. Berikut hasil saat telah muncul grafik pada cacti

Gambar 13.34 Tampilan utama cacti setelah beberapa saat (sudah ada grafik)

---END OF CHAPTER---

Bab 14
Router Debian

Router merupakan sebuah perangkat yang berfungsi untuk menghubungkan dua atau lebih perangkat jaringan yang memiliki ip address berbeda jaringan. Sebagai contoh, perhatikan ilustrasi berikut



Gambar 14.1 Contoh penggunaan router

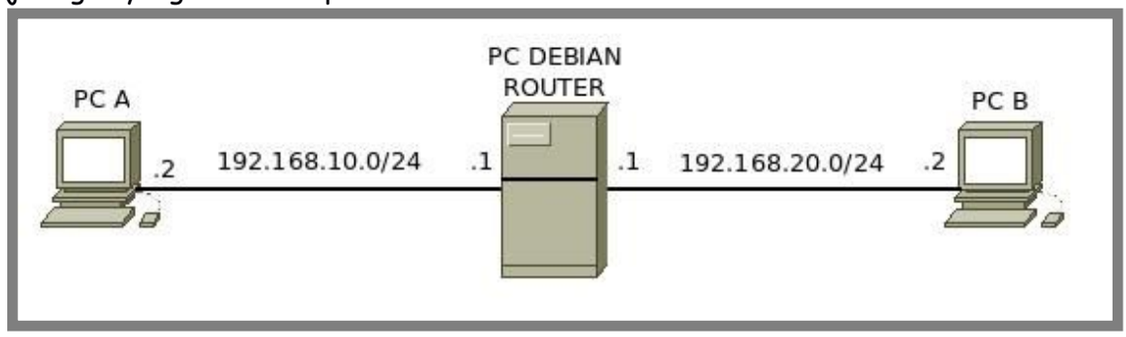
Perhatikan bahwa antara PC A dan PC B memiliki ip address yang berbeda jaringan, yaitu PC A memiliki ip address 192.168.10.2/24 sedangkan PC B memiliki ip address 192.168.20.2/24. Agar PC A dan PC B bisa saling berkomunikasi dan bertukar data, maka kita memerlukan sebuah router.

Saat ini terdapat beberapa vendor router yang beredar dipasaran, seperti mikrotik, cisco, juniper, dll. Perangkat-perangkat tersebut merupakan perangkat husus yang memang dibuat untuk menjadi sebuah router.

Selain menggunakan router-router tersebut diatas, kita bisa saja memanfaatkan sebuah komputer dengan sistem operasi debian untuk menjadi router. Tentunya syarat utama yang harus dipenuhi adalah komputer kita harus mempunyai minimal dua ethernet card. Hal ini dikarenakan fungsi router adalah menghubungkan dua atau perangkat jaringan yang memiliki ip address yang berbeda jaringan. Tentu tidak mungkin untuk menghubungkan dua perangkat jaringan jika hanya memiliki satu ethernet card.

Konfigurasi Router

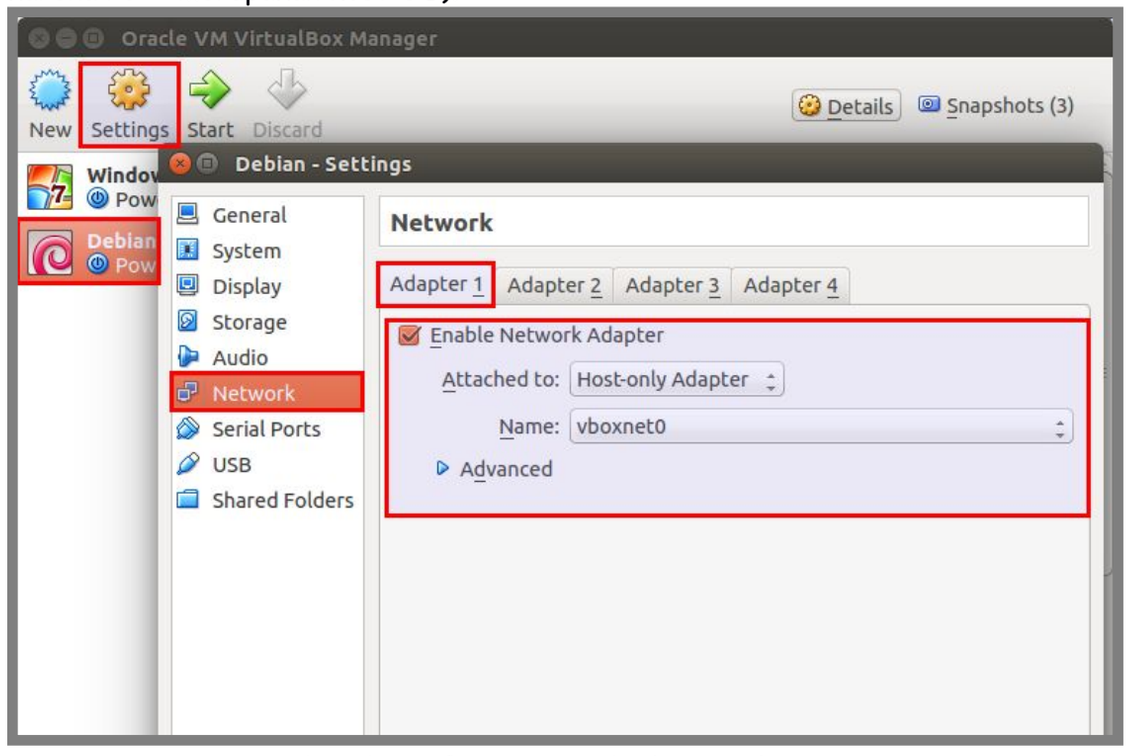
Mungkin akan sedikit sulit memahami konsep router jika hanya membaca teori-teori diatas. Karena itu, kita akan langsung praktik membuat sebuah router menggunakan komputer yang terinstall sistem operasi Debian. Berikut topologi jaringan yang akan kita praktikkan



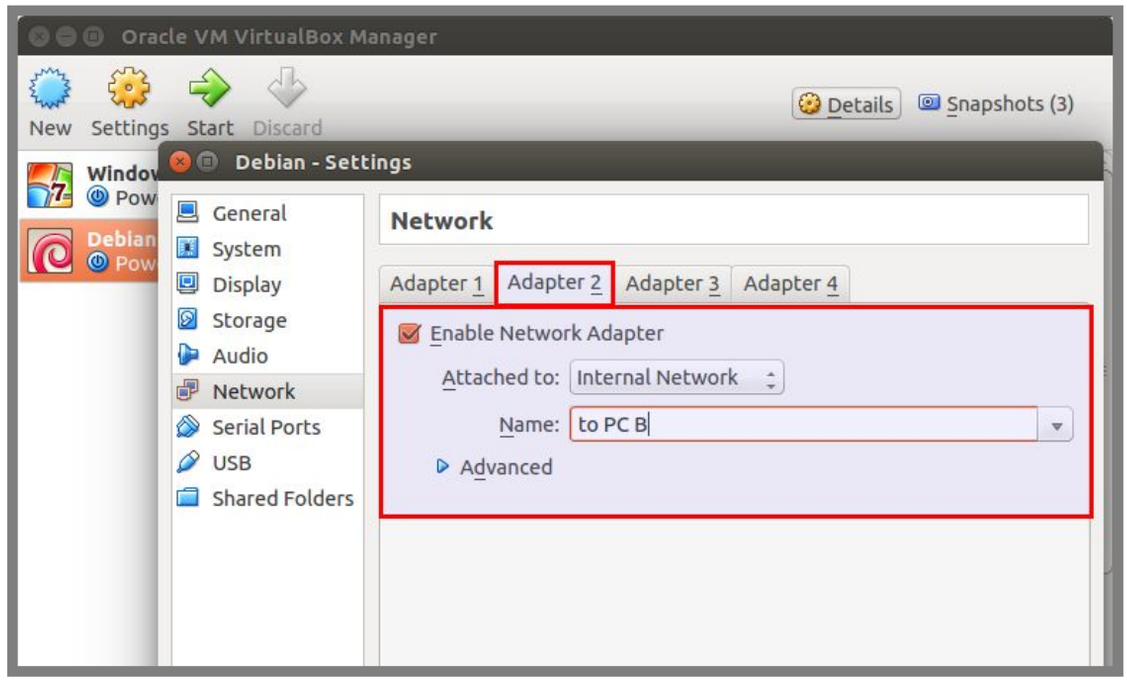
Gambar 14.2 Topologi jaringan untuk praktik konfigurasi router

Kita akan menggunakan guest OS debian yang telah kita gunakan pada praktik-praktik sebelumnya, hanya saja kita akan menambahkan ethernet. Hal ini dikarenakan saat ini kita masih mempunyai satu ethernet card, padahal kita tahu bahwa kita harus memiliki dua ethernet card sesuai topologi diatas.

Untuk PC A, kita akan menggunakan host OS (ubuntu), sedangkan PC B, kita akan menggunakan guest OS (windows). Berikut konfigurasi network adapter pada PC ROUTER (kita tidak perlu melakukan konfigurasi network adapter pada PC A, karena PC A merupakan host OS).

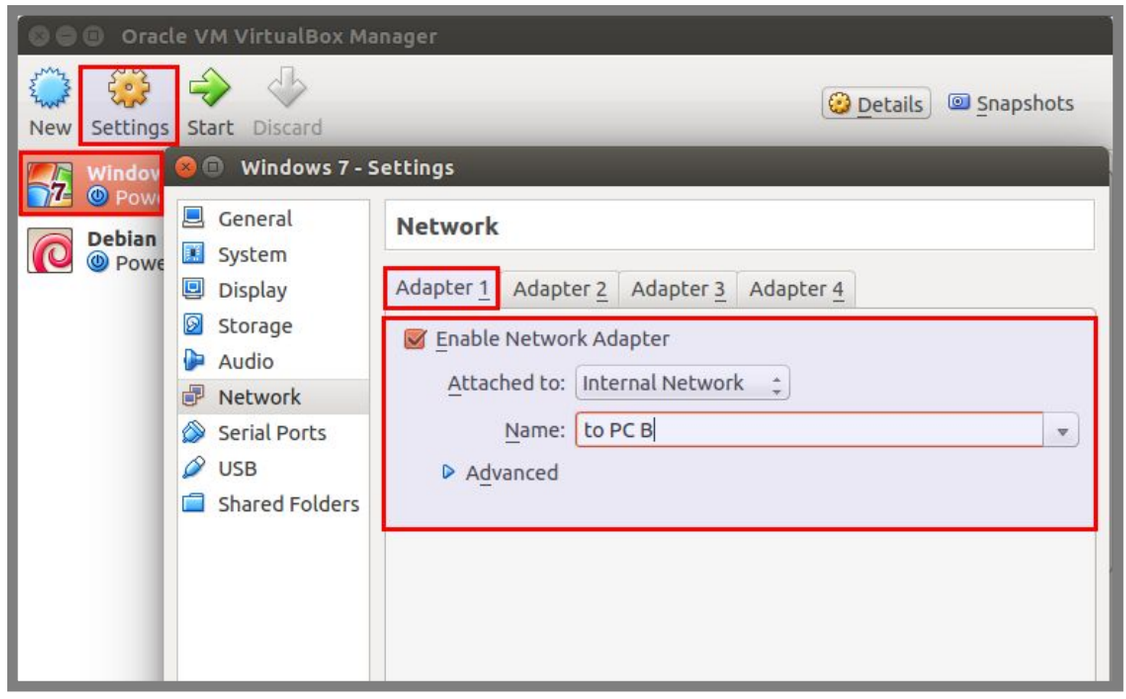


Gambar 14.3 Konfigurasi network adapter pada router



Gambar 14.4 Konfigurasi network adapter pada router

Sedangkan untuk konfigurasi network adapter pada PC B adalah sebagai berikut



Gambar 14.5 Konfigurasi network adapter pada PC B

Sampai saat ini, kita telah menghubungkan tiga komputer seperti topologi pada gambar 14.2. Selanjutnya kita harus melakukan konfigurasi pada router debian. Konfigurasi pertama yang harus dilakukan adalah konfigurasi ip address.

Berikut konfigurasi ip address pada router debian

```
root@forkits:~# nano /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.10.1
netmask 255.255.255.0

auto eth1
iface eth1 inet static
address 192.168.20.1
netmask 255.255.255.0
```

Gambar 14.6 Konfigurasi ip address pada router

Jangan lupa untuk merestart service network

```
root@forkits:~# service networking restart
[....] Running /etc/init.d/networking restart is deprecated because it may not
r[warnble some interfaces ... (warning).
[ ok ] Reconfiguring network interfaces...done.
root@forkits:~#
```

Gambar 14.7 Restart service networking pada router

Berikut hasil konfigurasi ip address yang baru saja kita lakukan

```
root@forkits:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1b:29:b8
          inet addr:192.168.10.1 Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1b:29b8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:738 errors:0 dropped:0 overruns:0 frame:0
          TX packets:588 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:65437 (63.9 KiB)  TX bytes:83023 (81.0 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:83:08:8d
          inet addr:192.168.20.1 Bcast:192.168.20.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe83:88d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:6317 (6.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:120 errors:0 dropped:0 overruns:0 frame:0
          TX packets:120 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11431 (11.1 KiB)  TX bytes:11431 (11.1 KiB)

root@forkits:~#
```

Gambar 14.8 Hasil konfigurasi ip address pada router

Setelah melakukan konfigurasi ip address, langkah selanjutnya yang harus kita lakukan adalah mengaktifkan fungsi routing pada debian.

Secara sederhana, routing dapat diartikan sebagai proses meneruskan paket dari satu jaringan ke jaringan yang lain. Jika mengacu pada topologi gambar 14.2, routing adalah proses pengiriman data dari PC A ke PC B atau sebaliknya.

Secara default, fungsi routing ini tidak aktif pada debian. Kita harus melakukan beberapa konfigurasi untuk mengaktifkan fungsi routing ini. Pada umumnya, fungsi routing ini juga disebut sebagai packet forward (penerusan paket).

Berikut konfigurasi yang perlu dilakukan

```
root@forkits:~# nano /etc/sysctl.conf
.....
.....
.....
.....
# Uncomment the next two lines to enable Spoof protection (reverse-path
filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
.....
.....
.....
.....
```

Gambar 14.9 Konfigurasi untuk mengaktifkan fungsi routing

Setelah melakukan konfigurasi diatas, kita harus merestart router debian untuk mengaktifkan perubahan yang baru saja kita lakukan diatas. Atau kita hanya perlu menggunakan perintah *sysctl -p*

```
root@forkits:~# sysctl -p
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
root@forkits:~#
```

Gambar 14.10 Mengaktifkan fungsi routing

Sampai saat ini kita telah selesai membuat sebuah router menggunakan komputer debian. Langkah selanjutnya adalah melakukan konfigurasi ip address pada PC A dan PC B.

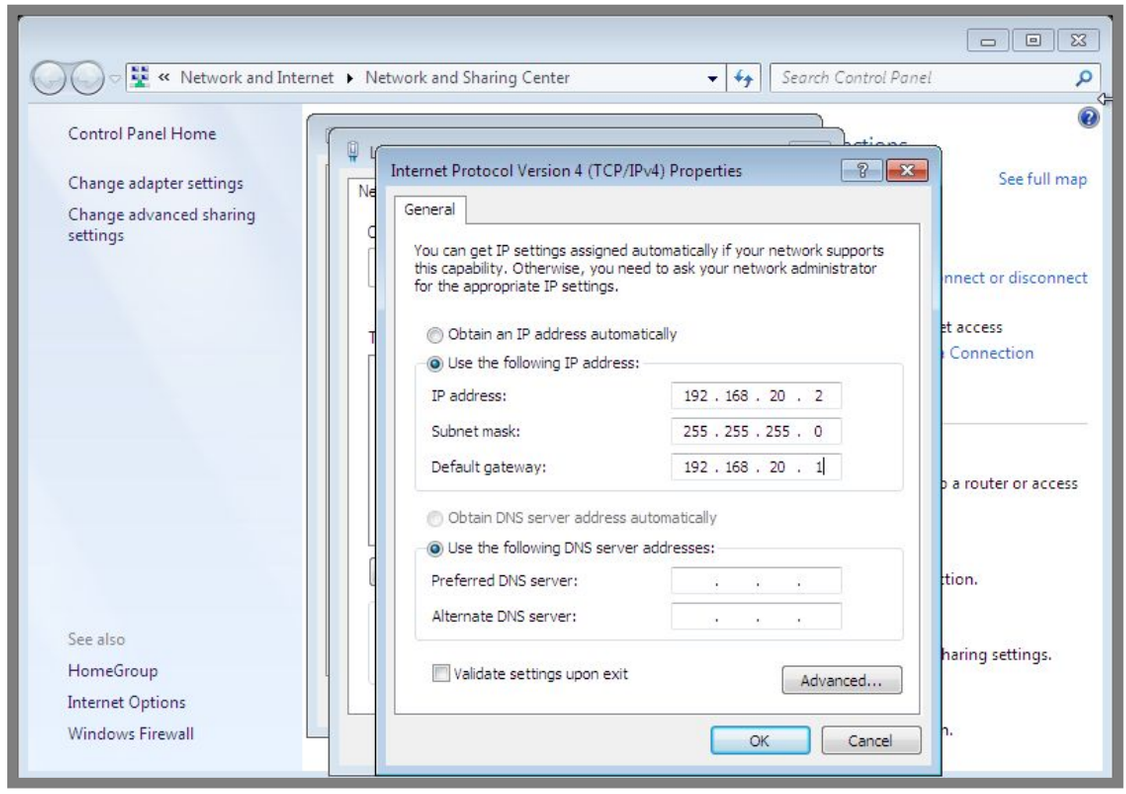
Berikut konfigurasi ip address yang perlu dilakukan pada PC A

```
admin@ubuntu:~$ sudo ifconfig vboxnet0 192.168.10.2/24
[sudo] password for admin: (tak terlihat)
admin@ubuntu:~$ sudo route add default gw 192.168.10.1
admin@ubuntu:~$
```

Gambar 14.11 Konfigurasi ip address pada PC A

Perhatikan gambar diatas, perintah pertama digunakan untuk konfigurasi ip address, sedangkan perintah kedua digunakan untuk konfigurasi gateway. Gateway diperlukan jika suatu komputer ingin berkomunikasi dengan komputer lain yang berbeda jaringan.

Selanjutnya berikut konfigurasi ip address pada PC B

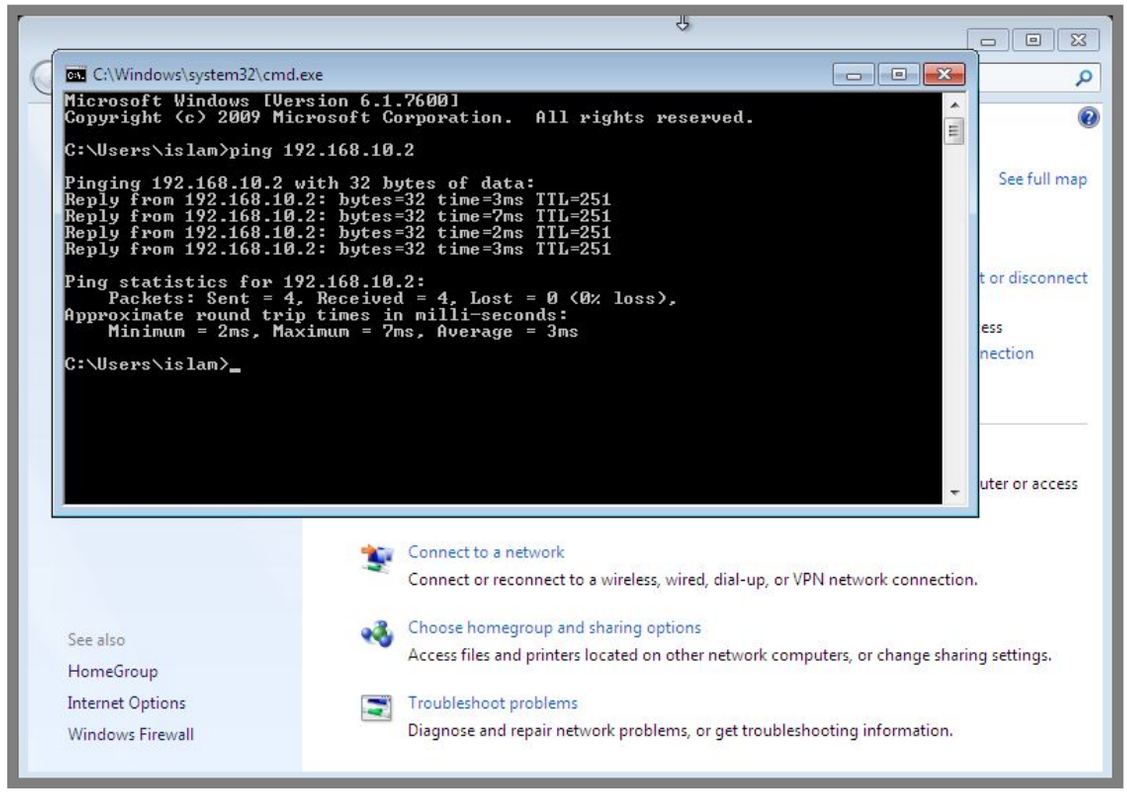


Gambar 14.12 Konfigurasi ip address pada PC B

Berikut hasil pengujian ping yang dilakukan dari PC A ke PC B dan sebaliknya

```
admin@ubuntu:~$ ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=127 time=2.22 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=127 time=1.33 ms
64 bytes from 192.168.20.2: icmp_seq=3 ttl=127 time=2.21 ms
```

Gambar 14.13 Pengujian ping dari PC A ke PC B

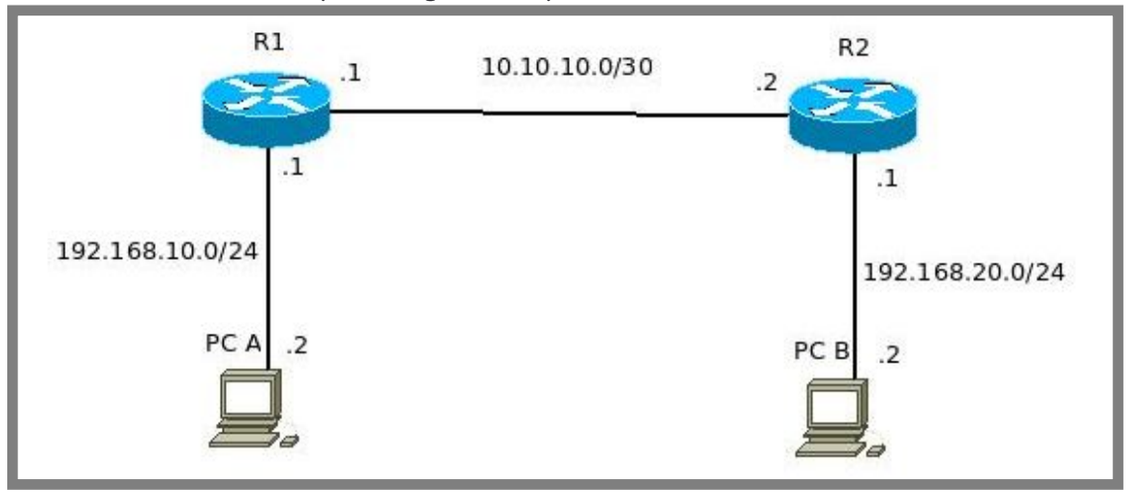


Gambar 14.14 Pengujian ping dari PC B ke PC A

Konfigurasi Routing Static

Routing static adalah sebuah teknik routing yang dilakukan dengan memasukkan *entry route* kedalam tabel routing secara manual oleh administrator jaringan. Tabel routing itu sendiri adalah sebuah tabel yang berisi informasi mengenai network-network yang dapat dituju oleh sebuah router. Jadi jika suatu network tidak ada dalam tabel routing, maka network tersebut tidak dapat dituju (dihubungi) oleh router.

Untuk memahami konsep routing static, perhatikan ilustrasi berikut



Gambar 14.15 Topologi jaringan untuk praktik konfigurasi routing static

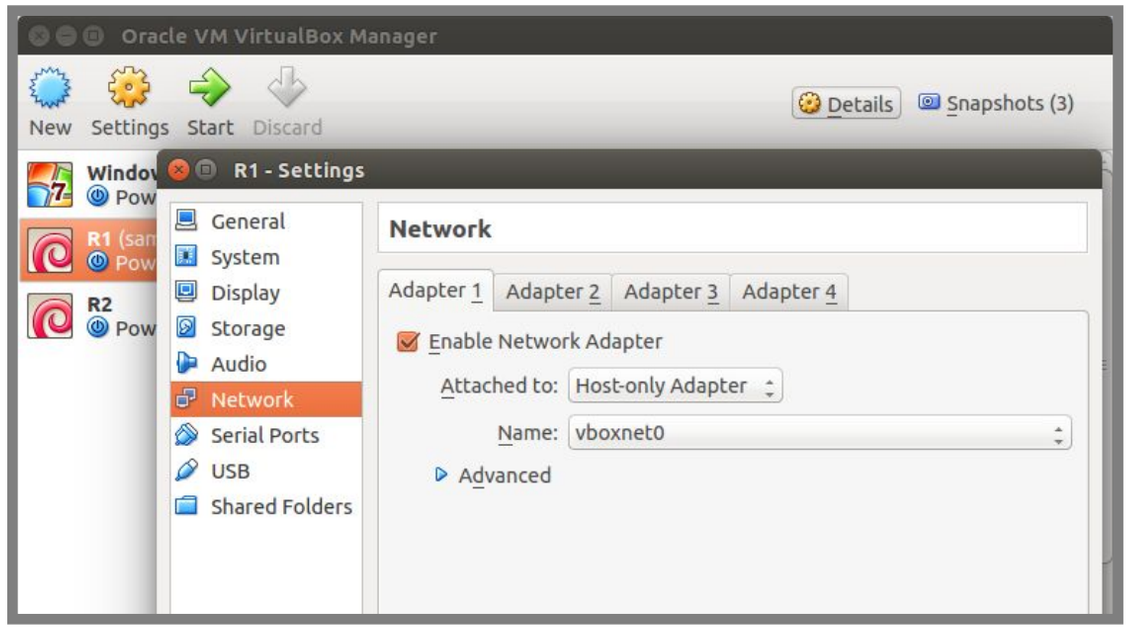
Perhatikan gambar diatas, terlihat bahwa terdapat dua router pada topologi tersebut, yaitu R1 dan R2. R1 terhubung langsung dengan network 10.10.10.0/30 dan 192.168.10.0/24, sedangkan R2 terhubung langsung dengan network 10.10.10.0/30 dan 192.168.20.0/24.

Network yang terhubung langsung dengan suatu router disebut dengan directly connected. Network ini akan otomatis dimasukkan oleh router kedalam tabel routingnya.

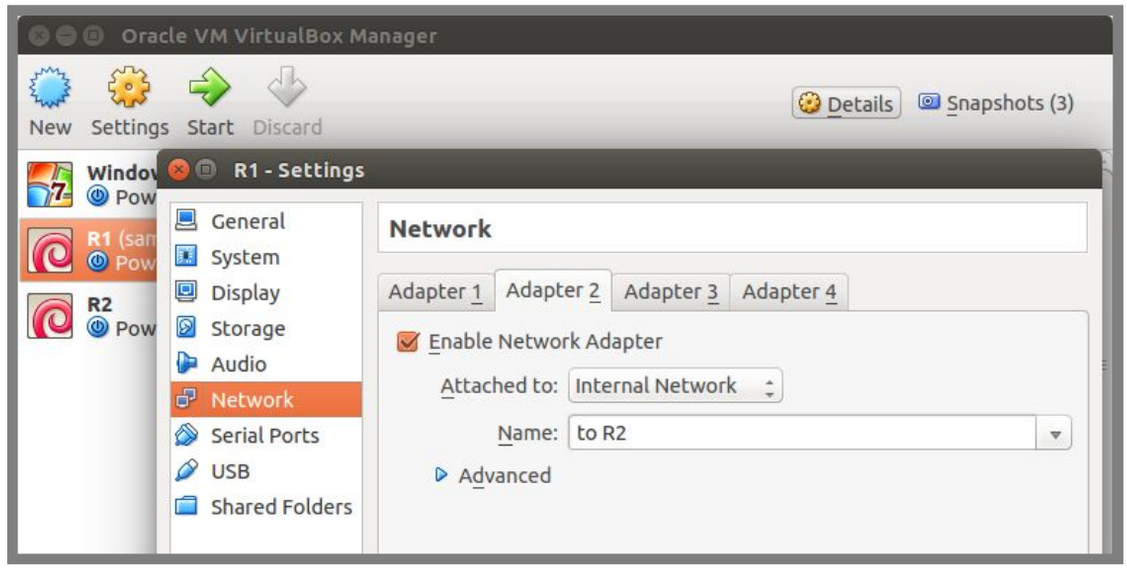
Selain directly connected, ada satu istilah lagi, yaitu remote network. Remote network adalah sebuah network yang tidak terhubung langsung dengan suatu router. Network ini tidak akan ada dalam tabel routing jika administrator jaringan tidak menambahkannya ke tabel routing. Bagi R1, remote networknynya adalah 192.168.10.0/24, sedangkan bagi R2, remote networknya adalah 192.168.10.0/24.

Mungkin penjelasan diatas sedikit membingungkan, karena itu kita akan langsung praktik sesuai dengan topologi pada gambar 14.15. Kita akan menggunakan guest os dengan sistem operasi debian sebagai R1 dan R2, sedangkan untuk PC A kita akan menggunakan host os (ubuntu) dan sebagai PC B kita akan menggunakan guest os (windows).

Berikut konfigurasi network adapter pada R1



Gambar 14.16 Konfigurasi network adapter pada R1



Gambar 14.17 Konfigurasi network adapter pada R1

Untuk R2, konfigurasi network adapter adalah sebagai berikut

Gambar 14.18 Konfigurasi network adapter pada R2

Gambar 14.19 Konfigurasi network adapter pada R1

Berikut konfigurasi network adapter untuk PC B

Gambar 14.20 Konfigurasi network adapter pada PC B

Langkah pertama yang harus dilakukan tentunya adalah konfigurasi ip address pada R1 dan R2. Berikut konfigurasi ip address pada R1

```
root@R1:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1b:29:b8
          inet addr:192.168.10.1 Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1b:29b8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:46 errors:0 dropped:0 overruns:0 frame:0
          TX packets:101 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7605 (7.4 KiB)  TX bytes:12483 (12.1 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:83:08:8d
          inet addr:10.10.10.1  Bcast:10.10.10.3  Mask:255.255.255.252
          inet6 addr: fe80::a00:27ff:fe83:88d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4200 (4.1 KiB)  TX bytes:6678 (6.5 KiB)
```

Gambar 14.21 Konfigurasi IP Address pada R1

Sedangkan konfigurasi ip address pada R2 adalah sebagai berikut

```
root@R2:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:b3:53:1a
          inet addr:10.10.10.2  Bcast:10.10.10.3  Mask:255.255.255.252
          inet6 addr: fe80::a00:27ff:feb3:531a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:118 errors:0 dropped:0 overruns:0 frame:0
          TX packets:124 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17137 (16.7 KiB)  TX bytes:9243 (9.0 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:1f:4b:27
          inet addr:192.168.20.1 Bcast:192.168.20.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1f:4b27/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:936 (936.0 B)
```

Gambar 14.22 Konfigurasi IP Address pada R2

Langkah selanjutnya adalah mengaktifkan fungsi routing di R1 dan R2

```
root@R1:~# sysctl -p
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
root@R1:~#
```

Gambar 14.23 Mengaktifkan fungsi routing di R1

```
root@R2:~# sysctl -p
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
root@R2:~#
```

Gambar 14.24 Mengaktifkan fungsi routing di R1

Sampai saat ini kita sudah selesai mengkonfigurasi R1 dan R2. Namun tentu saja R1 dan R2 masih belum mengetahui remote networknya masing-masing. Sehingga untuk saat ini R1 masih belum bisa menjangkau 192.168.20.0/24 dan R2 juga masih belum bisa menjangkau 192.168.10.0/24 yang mengakibatkan PC A dan PC B masih belum bisa saling berkomunikasi.

Untuk membuktikan pernyataan-pernyataan diatas, perhatikan tabel routing dari R1 dan R2 berikut

```
root@R1:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.10.0      *               255.255.255.252 U     0      0        0 eth1
192.168.10.0    *               255.255.255.0   U     0      0        0 eth0
root@R1:~#
```

Gambar 14.25 Tabel routing R1

```
root@R2:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.10.0      *               255.255.255.252 U     0      0        0 eth0
192.168.20.0    *               255.255.255.0   U     0      0        0 eth1
root@R2:~#
```

Gambar 14.26Tabel routing R2

Perhatikan tabel routing R1 dan R2 diatas, terlihat bahwa R1 masih belum mengetahui network 192.168.20.0/24 dan R2 juga belum mengetahui network 192.168.10.0/24.

Berikut perintah yang dapat kita gunakan untuk menambahkan remote network ke R1 dan R2

```
root@R1:~# route add -net 192.168.20.0 netmask 255.255.255.0 gw 10.10.10.2
root@R1:~#
```

Gambar 14.27 Konfigurasi routing static pada R1

```
root@R2:~# route add -net 192.168.10.0 netmask 255.255.255.0 gw 10.10.10.1
root@R2:~#
```

Gambar 14.28 Konfigurasi routing static pada R2

Periksa kembali tabel routing di R1 dan R2

```
root@R1:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.10.0      *               255.255.255.252 U     0      0        0 eth1
192.168.10.0    *               255.255.255.0   U     0      0        0 eth0
192.168.20.0    10.10.10.2      255.255.255.0   UG    0      0        0 eth1
root@R1:~#
```

Gambar 14.29 Tabel routing pada R1

```
root@R2:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.10.0      *               255.255.255.252 U     0      0        0 eth0
192.168.10.0    10.10.10.1      255.255.255.0   UG    0      0        0 eth0
192.168.20.0    *               255.255.255.0   U     0      0        0 eth1
root@R2:~#
```

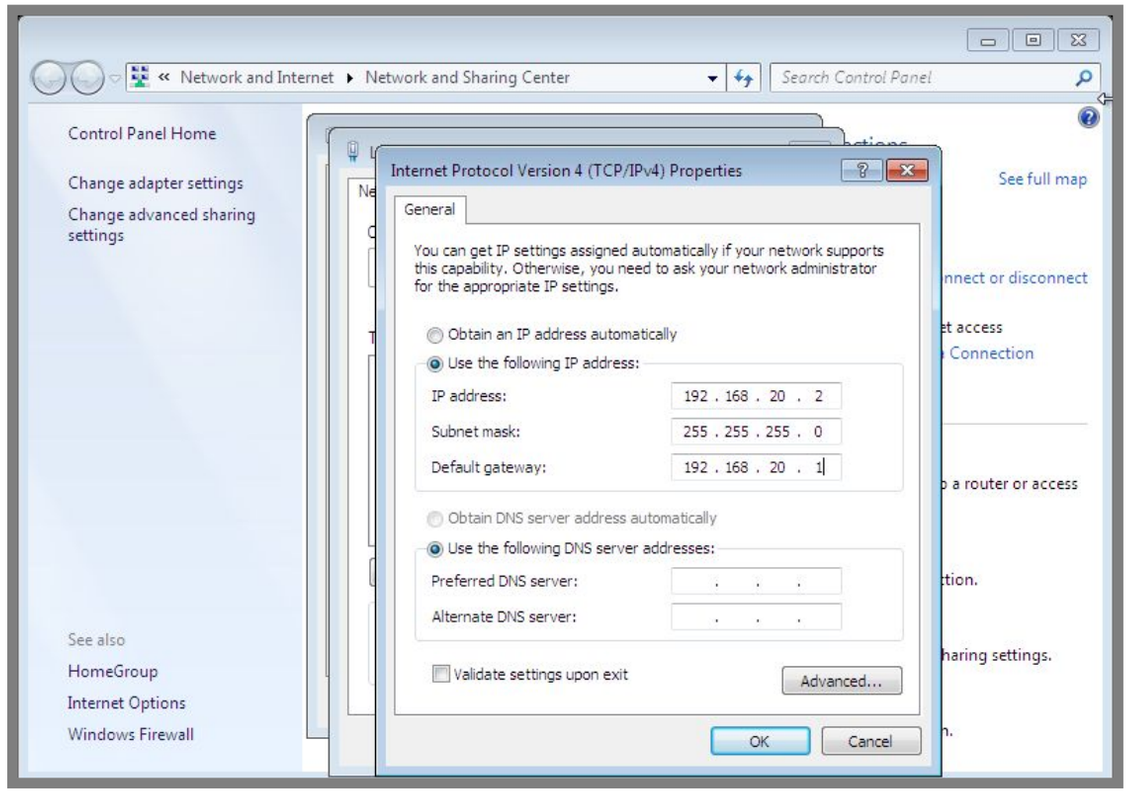
Gambar 14.30 Tabel routing pada R2

Perhatikan bahwa saat ini antara R1 dan R2 masing-masing telah mengetahui remote networknya. Sehingga saat ini seharusnya antara PC A dan PC B sudah bisa saling berkomunikasi.

Tentu saja antara PC A dan PC B harus dikonfigurasi ip address dan gateway terlebih dahulu.

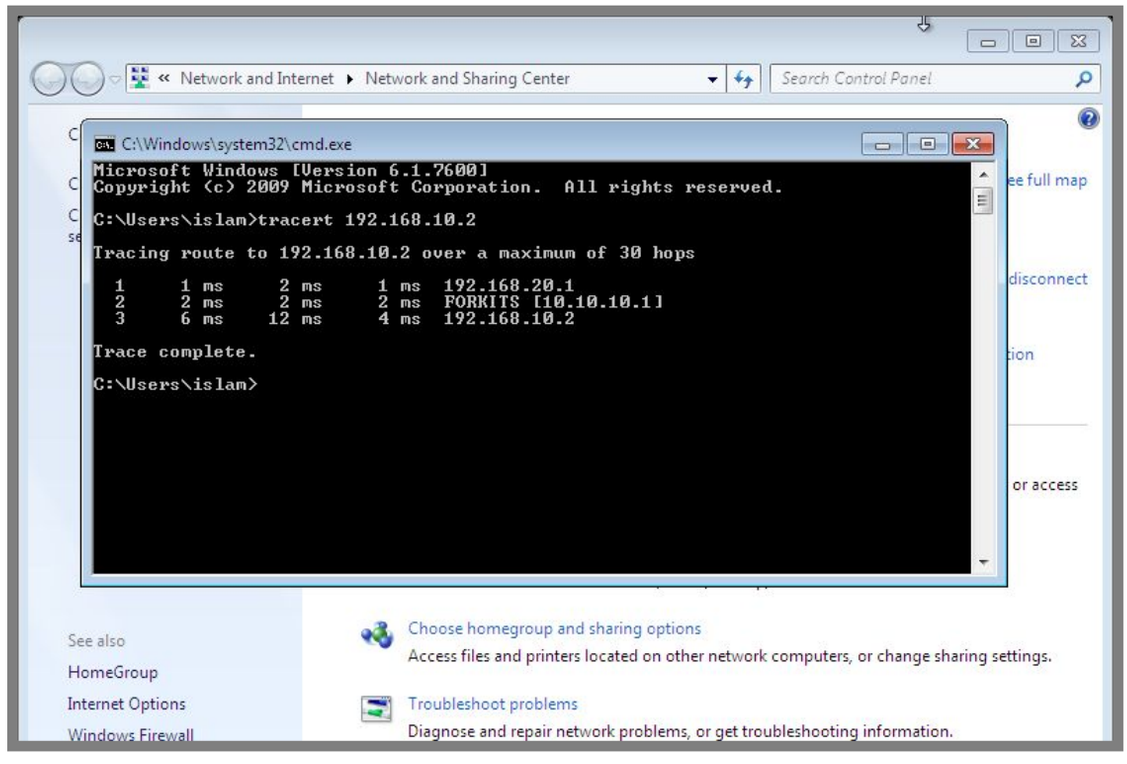
```
admin@ubuntu:~$ sudo ifconfig vboxnet0 192.168.10.2/24
[sudo] password for admin: (tak terlihat)
admin@ubuntu:~$ sudo route add default gw 192.168.10.1
admin@ubuntu:~$
```

Gambar 14.31 Konfigurasi ip address pada PC A



Gambar 14.32 Konfigurasi ip address pada PC B

Selanjutnya untuk melakukan pengujian kita bisa memanfaatkan tool ping maupun traceroute. Tentunya kita sudah tahu apa itu fungsi dari ping, sedangkan traceroute adalah tool yang digunakan untuk melihat jalur yang dilewati suatu paket untuk mencapai suatu tujuan. Berikut contoh penggunaan trace route dari PC B ke PC A



Gambar 14.33 Traceroute dari PC B ke PC A

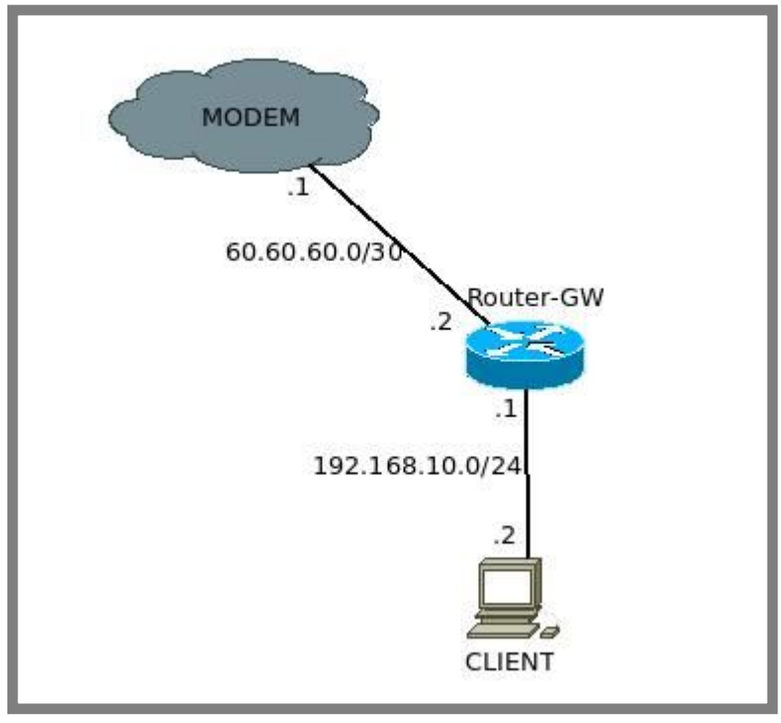
Perhatikan gambar diatas, terlihat bahwa jika PC B ingin menuju ke PC A, PC B harus melewati 192.168.20.1 (R2) kemudian 10.10.10.1 (R1), baru bisa menuju ke PC A (192.168.10.2).

Konfigurasi Router Gateway

Pada sub bab sebelumnya, kita telah membahas bagaimana cara membuat router menggunakan debian. Pada sub bab ini kita akan belajar bagaimana membuat router gateway menggunakan debian.

Kemudian apa bedanya router dengan router gateway?? Kita tentu sudah tahu jika router berfungsi untuk menghubungkan dua atau lebih perangkat jaringan yang memiliki ip address berbeda jaringan.

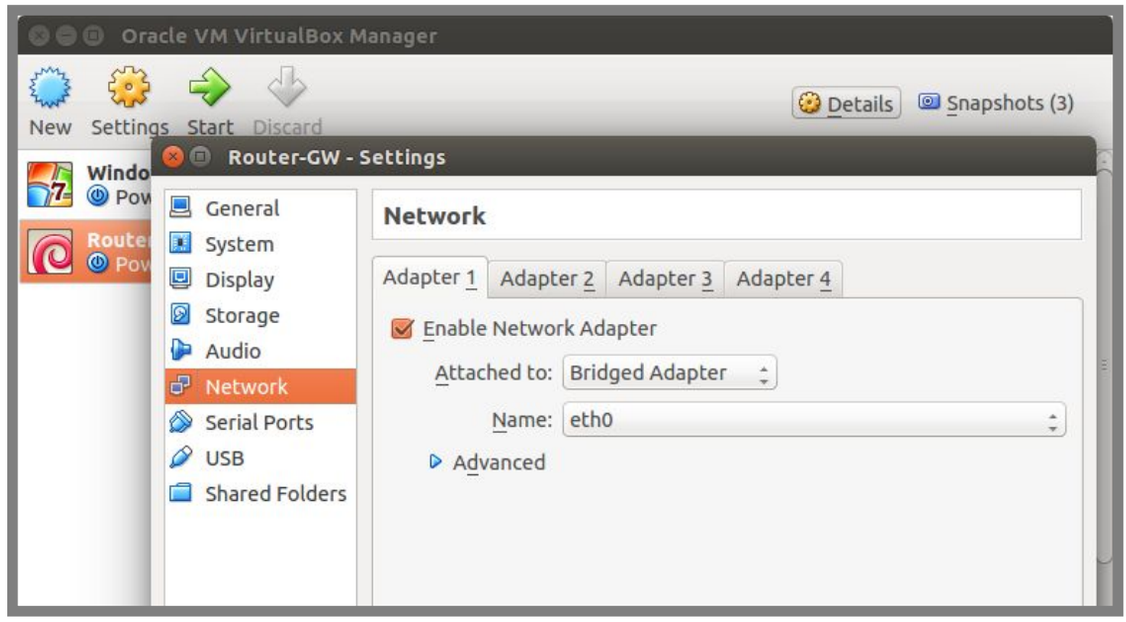
Sebenarnya router gateway juga memiliki pengertian yang sama dengan router. Hanya saja router gateway dihususkan untuk menghubungkan komputer-komputer yang ada di jaringan lokal dengan jaringan internet. Perhatikan ilustrasi berikut



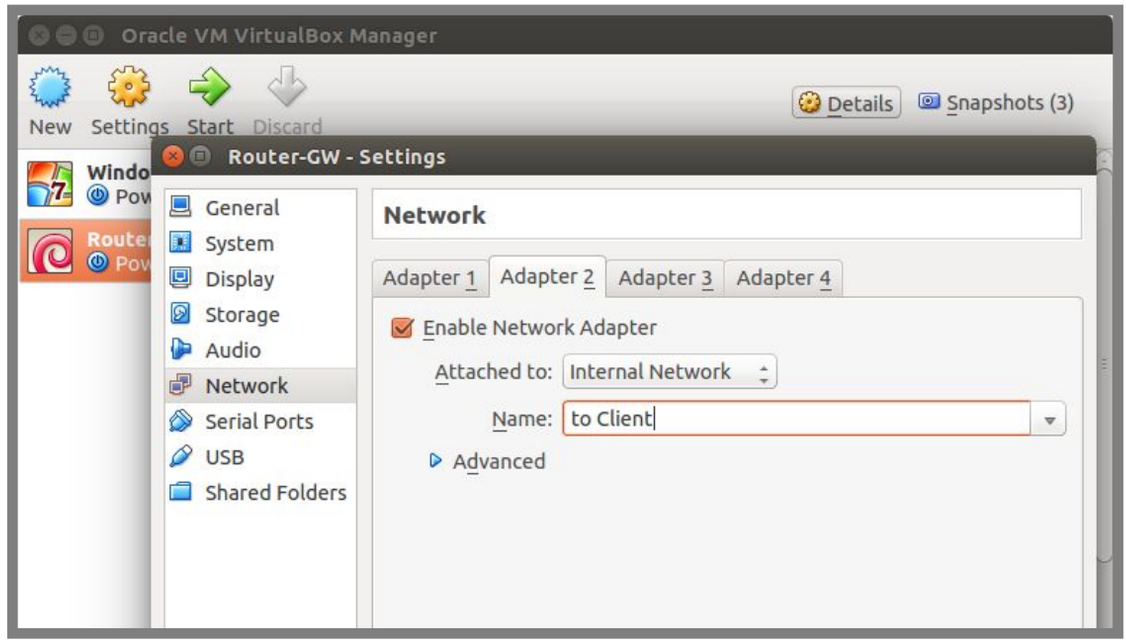
Gambar 14.34 Topologi jaringan untuk praktik konfigurasi router gateway

Perhatikan gambar diatas, terlihat bahwa Router-GW mendapat akses internet melalui modem. Tujuan dari Router-GW adalah melakukan sharing koneksi internet ke komputer client. Sehingga nantinya komputer client dapat berselancar di dunia internet.

Perlu diketahui bahwa untuk praktik pada sub bab ini, kita memerlukan koneksi internet. Diasumsikan bahwa saat ini kita mendapat koneksi internet menggunakan kabel UTP, sehingga berikut konfigurasi network adapter yang perlu dilakukan pada Router-GW

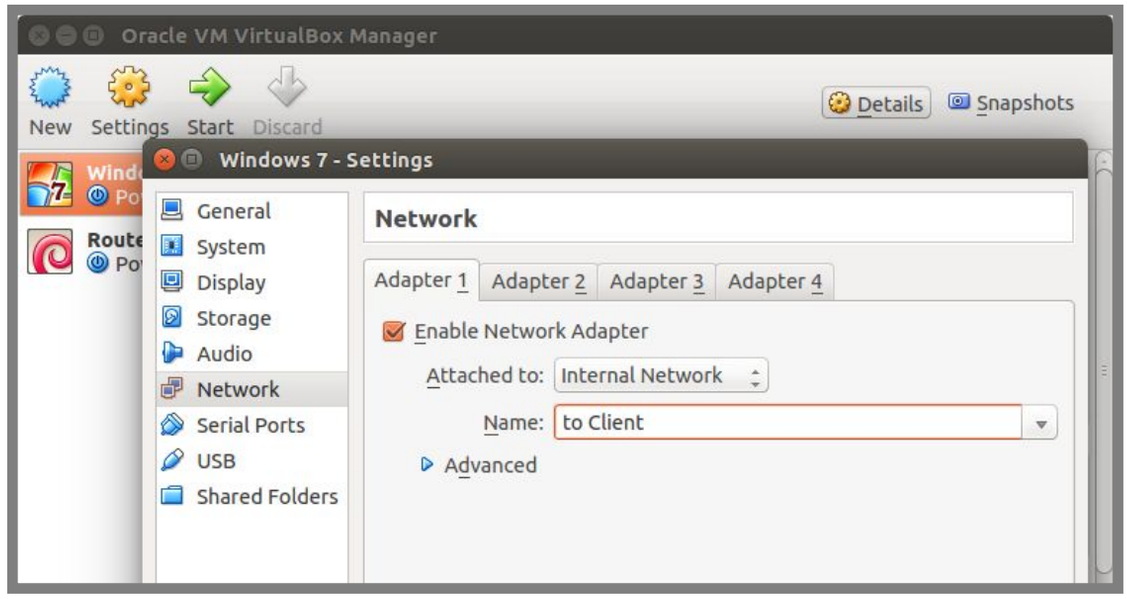


Gambar 14.35 Konfigurasi network adapter pada Router-GW



Gambar 14.36 Konfigurasi network adapter pada Router-GW

Sedangkan pada client, berikut konfigurasi network adapternya



Gambar 14.37 Konfigurasi network adapter pada client

Selanjutnya hal yang harus kita lakukan adalah konfigurasi ip address pad Router-GW

```
root@Router-GW:~# nano /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 60.60.60.2
netmask 255.255.255.252
gateway 60.60.60.1

auto eth1
iface eth1 inet static
address 192.168.10.1
netmask 255.255.255.0
```

Gambar 14.38 Konfigurasi ip address pada Router-GW

Jangan lupa untuk merestart service network

```
root@Router-GW:~# service networking restart
[....] Running /etc/init.d/networking restart is deprecated because it may not
r[warnble some interfaces ... (warning).
[ ok ] Reconfiguring network interfaces...done.
root@Router-GW:~#
```

Gambar 14.39 Restart service network

Selanjutnya kita harus konfigurasi dns resolver di komputer server. Kita bisa mengarahkan ke open dns yang ada diinternet, misal open dns nawala

```
root@Router-GW:~# nano /etc/resolv.conf
nameserver 180.131.144.144
nameserver 180.131.145.145
```

Gambar 14.40 Konfigurasi dns resolver

Sampai saat ini seharusnya Router-GW sudah bisa ping ke google.com

```
root@Router-GW:~# ping google.com
64 bytes from 74.125.200.139: icmp_seq=3 ttl=51 time=45.8 ms
64 bytes from 74.125.200.139: icmp_seq=3 ttl=51 time=45.8 ms
64 bytes from 74.125.200.139: icmp_seq=3 ttl=51 time=45.8 ms
64 bytes from 74.125.200.139: icmp_seq=3 ttl=51 time=45.8 ms
```

Gambar 14.41 Pengujian ping ke domain di internet

Hal yang harus dilakukan selanjutnya adalah mengaktifkan fungsi routing di Router-GW.

```
root@Router-GW:~# sysctl -p
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
root@Router-GW:~#
```

Gambar 14.42 Mengaktifkan fungsi routing di Router-GW

Selanjutnya kita harus membuat sebuah rule firewall nat yang berfungsi untuk menyembunyikann client (jaringan lokal) dari jaringan internet. Hal ini dikarenakan ip address yang dikenali oleh jaringan internet hanya ip address eth0 Router-GW saja (60.60.60.2), jaringan internet tidak mengenal ip address pada jaringan local (192.168.10.0/24). Sehingga jika kita menginginkan jaringan local bisa mengakses internet, kita harus menyembunyikannya.

Berikut langkah-langkah konfigurasi firewall nat yang perlu dilakukan

```
root@Router-GW:~# nano /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
exit 0
```

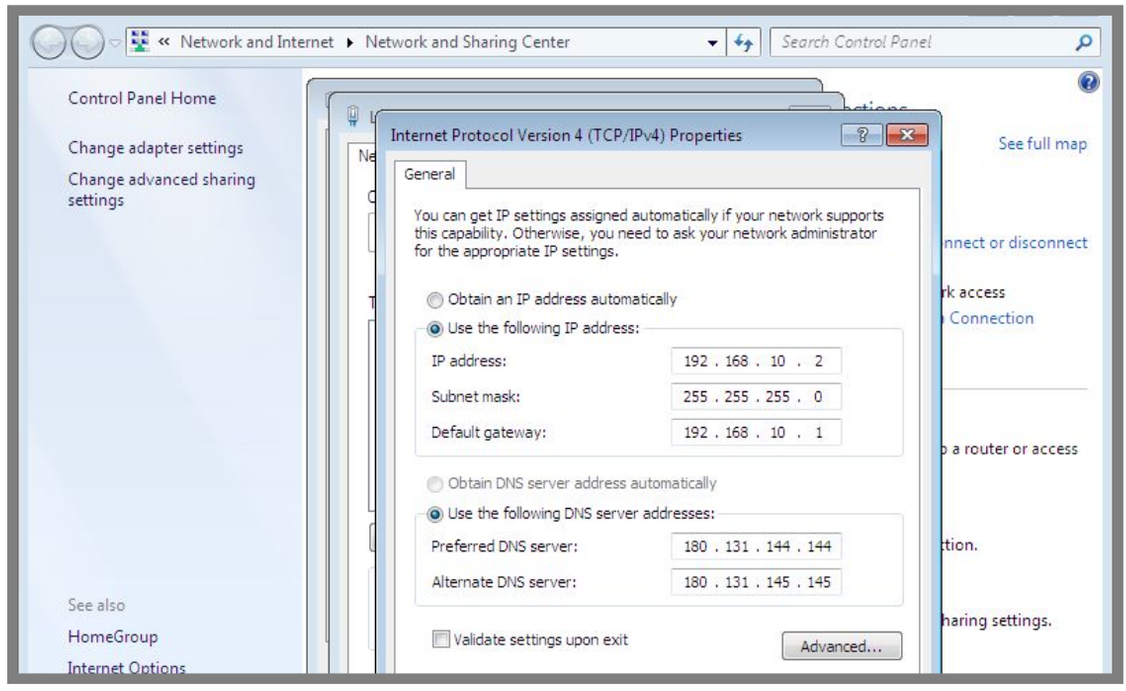
Gambar 14.43 Konfigurasi firewall nat

Untuk menyimpan perubahan diatas, kita harus merestart Router-GW atau cukup dengan menjalankan perintah */etc/rc.local/* seperti berikut

```
root@Router-GW:~# /etc/rc.local
root@Router-GW:~#
```

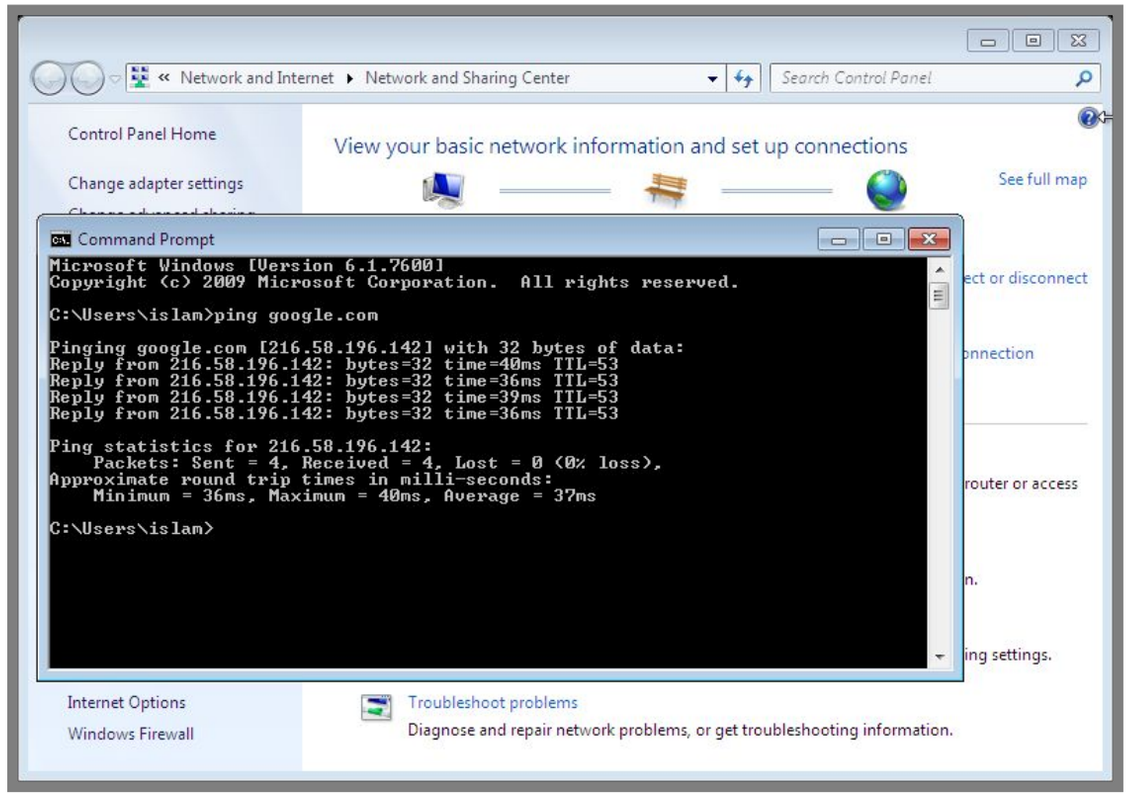
Gambar 14.44 Menjalankan konfigurasi firewall nat yang telah dilakukan

Selanjutnya kita harus melakukan konfigurasi ip address, gateway, dan dns resolver pada komputer client



Gambar 14.45 Konfigurasi ip address pada client

Saat ini seharusnya komputer client telah bisa melakukan pingg ke google.com

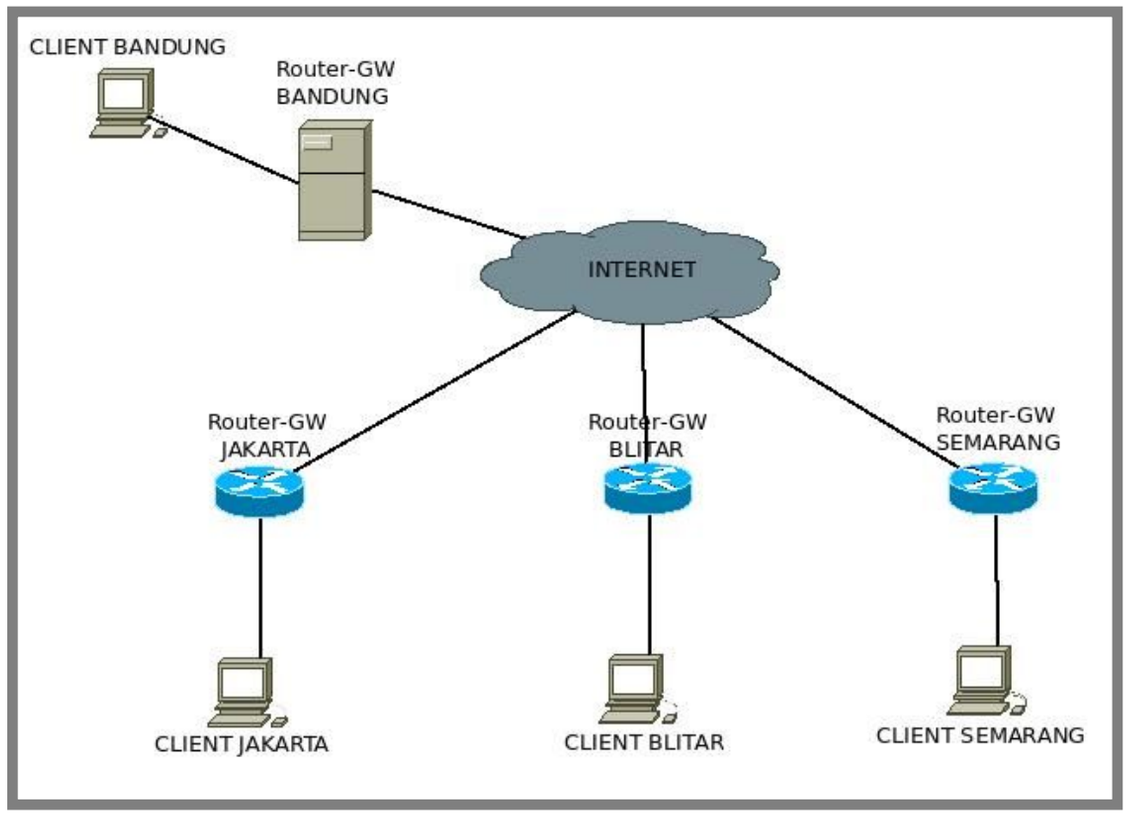


Gambar 14.46 Pengujian dari komputer client

---END OF CHAPTER---

Bab 15
Virtual Private Network Server

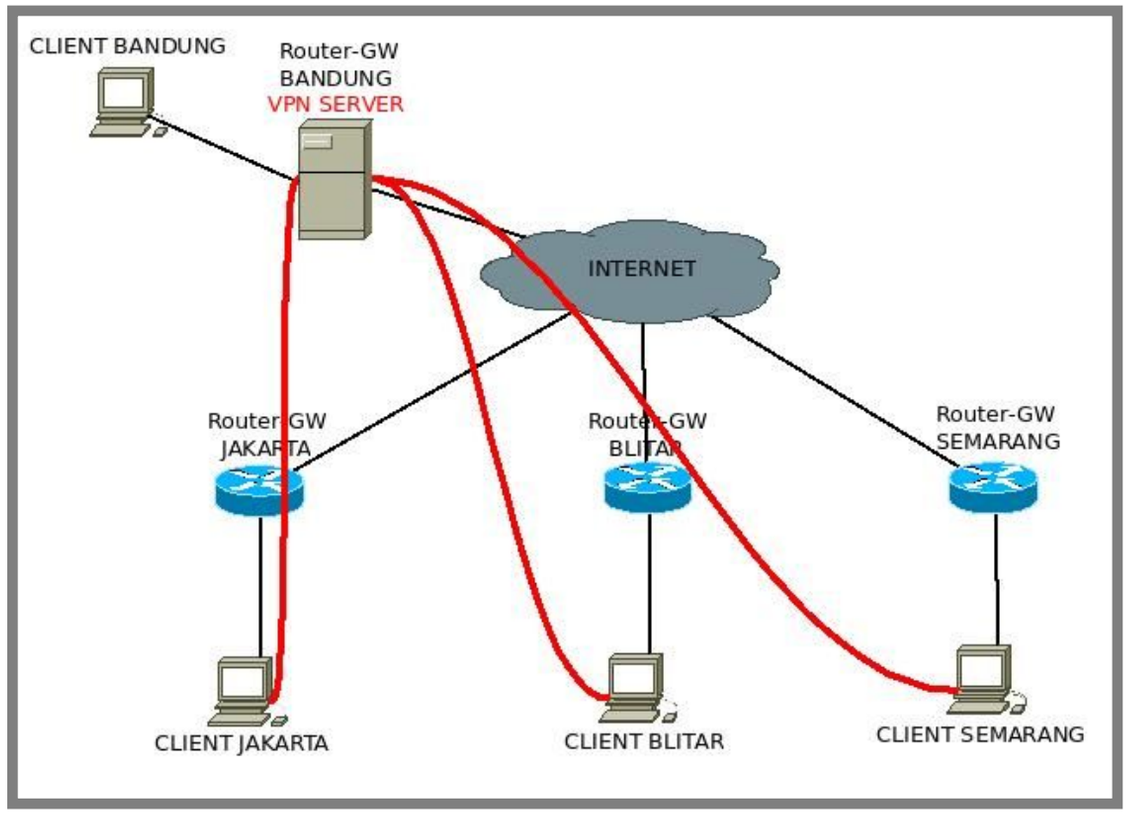
Virtual Private Network (VPN) adalah sebuah protocol yang memungkinkan kita untuk membuat sebuah jaringan lokal dibawah jaringan internet. Namun tentu saja jaringan local yang dibuat hanyalah bersifat virtual (tidak nyata). Perhatikan ilustrasi berikut



Gambar 15.1 Contoh jaringan tanpa vpn

Perhatikan gambar diatas, terlihat bahwa komputer client yang berada di Bandung, Jakarta, Blitar, dan Semarang berada di jaringan lokal. Sehingga keempat komputer client tersebut tidak akan bisa saling berkomunikasi.

Terus bagaimana jika kita diharuskan untuk menghubungkan Keempat komputer client tersebut?? Perhatikan ilustrasi berikut



Gambar 15.2 Contoh penerapan vpn

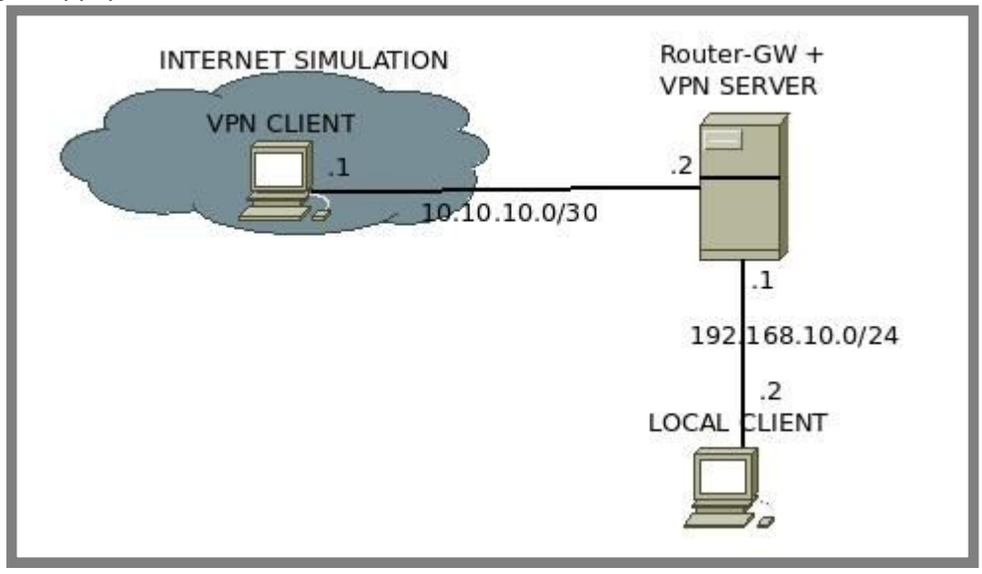
Dengan memanfaatkan VPN Server, komputer client yang ada di Jakarta, Blitar, dan Semarang akan membuat jalan pintas (tunnel) menuju VPN Server tanpa melalu Router di masing-masing kota. Sedangkan client yang ada di Bandung tidak perlu membuat tunnel ke VPN Server, hal ini dikarenakan client tersebut sudah connect secara langsung ke VPN Server.

Nantinya ketiga komputer client yang ada di Jakarta, Blitar, dan Semarang akan mendapat ip address dari VPN Server dan bisa saling berkomunikasi layaknya berada di jaringan lokal. Sedangkan untuk client yang berada di Bandung, dia tidak akan mendapat ip address dari VPN Server, dia hanya akan menggunakan ip address dari LAN, namun client yang ada di Bandung ini juga bisa berkomunikasi dengan client-client yang sudah terkoneksi dengann VPN Server layaknya jaringan lokal.

Biasanya VPN Server dibangun pada sebuah komputer yang memiliki ip public, sehingga VPN Server tersebut dapat diakses dari manapun asalkan terhubung dengan internet. Hal ini dikarenakan kita tidak akan bisa connect dengan VPN Server jika kita tidak bisa berkomunikasi dengan VPN Server.

Konfigurasi VPN Server dengan PPTP

Terdapat beberapa aplikasi yang dapat kita gunakan untuk membuat VPN Server, salah satu yang terkenal dan sangat mudah untuk dikonfigurasi adalah pptp. Berikut topologi jaringan yang akan kita gunakan untuk membuat vpn server dengann pptp

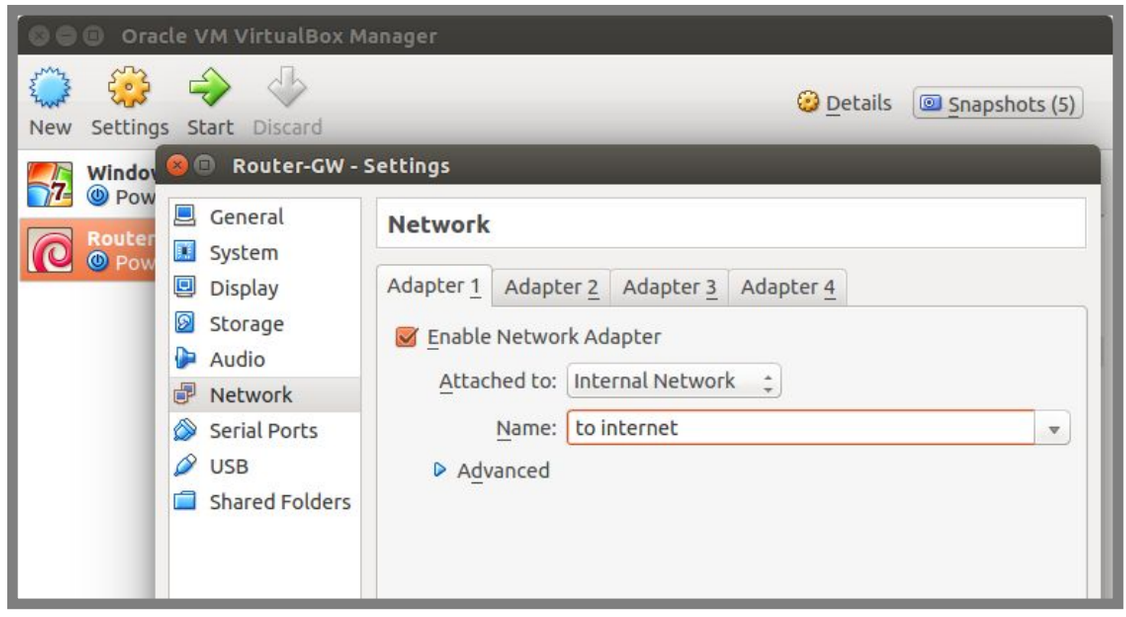


Gambar 15.3 Toplogi jaringan untuk praktik vpn server

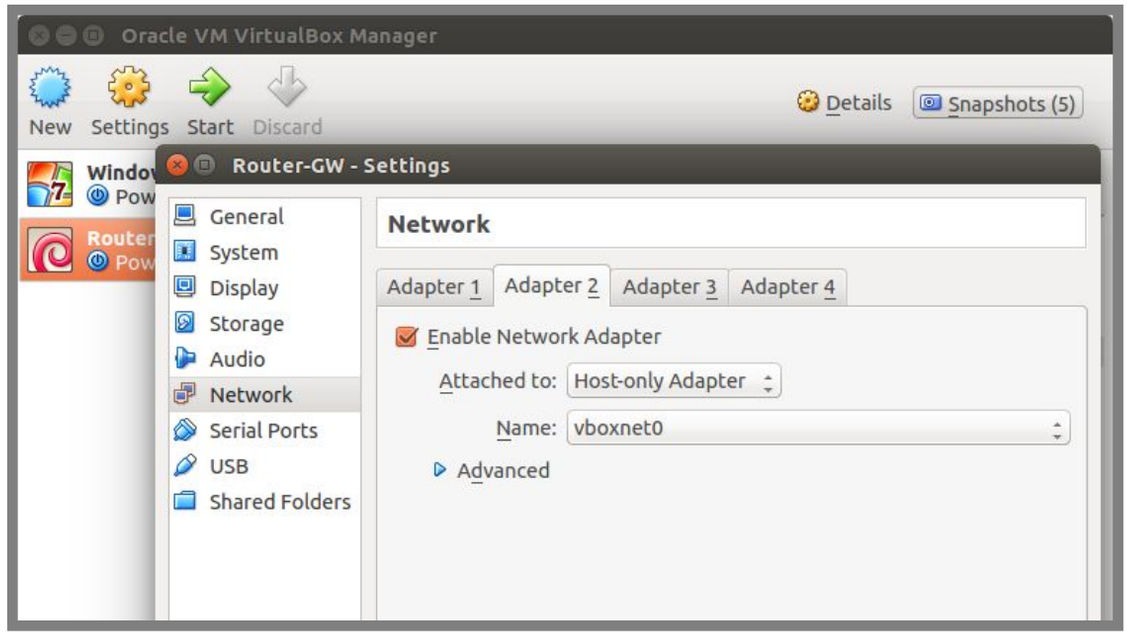
Tujuan kita adalah bagaimana agar vpn client yang berada di internet (internet simmulation) nantinya bisa berkomunikasi dengan local client.

Kita akan menggunakan guest os windows untuk vpn client, begitu juga untuk vpn server kita juga akan menggunakan guest os debian. Sedangkan untuk local client kita akan menggunakan host os ubuntu.

Berikut konfigurasi network adapter pada vpn server

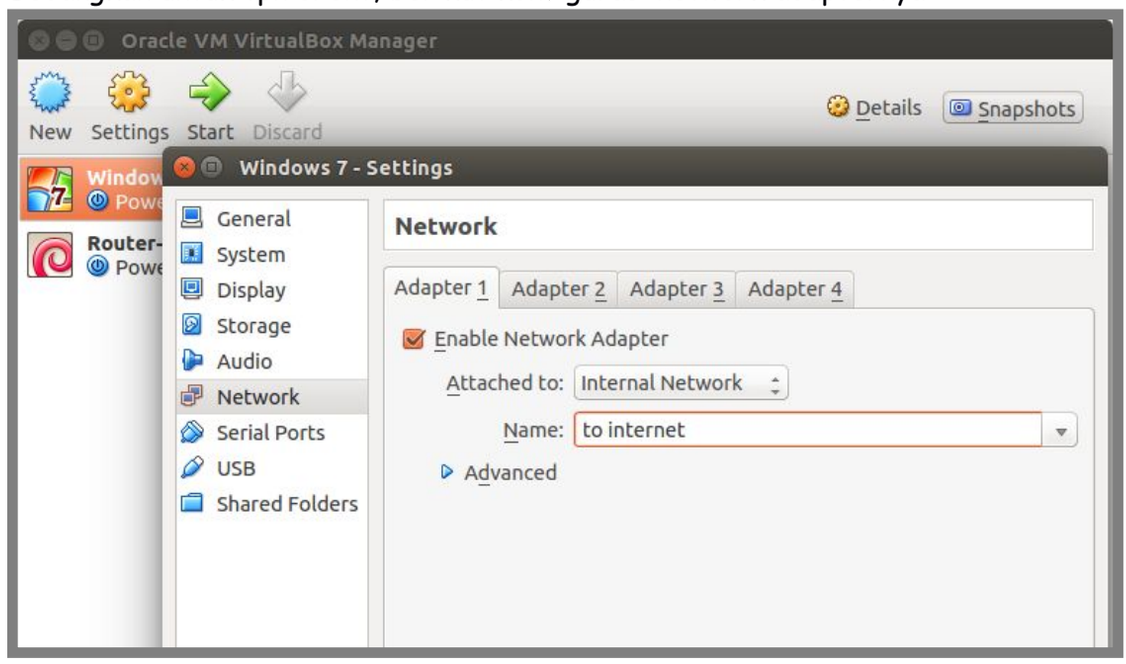


Gambar 15.4 Konfigurasi network adapter pada vpn server



Gambar 15.5 Konfigurasi network adapter pada vpn server

Sedangkan untuk vpn client, berikut konfigurasi network adapternya



Gambar 15.6 Konfigurasi network adapter pada vpn client

Pada parktik ini, kita asumsikan pada vpn server telah dikonfigurasi ip address sesuai topologi dan sudah diaktifkan fungsi routing. Selanjutnya kita akan fokus installasi dan konfigurasi vpn server dengan pptp. Berikut perintah yang dapat kita gunakan untuk menginstall aplikasi pptp

```
root@forkits:~# apt-get install pptpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bcrelay libpcap0.8 ppp
The following NEW packages will be installed:
  bcrelay libpcap0.8 ppp pptpd
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/630 kB of archives.
After this operation, 1456 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 15.7 Installasi aplikasi pptp

Selanjutnya kita harus melakukan konfigurasi pada pptp. Berikut konfigurasi yang perlu dilakukan

```
root@forkits:~# nano /etc/pptpd.conf
.....
.....
.....
.....
# (Recommended)
localip 100.100.100.1
remoteip 100.100.100.2-10,100.100.100.20
# or
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
```

Gambar 15.8 Konfigurasi vpn server dengan pptp

Perhatikan gambar diatas, terlihat bahwa kita melakukan perubahan pada baris nomor 4 dan 5 dari bawah. Parameter local ip menunjukkan ip address yang natinya akan digunakan vpn server, sedangkan remoteip menunjukkan rentang ip address yang nantinya akan diberikan kepada vpn client.

Selanjutnya lakukan konfigurasi berikut

```
root@forkits:~# nano /etc/ppp/pptpd-options
.....
ms-dns 192.168.10.1
#ms-dns 10.0.0.2
.....
```

Gambar 15.9 Konfigurasi vpn server dengan pptp

Konfigurasi diatas menunjukkan ip address yang nantinya akan digunakan sebagai dns resolver oleh vpn client. Langkah terahir kita harus membuat username dan password yang nantinya akan digunakan oleh vpn client untuk *connect* ke vpn server

```
root@forkits:~# nano /etc/ppp/pptpd-options
# Secrets for authentication using CHAP
# client        server      secret                  IP addresses
user1           pptpd       pass-user1              *
user2           pptpd       pass-user2              100.100.100.20
```

Gambar 15.10 Konfigurasi username dan password untuk vpn server

Perhatikan gambar diatas, terlihat kita membuat dua user. User pertama dengan nama *user1* dan password *pass-user1* dan dengan ip address dynamic (artinya akan dipilihkan secara acak dari range ip address yang kita konfigurasikan pada pptpd.conf tadi).

User kedua dengan nama *user2* dengan password *pass-user2* dan dengan ip address static, yaitu 100.100.100.20. Sehingga jika vpn client connect ke vpn server dengan user *user2,* maka ip addressnya pasti 100.100.100.20. Selanjutnya restart service pptp

```
root@forkits:~# service pptpd restart
Restarting PPTP:
Stopping PPTP: pptpd.
Starting PPTP Daemon: pptpd.
root@forkits:~#
```

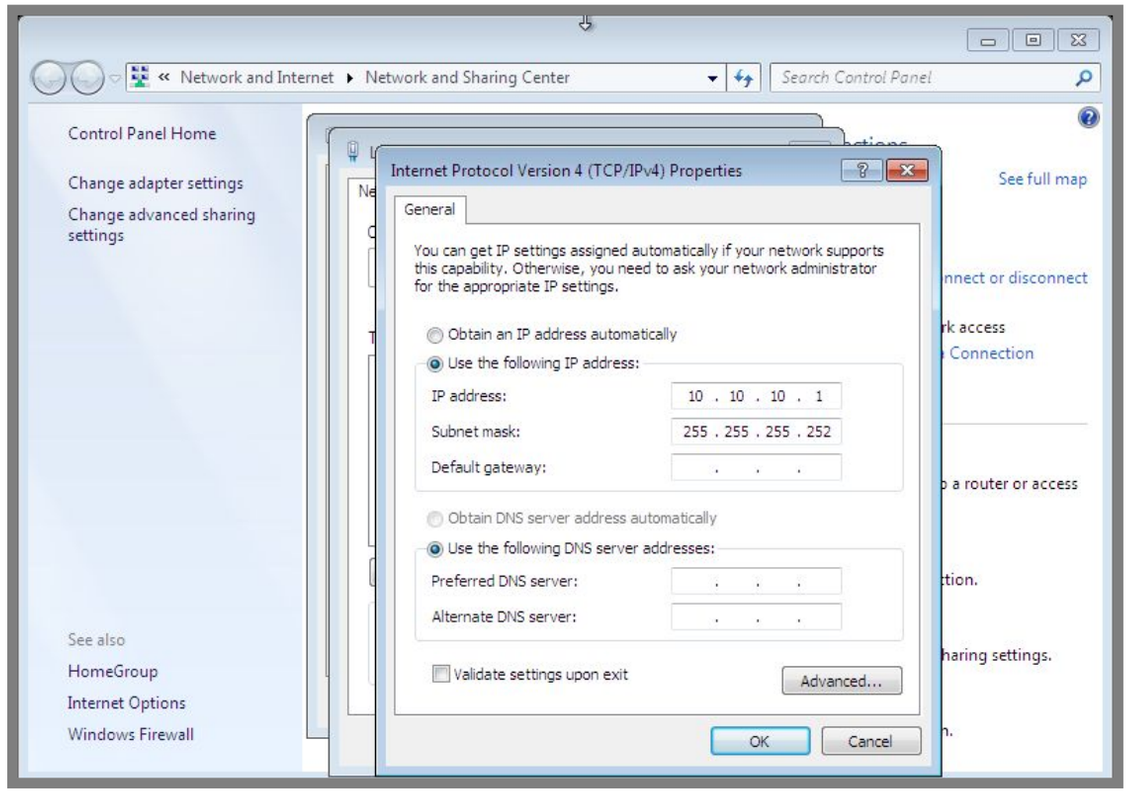
Gambar 15.11 Restart service pptp

Sampai saat ini kita telah selesai mengkonfigurasi VPN Server. Selanjutnya kita harus melakukan konfigurasi ip address pada client, baik pada vpn client maupun pada local client. Berikut konfigurasi ip address pada local client

```
admin@ubuntu:~$ sudo ifconfig vboxnet0 192.168.10.2/24
[sudo] password for admin: (tak terlihat)
admin@ubuntu:~$ sudo route add default gw 192.168.10.1
admin@ubuntu:~$
```

Gambar 15.12 Konfigurasi ip address pada local client

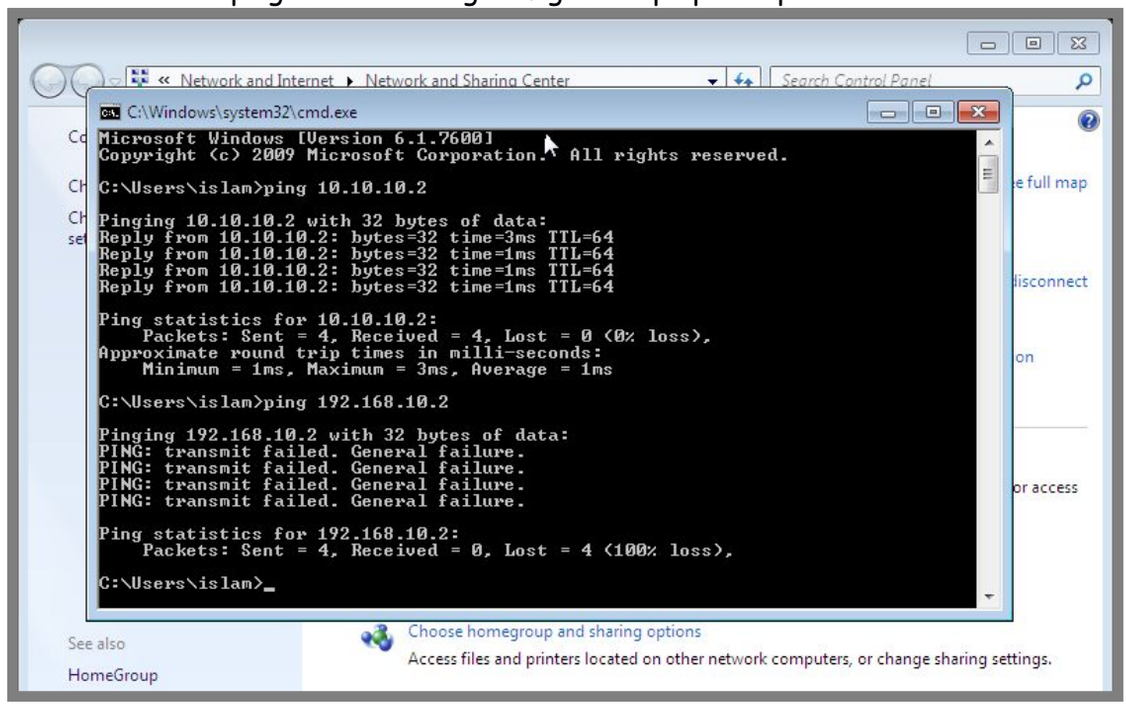
Sedangkan konfigurasi ip address pada vpn client adalah sebagai berikut



Gambar 15.13 Konfigurasi ip address pada vpn client

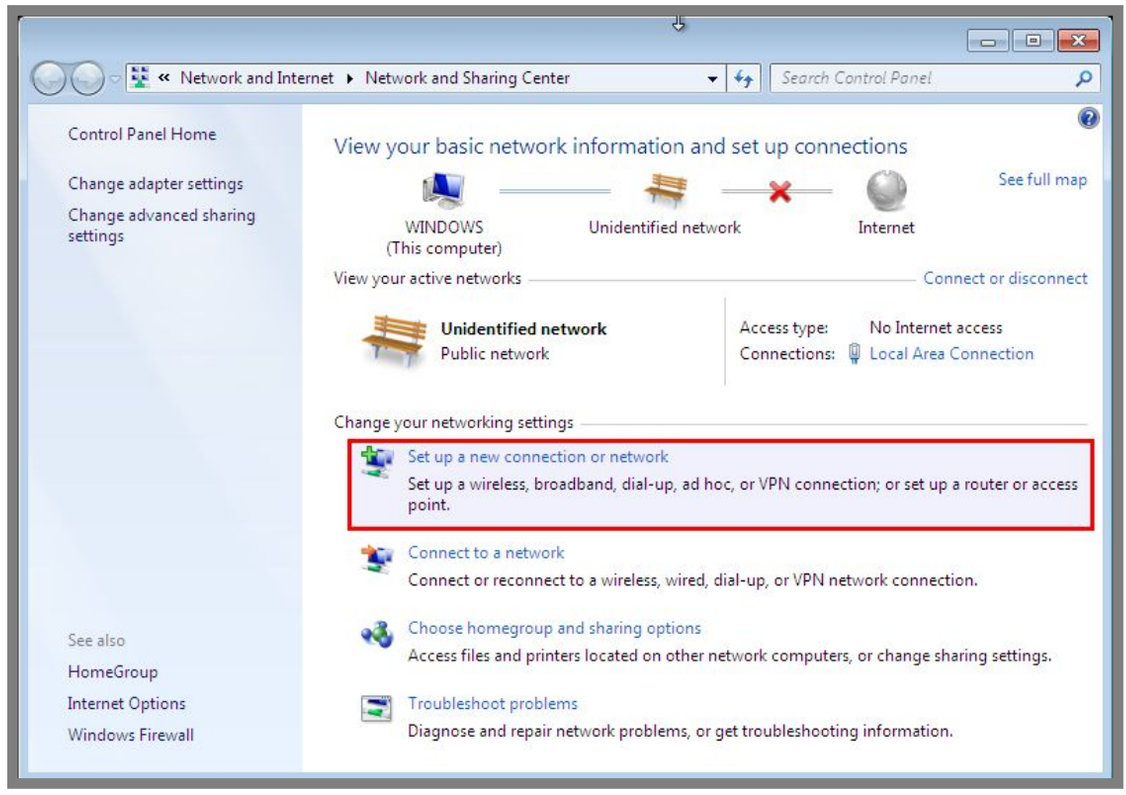
Perhatikan bahwa kita tidak boleh mengkonfigurasi gateway pada vpn client. Karena jika kita mengkonfigurasi gateway, maka otomatis vpn client bisa berkomunikasi dengan local client tanpa vpn.

Perhatikan hasil ping sebelum mengkonfigurasi vpn pada vpn client



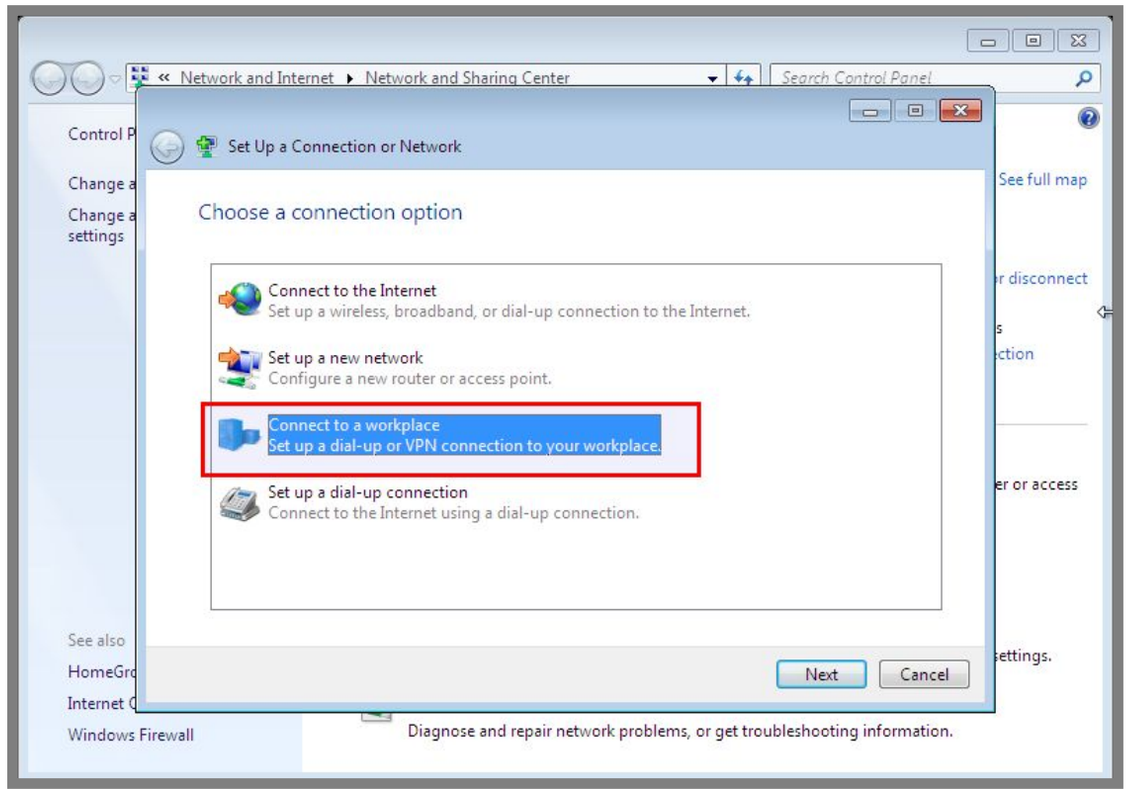
Gambar 15.14 Pengujian ping ke local client

Perhatikan bahwa vpn client sudah bisa ping ke vpn server namun belum bisa ping ke local client. Karena itu kita harus mengkonfigurasi vpn pada vpn client, berikut langkah konfigurasi yang perlu dilakukan pada vpn client



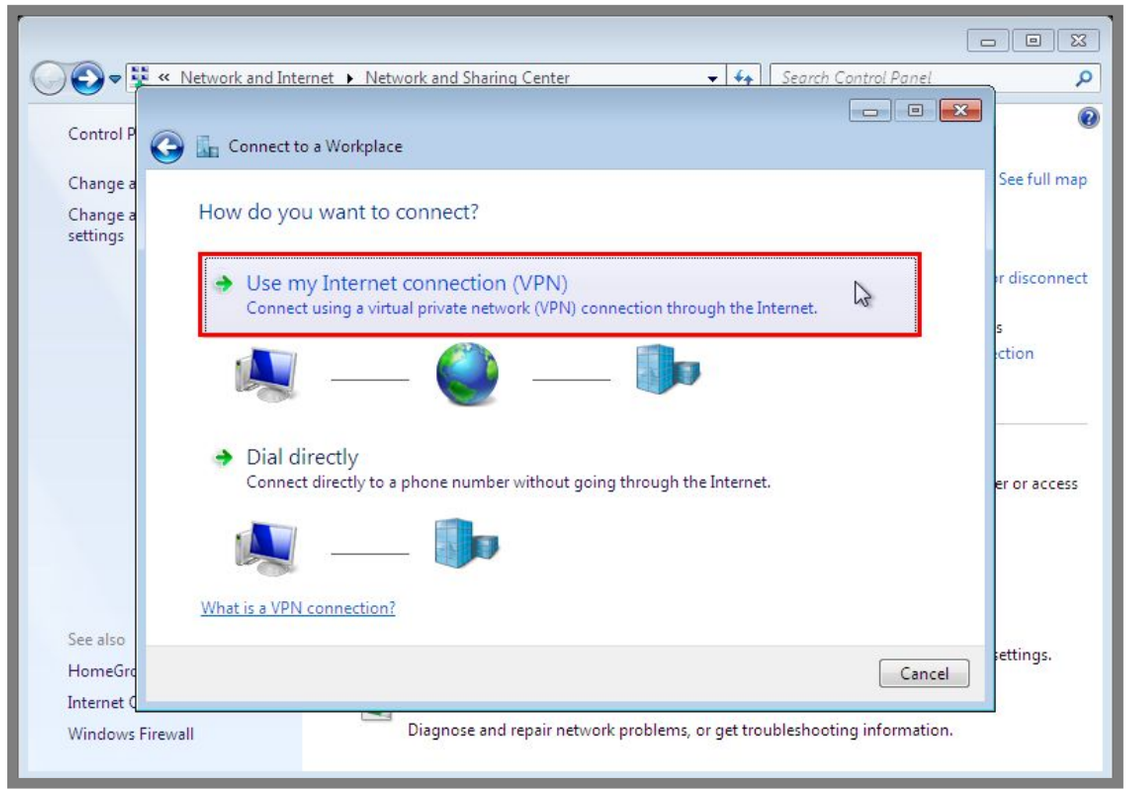
Gambar 15.15 Konfigurasi vpn client

Pilih *Connect to a workplace*



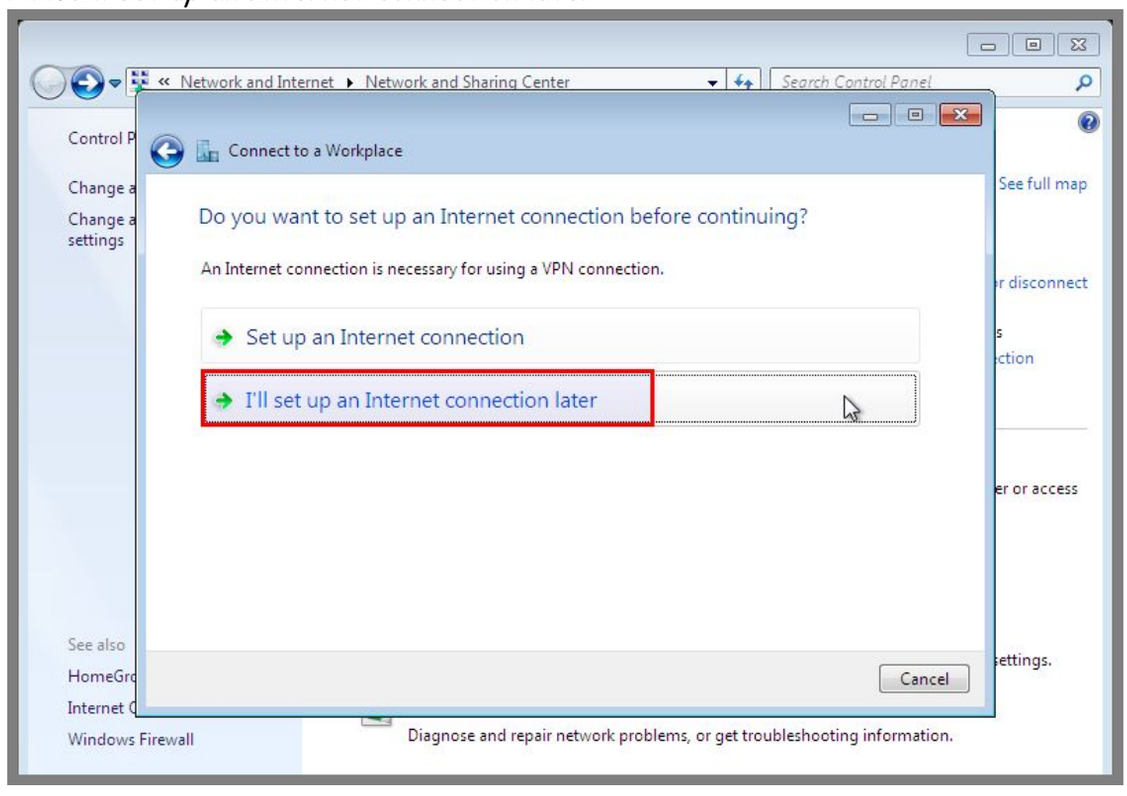
Gambar 15.16 Konfigurasi vpn client

Pilih *Use my Internet Connection (VPN)*



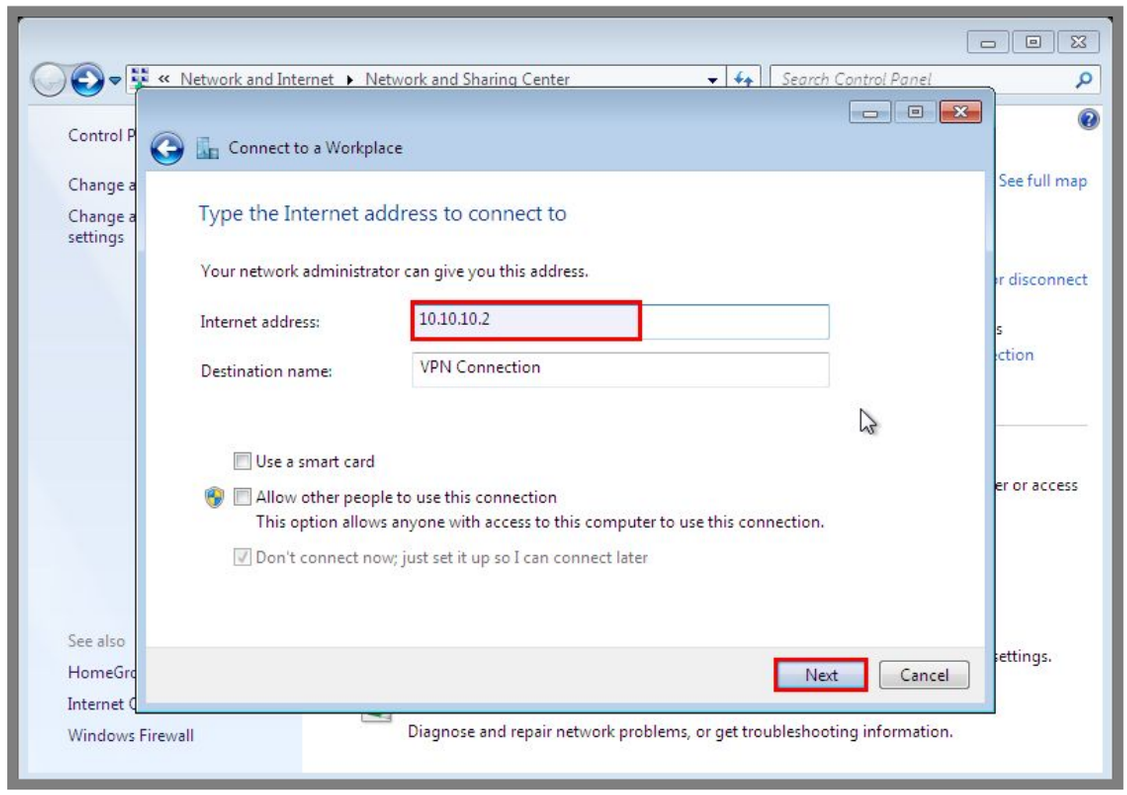
Gambar 15.17 Konfigurasi vpn client

Pilih *I'll set up an Internet connection later*



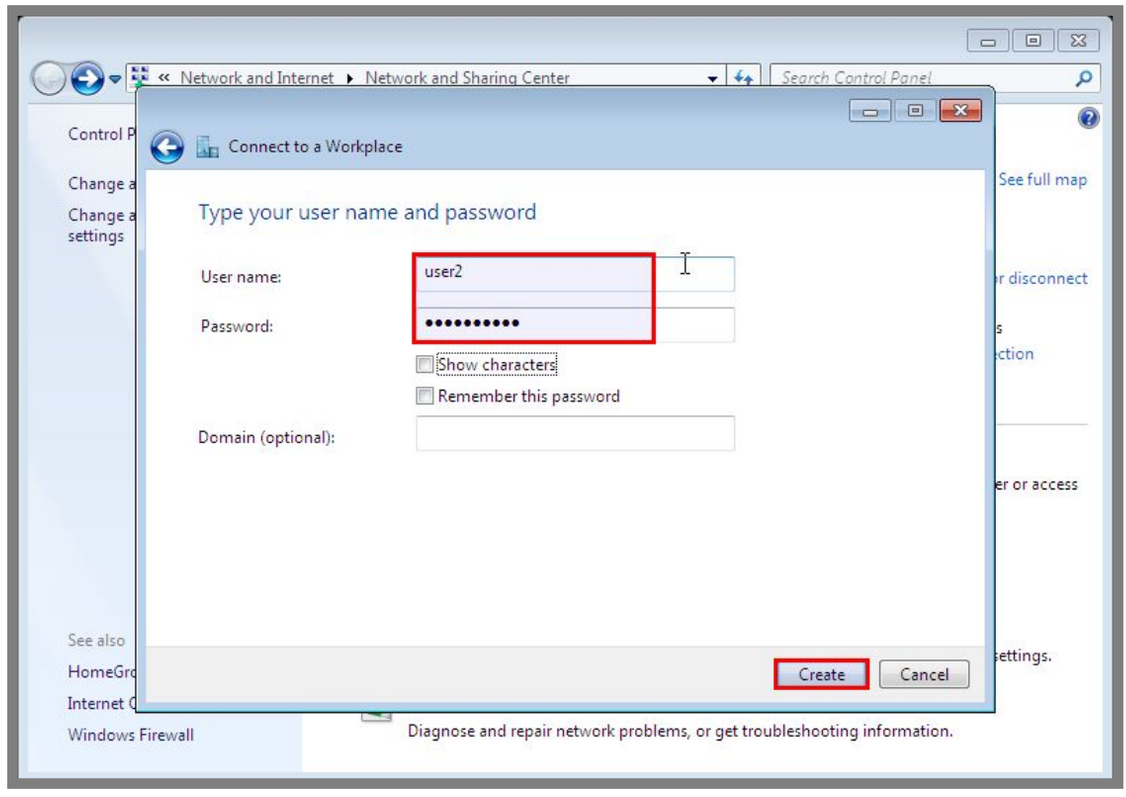
Gambar 15.18 Konfigurasi vpn client

Masukkan ip address dari vpn server (ip address vpn server harus bisa diping oleh vpn client)



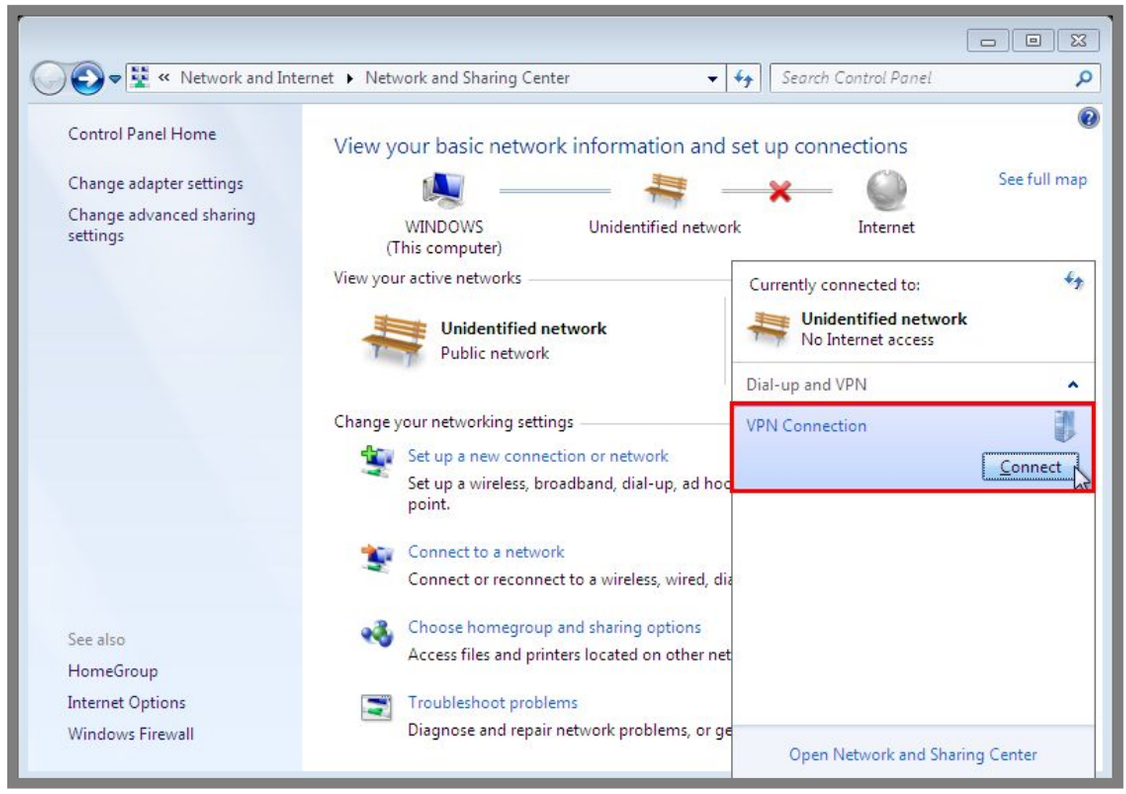
Gambar 15.19 Konfigurasi vpn client

Masukkan username dan password yang ingin digunakan saat akan connect ke vpn server



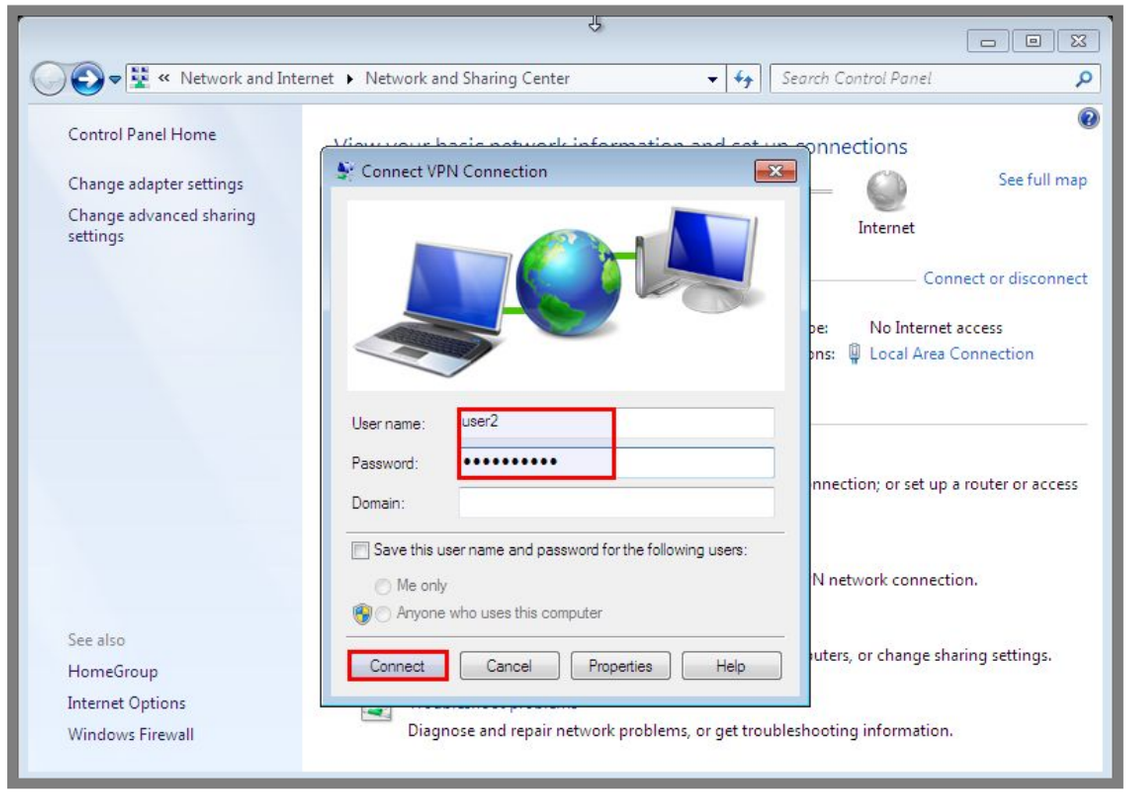
Gambar 15.20 Konfigurasi vpn client

Setelah selesai setup, silahkan connect ke vpn yang baru saja kita setup tadi

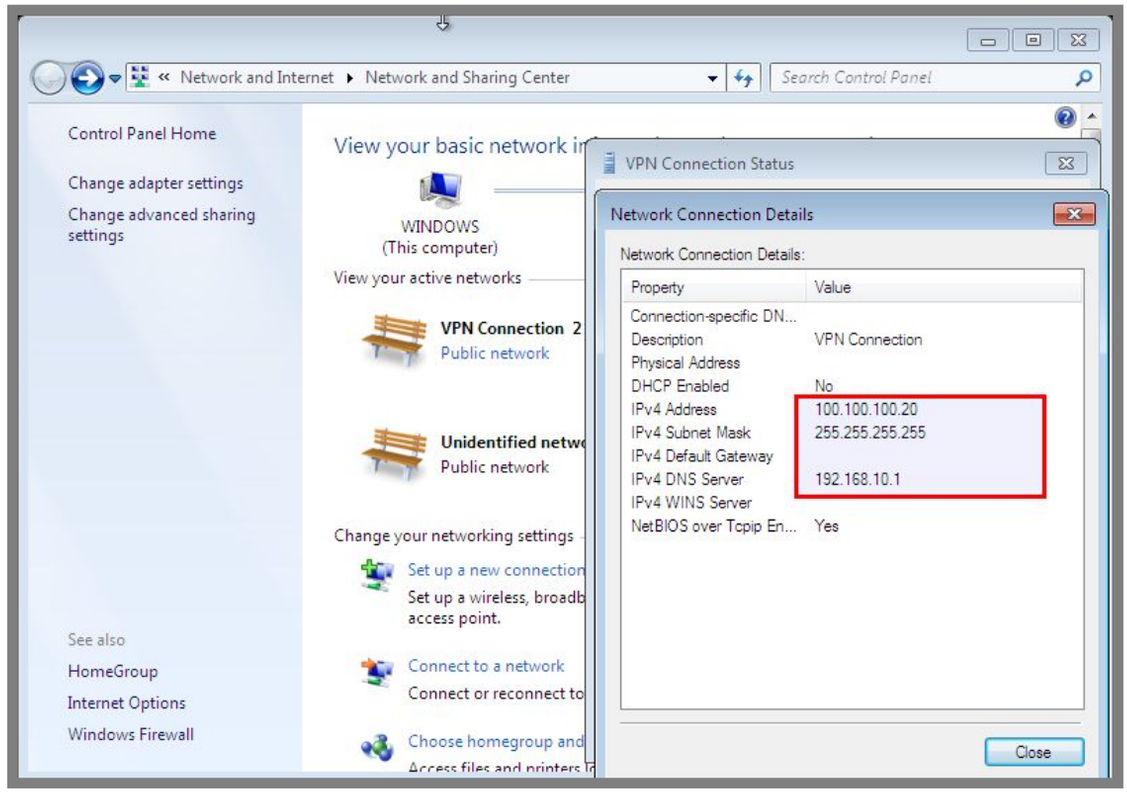


Gambar 15.21 Connect ke vpn server

Masukkan username dan password yang ingin digunakan untuk connect ke vpn server

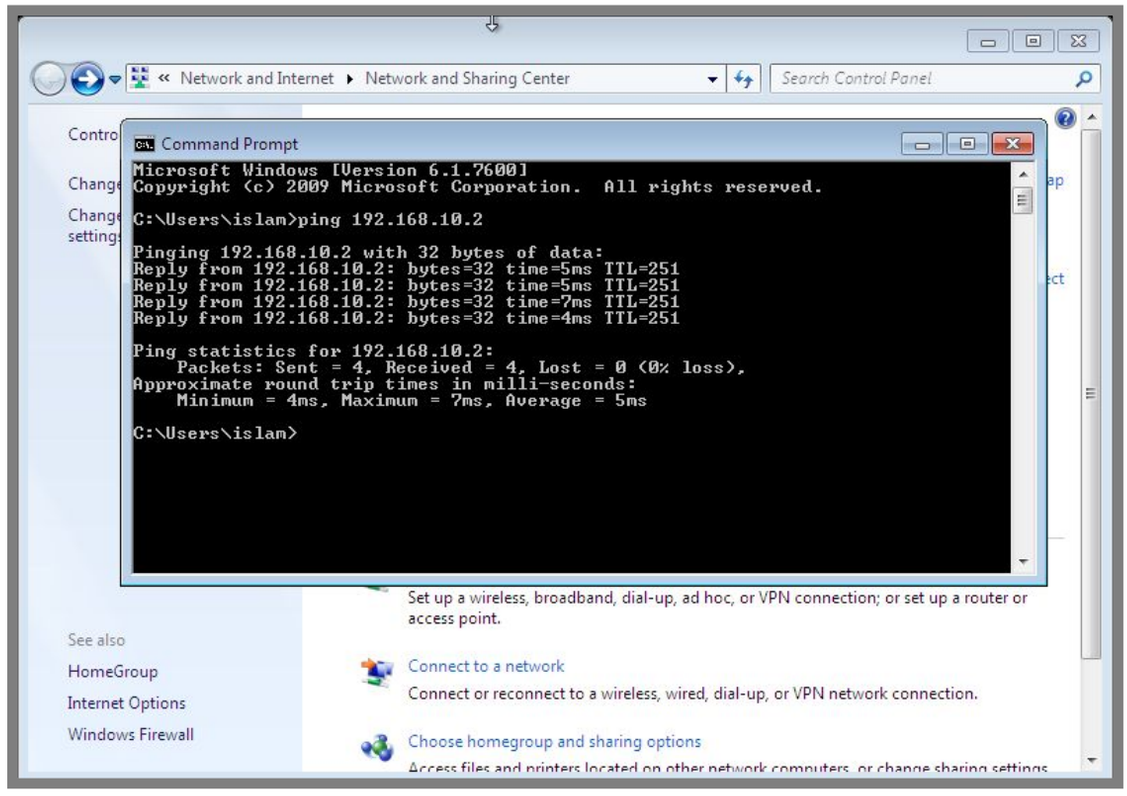


Gambar 15.22 Masukkan username dan password untuk connect ke vpn server



Gambar 15.23 Hasil koneksi dengan vpn server

Perhatikan gambar diatas, terlihat bahwa kita telah connect dengan vpn server dan mendapat ip address 100.100.100.20 dengan dns 192.168.10.1. Sampai saat ini seharusnya kita sudah bisa ping ke local client



Gambar 15.24 Pengujian ping dari vpn client ke local client

Konfigurasi VPN Server dengan OpenVPN

Sebenarnya tidak ada perbedaan yang jauh antara PPTP dan OpenVPN jika dilihat dari segi fungsinya, yaitu sama-sama aplikasi yang dapat kita gunakan untuk membangun sebuah VPN Server.

Perbedaannya terletak pada keamanan dan kesulitan konfigurasi. OpenVPN cenderung lebih aman, dan tentunya juga lebih sulit dikonfigurasi (ingat prinsip ini "***Keamanan selalu berbanding terbalik dengan kenyamanan***").

OpenVPN dinyatakan lebih aman karena authentikasi yang dilakukan saat client dan server membuat sebuah koneksi vpn adalah berbasis certificate dan key. Sehingga client harus mempunyai file certificate yang cocok dengan key yang dimiliki server jika ingin terkoneksi ke server.

Hal ini tentu akan membuat koneksi vpn sangat aman, namun sekali lagi "Keamanan selalu berbanding terbalik dengan Kenyamanan". Selain aman, client juga harus melakukan konfigurasi yang sedikit rumit jika dibanding menggunakan PPTP.

Untuk praktik pada sub bab ini, kita akan menggunakan topologi jaringan yang sama dengan yang kita gunakan pada sub bab sebelumnya. Perhatikan gambar 15.3. Seluruh skenario pada sub bab ini juga sama persis dengan sub bab sebelumnya, mulai dari tujuan topologi, yaitu menghubungkan vpn client dengan local client. Begitu juga dengan konfigurasi network adapter pada vpn server dan vpn client. Semuanya sama, kecuali konfigurasi pada vpn server.

Selanjutnya kita hanya akan fokus pada konfigurasi vpn server menggunakan openvpn. Diasumsikan vpn server telah dikonfigurasi ip address sesuai topologi dan telah diaktifkan fungsi routing. Selanjutnya untuk menginstall aplikasi openvpn, kita bisa menggunakan perintah sebagai berikut

```
root@forkits:~# apt-get install openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  liblzo2-2 libpkcs11-helper1
The following NEW packages will be installed:
  liblzo2-2 libpkcs11-helper1 openvpn
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/633 kB of archives.
After this operation, 1523 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 15.25 Installasi aplikasi openvpn

Secara default, didalam direktori openvpn (/etc/openvpn) tidak ada file-file konfigurasi yang dibutuhkan. Karena itu, kita harus mengcopy file-file konfigurasi yang sudah disediakan oleh opevpn di direktori /usr. Kita bisa menggunakan perintah sebagai berikut

```
root@forkits:~# cp -rf /usr/share/doc/openvpn/examples/easy-rsa/2.0/
/etc/openvpn/
root@forkits:~# cp
/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/
root@forkits:~#
```

Gambar 15.26 Copy file contoh konfigurasi openvpn ke direktori openvpn

Selanjutnya kita harus melakukan sedikit perubahan pada file-file konfigurasi openvpn.

```
root@forkits:~# cd /etc/openvpn/2.0/
root@forkits:/etc/openvpn/2.0# nano vars
.....
.....
.....
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="ID"
export KEY_PROVINCE="Jawa Timur"
export KEY_CITY="Blitar"
export KEY_ORG="ForKITS"
export KEY_EMAIL="admin@forkits.com"
export KEY_EMAIL=admin@forkits.com
#export KEY_CN=changeme
#export KEY_NAME=changeme
#export KEY_OU=changeme
#export PKCS11_MODULE_PATH=changeme
#export PKCS11_PIN=1234
```

Gambar 15.27 Konfigurasi pada *vars*

Perhatikan gambar diatas, terlihat bahwa kita melakukan perubahan pada bagian teks warna hijau (baris-baris ini berada di baris terahir file *vars*).

Telah dikatakan sebelumnya, bahwa OpenVPN akan menggunakan authentikasi berbasis certificate dan key. Oleh karena itu kita harus membuat file certificate dan key. Berikut perintah yang dapat kita gunakan

```
root@forkits:/etc/openvpn/2.0# source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on
/etc/openvpn/2.0/keys
root@forkits:/etc/openvpn/2.0# ./clean-all
root@forkits:/etc/openvpn/2.0# ./build-dh
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
+.....+.....+.....+.....+.....+.....+.....+.....
.+.....+....+.....+.....+.....+.....+..
...+.....+...+.....
.....+.....++*++*++*
root@forkits:/etc/openvpn/2.0# ./pkitool --initca
Using CA Common Name: ForKITS CA
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'ca.key'
root@forkits:/etc/openvpn/2.0# ./pkitool --server server
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'server.key'
.....
.....
.....
root@forkits:/etc/openvpn/2.0# ./pkitool client
Generating a 1024 bit RSA private key
..++++++
.....++++++
writing new private key to 'client.key'
.....
.....
.....
root@forkits:/etc/openvpn/2.0#
```

Gambar 15.28 Membuat certificate dan key

Selanjutnya copy file certificate dan key yang telah kita buat tadi ke direktori openvpn

```
root@forkits:/etc/openvpn/2.0# cp keys/ca.crt /etc/openvpn/
root@forkits:/etc/openvpn/2.0# cp keys/dh1024.pem /etc/openvpn/
root@forkits:/etc/openvpn/2.0# cp keys/server.key /etc/openvpn/
root@forkits:/etc/openvpn/2.0# cp keys/server.crt /etc/openvpn/
root@forkits:/etc/openvpn/2.0#
```

Gambar 15.29 Copy file certificate dan key ke direktori openvpn

Langkah selanjutnya kita harus membuat user untuk openvpn,

```
root@forkits:/etc/openvpn/2.0# useradd -m -s /bin/false uservpn
root@forkits:/etc/openvpn/2.0# passwd uservpn
Enter new UNIX password: (tak terlihat)
Retype new UNIX password:   (tak terlihat)
passwd: password updated successfully
root@forkits:/etc/openvpn/2.0#
```

Gambar 15.30 Menambahkan user untuk openvpn

Parameter *-m* pada perintah *useradd* dimaksudkan untuk membuat home direktori, sedangkan parameter *-s /bin/false* dimaksudkan agar user tersebut tidak bisa login ke local komputer. Selanjutnya kita harus mengcopy file-file certificate dan key yang dibutuhkan client ke home direktori user yang baru saja kita buat

```
root@forkits:/etc/openvpn/2.0# cp keys/ca.crt /home/uservpn/
root@forkits:/etc/openvpn/2.0# cp keys/client.key /home/uservpn/
root@forkits:/etc/openvpn/2.0# cp keys/client.crt /home/uservpn/
root@forkits:/etc/openvpn/2.0# chmod 755 /home/uservpn/ -R
```

Gambar 15.31 Copy file certificate dan key yang dibutuhkan client

Setelah selesai membuat file-file certificate dan key yang dibutuhkan, selanjutnya kita harus melakukan konfigurasi pada openvpn. Berikut langkah-langkah konfigurasi yang perlu dilakukan

```
root@forkits:/etc/openvpn/2.0# cd ..
root@forkits:/etc/openvpn# gunzip server.conf.gz
root@forkits:/etc/openvpn# nano server.conf
.....
server 100.100.100.0 255.255.255.0      >> cari dengan kata kunci 10.8

Push "route 192.168.10.0 255.255.255.0"   >> cari dengan kata kunci
                                          192.168.10.0 (isikan dengan network
                                       yang dimiliki local client)

push "redirect-gateway def1 bypass-dhcp"    >> cari dengan kata kunci def1

push "dhcp-option DNS 192.168.10.1" >> ada dibawah push "redirect-gateway.."
;push "dhcp-option DNS 208.67.220.220"

client-to-client   >> ada dibawah push "dhcp-option....."

duplicate-cn   >> ada dibawah client-to-client
.....
.....
```

Gambar 15.32 Konfigurasi openvpn

Selanjutnya restart service openvpn

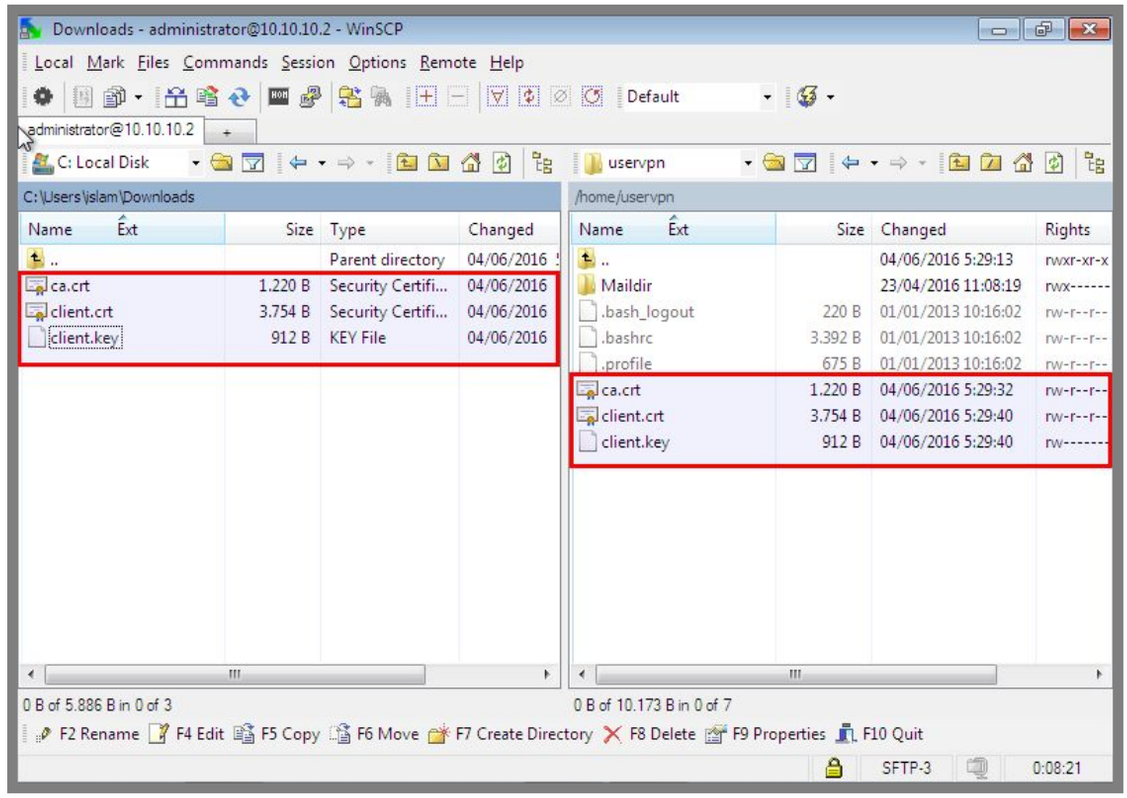
```
root@forkits:/etc/openvpn# service openvpn restart
[ ok ] Stopping virtual private network daemon:.
[ ok ] Starting virtual private network daemon: server.
root@forkits:/etc/openvpn#
```

Gambar 15.33 Restart service openvpn

Sampai saat ini kita telah selesai melakukan konfigurasi pada komputer server. Selanjutnya kita hanya perlu melakukan konfigurasi pada client untuk connect ke openvpn server.

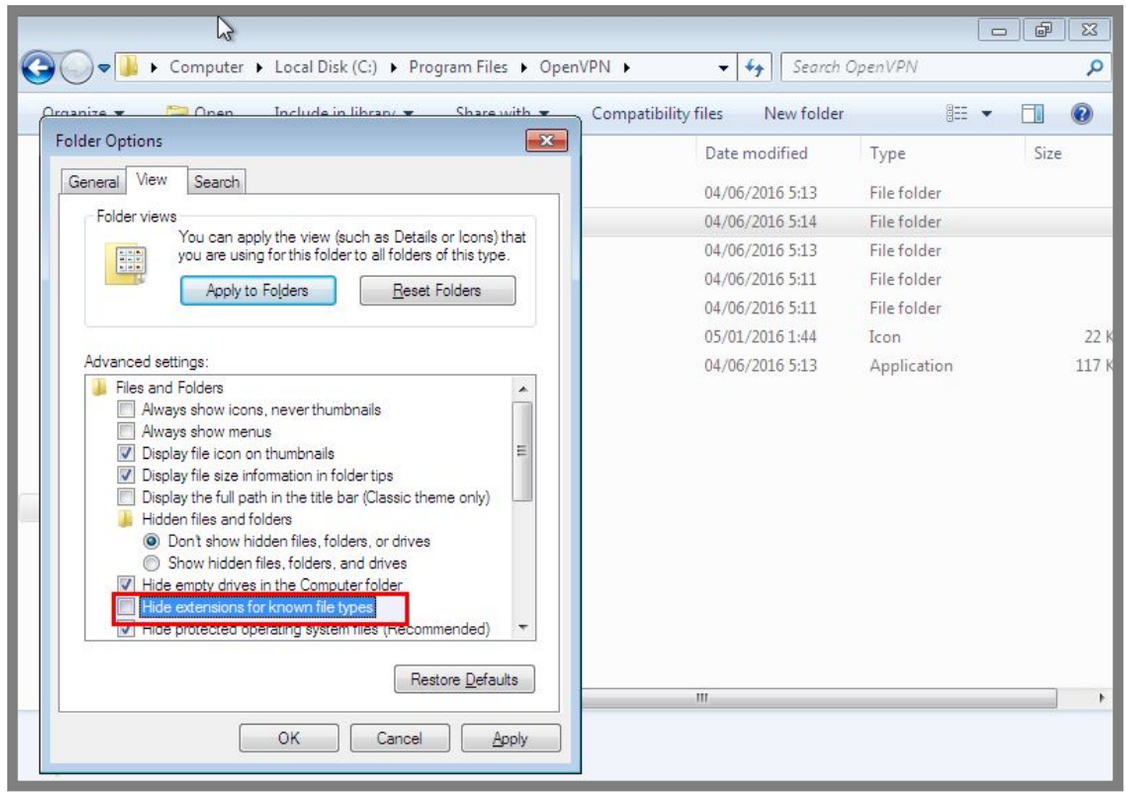
Ada beberapa aplikasi yang dibutuhkan oleh client untuk connect ke openvpn server. Yaitu winscp (untuk download file certificate dan key dari server) dan openvpn client (untuk connect ke openvpn server). Kita bisa download kedua aplikasi tersebut dari internet secara gratis. Diasumsikan kita telah memiliki kedua aplikasi tersebut

Diasumsikan kedua aplikasi tersebut telah terinstall dengan baik di komputer client, sehingga kita akan fokus pada konfigurasi vpn client menggunakan openvpn. Pertama yang harus kita lakukan adalah download file certificate dan key yang dibutuhkan oleh client dari komputer server, kita bisa memanfaatkann aplikasi winscp



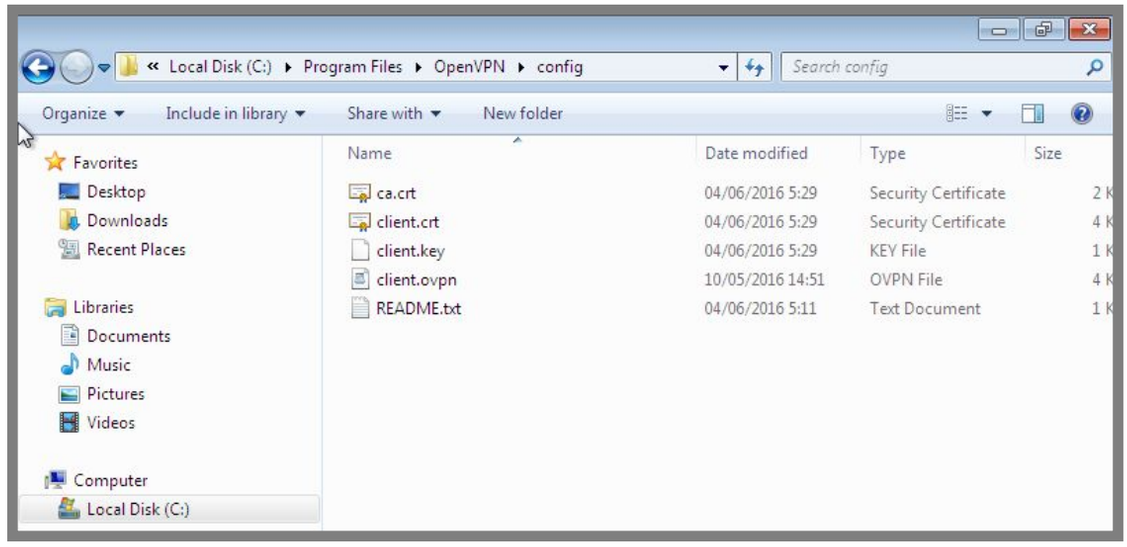
Gambar 15.34 Download file certificate yang dibutuhkan client

Selanjutnya buka file explorer, kemudian klik *Organize >> Folder and search options >> view* kemudian hilangkan centang pada *Hide extensions for known file types*



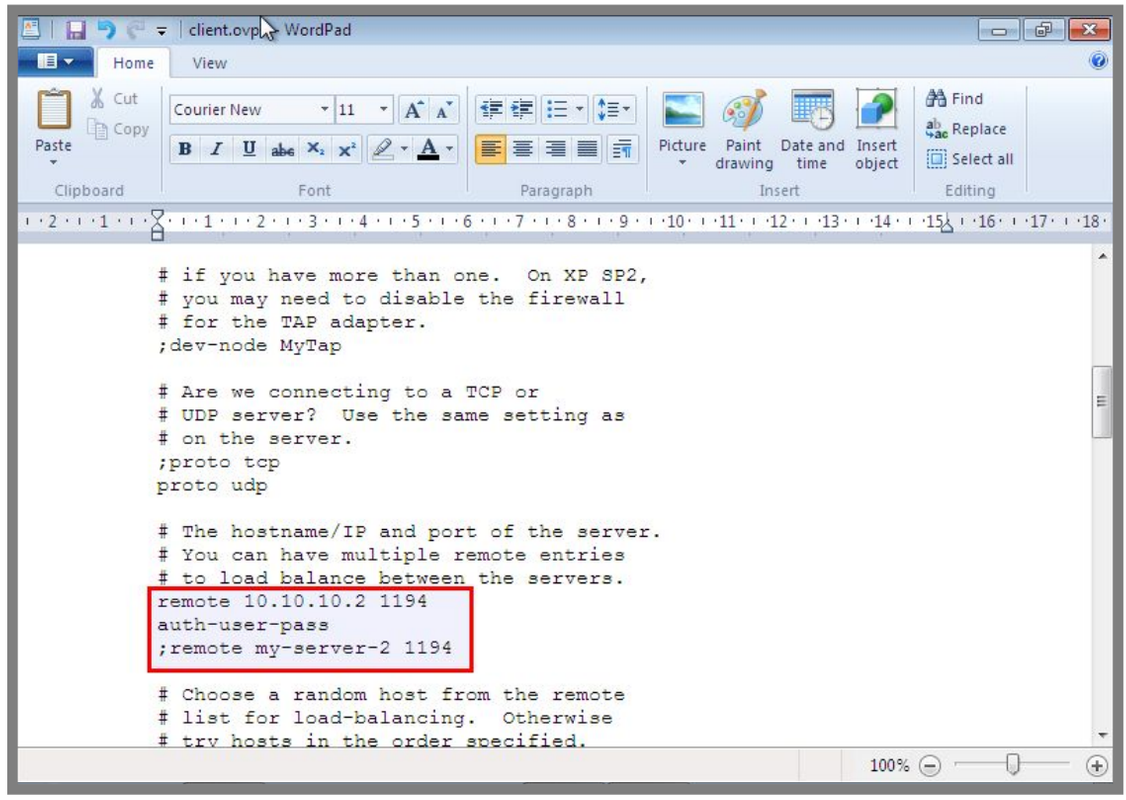
Gambar 15.35 Konfigurasi pada file explorer

Selanjutnya pindahkan file-file yang telah kita download tadi ke *C:\Program Files\OpenVPN\config.* Copykan juga file *client.ovpn* dari *C:\Program Files\OpenVPN\sample-cnfig* ke folder tersebut. Seharusnya isi dari folder tersebut terlihat sebagai berikut



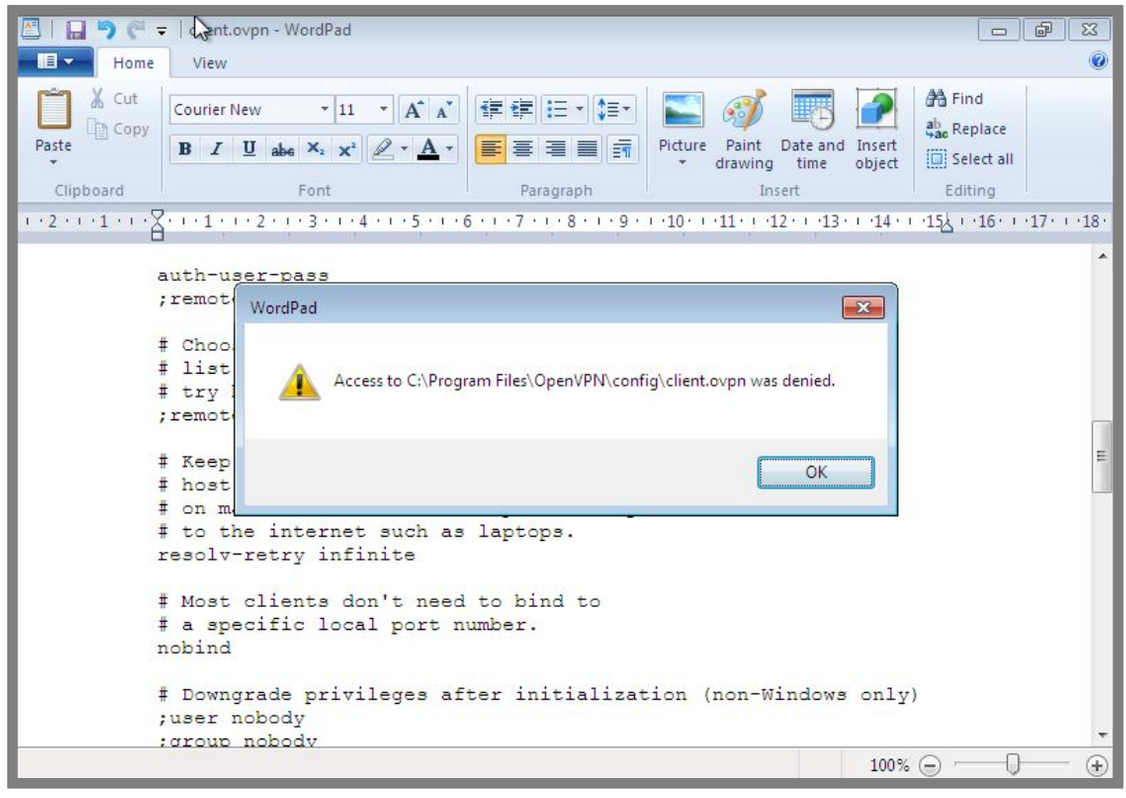
Gambar 15.36 memindahkan file-file yang diperlukan client

Edit file client.ovpn diatas pada bagian ini



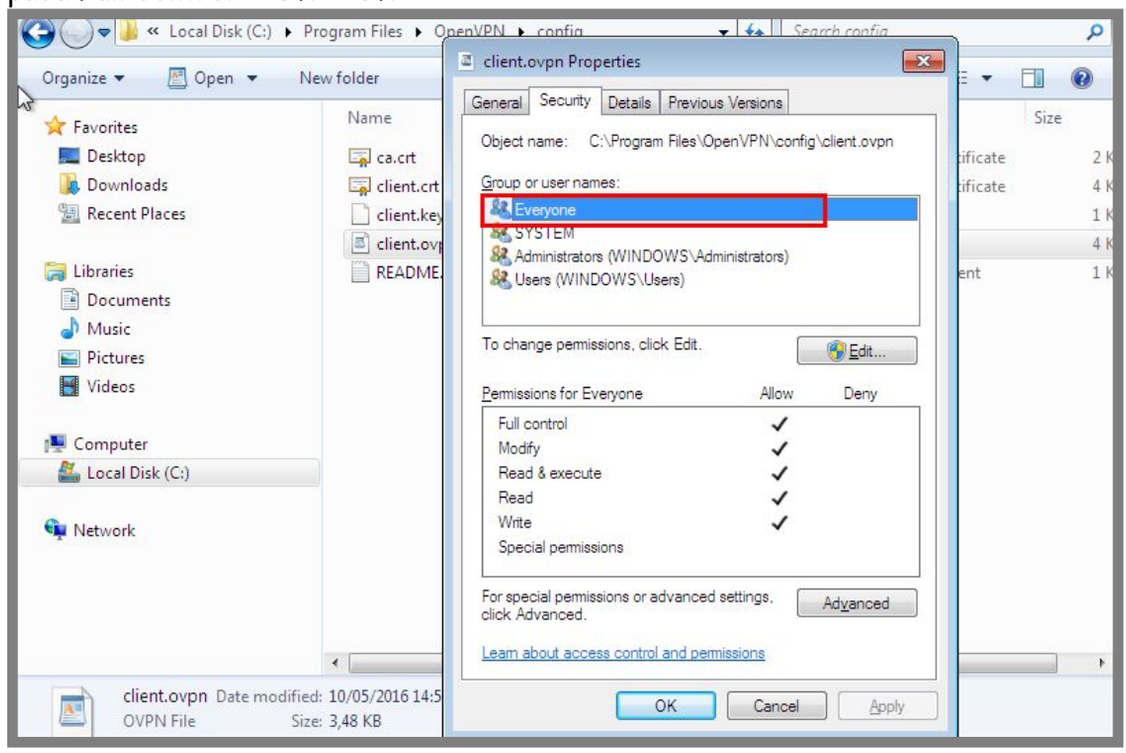
Gambar 15.37 Konfigurasi openvpn client

Save perubahan tersebut, namun mungkin akan ada peringatan sebagai berikut



Gambar 15.38 Error saat konfigurasi

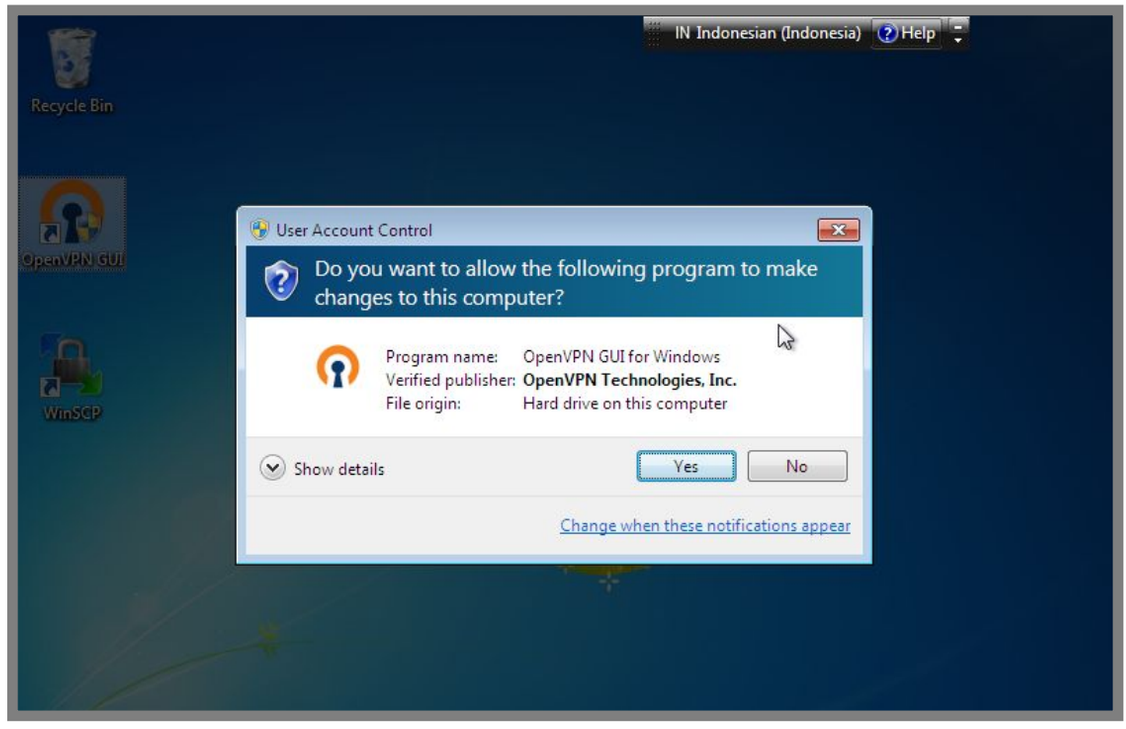
Jika ada peringatan seperti diatas, kilk kanan pada file client.ovpn, *Security >> Edit >> Add >> Advanced >> Find Now >>* Double Click *Everyone >> OK >>* Centang pada *Full Control >> OK >> OK*



Gambar 15.39 Mengatasi error yang terjadi

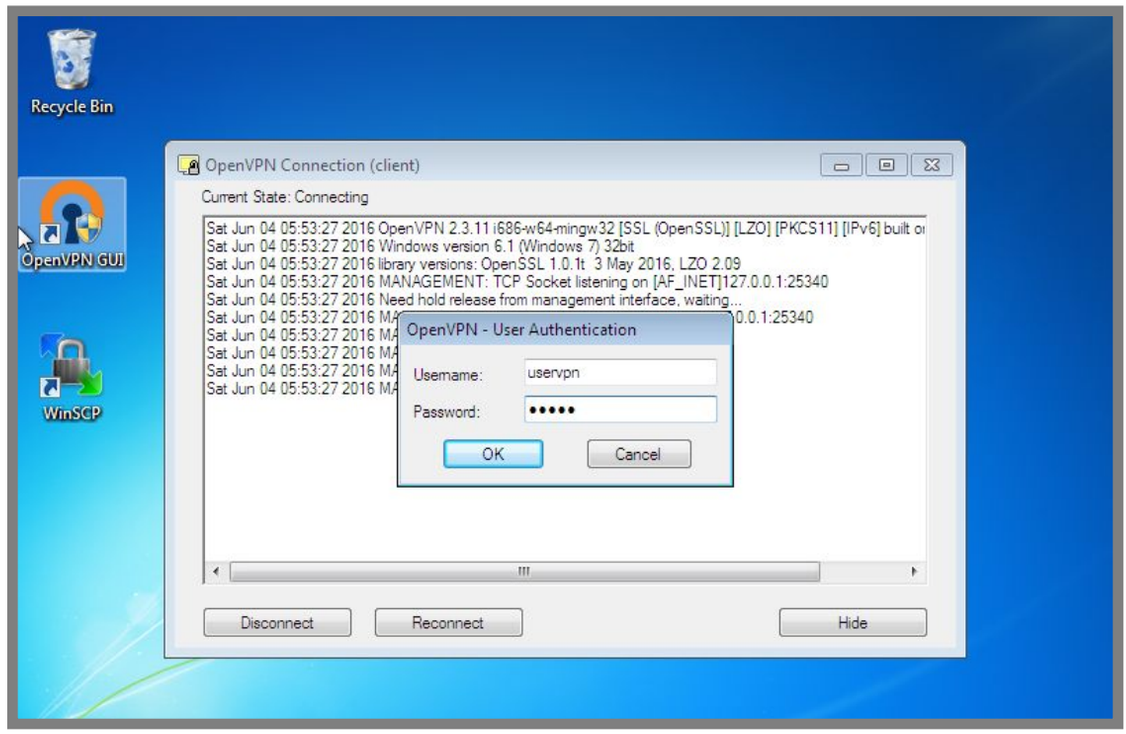
Setelah melakukann langkah diatas, seharusnya tidak akan ada masalah lagi saat kita mencoba untuk menyimpan perubahan yang kita lakukan pada file client.ovpn.

Sampai saat ini kita sudah selesai melakukan konfigurasi pada client, selanjutnya kita bisa menjalankan aplikasi openvpn client untuk connect ke openvpn server.



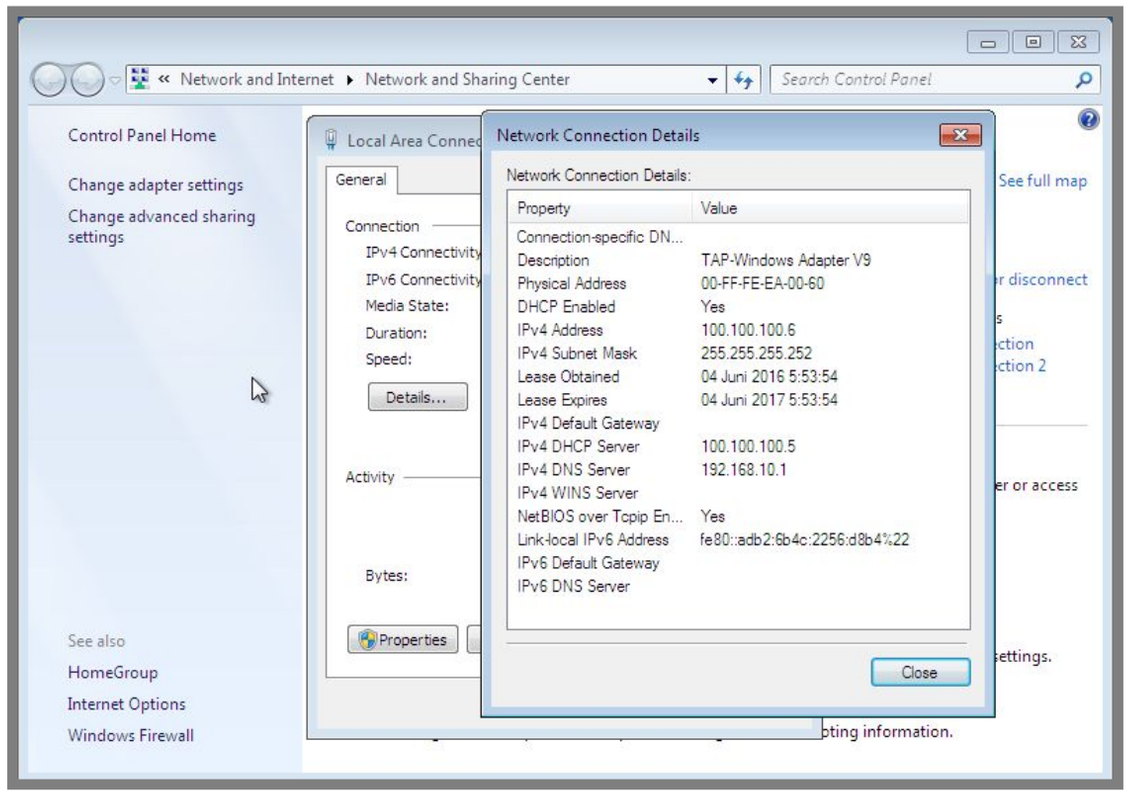
Gambar 15.40 Menjalankan openvpn client

Klik kanan pada icon openvpn yang ada ditry icon kemudian klik connect, selanjutnya kita akan diminta untuk memasukkan username dan password seperti berikut



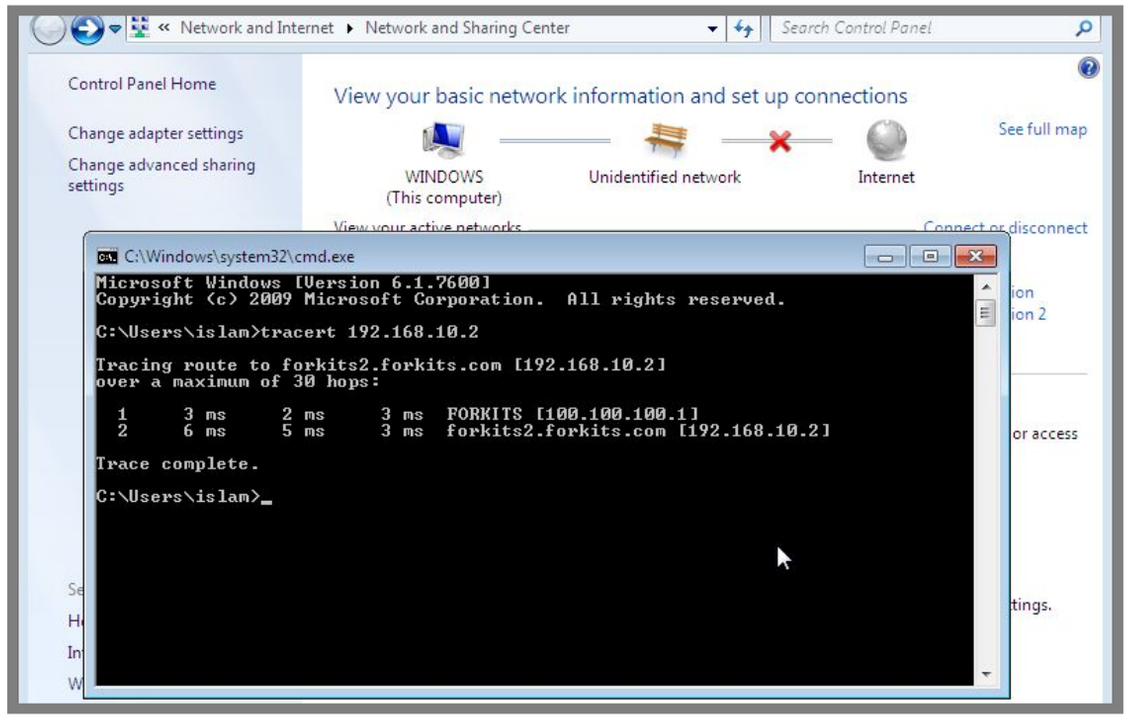
Gambar 15.41 Masukkan username dan password untuk connect openvpn

Perhatikan gambar berikut, terlihat bahwa kita telah mendapat ip address dari openvpn server



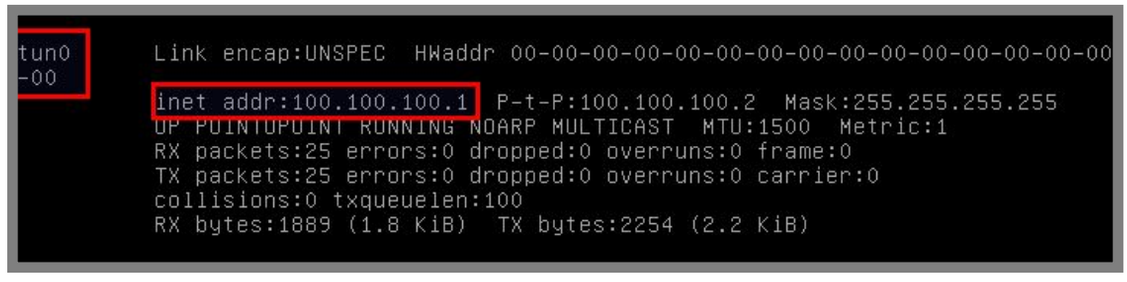
Gambar 15.42 Berhasil connect ke openvpn

Sampai saat ini seharusnya kita sudah bisa menghubungi local client dari vpn client.



Gambar 15.43 Traceroute dari vpn client ke local client

Perhatikan hasil traceroute diatas, terlihat bahwa jika vpn client ingin menuju local client, vpn client harus melewati ip address 100.100.100.1 (ip address openvpn). Vpn clien tidak menggunakan ip address milik ethernet dari vpn server (10.10.10.2), melainkan menggunakan ip address yang digunakan oleh openvpn (yaitu 100.100.100.1). Perhatikan hasil perintah ifconfig pada komputer server berikut



Gambar 15.44 IP Address openvpn pada server

Terlihat bahwa ip address 100.100.100.1 adalah milik interface tun0-00, interface ini adalah interface yang digunakan oleh openvpn.

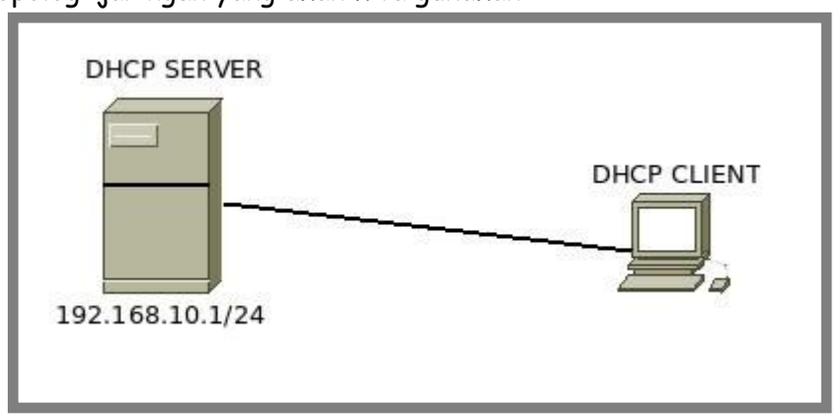
---END OF CHAPTER---

Bab 16
Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) adalah sebuah protocol yang berfungsi untuk memberikan ip address secara dynamic (otomatis) kepada client. Sehingga client yang terkoneksi ke server nantinya tidak perlu mensetting ip address secara manual, karena client akan otomatis mendapat ip address dari server. DHCP Server tidak hanya memberikan ip address, namun juga subnetmask, gateway, dns resolver, dan beberapa paramter lainnya.

Konfigurasi DHCP Server

Telah dijelaskan sebelumnya mengenai fungsi dari dhcp server. Pada sub bab ini kita akan belajar bagaimana mengkonfigurasi dhcp server pada server debian. Berikut topologi jaringan yang akan kita gunakan



Gambar 16.1 Topologi jaringan untuk praktik dhcp server

Kita akan menggunakan guest os debian sebagai dhcp server dan guest os windows sebagai dhcp client. Karena kita ingin menghubungkan dua guest os, tentunya type network adapter yang harus kita gunakan adalah internal network. Saya tidak akan menunjukkan lagi bagaimana konfigurasi network adapter yang perlu dilakukan.

Diasumsikan bahwa dhcp server sudah dikonfigurasi ip address sesuai topologi. Selanjutnya kita akan fokus konfigurasi dhcp server. Hal pertama yang harus kita

lakukan adalah menginstall aplikasi yang dibutuhkan untuk membuat dhcp server dii debian, yaitu isc-dhcp-server. Berikut perintah yang dapat kita gunakan untuk menginstall aplikasi tersebut

```
root@forkits:~# apt-get install isc-dhcp-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  isc-dhcp-server
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/936 kB of archives.
After this operation, 2225 kB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
  isc-dhcp-server
Install these packages without verification [y/N]? y
```

Gambar 16.2 Installasi aplikasi untuk dhcp server

Saat proses installasi, akan muncul pesan error seperti berikut, namun kita tidak perlu mempermasalahkannya,, abaikan saja!

```
[FAIL] Starting ISC DHCP server: dhcpd[....] check syslog for diagnostics. ... f
ailed!
 failed!
invoke-rc.d: initscript isc-dhcp-server, action "start" failed.
root@forkits:~# _
```

Gambar 16.3 Error saat proses installasi

Selanjutnya kita bisa langsung melakukan konfigurasi dhcp server, lakukan langkah berikut

```
root@forkits:~# nano /etc/dhcp/dhcpd.conf
.....
.....
# A slightly different configuration for an internal subnet.
subnet 192.168.10.0 netmask 255.255.255.0 { >>network yg digunakan server
  range 192.168.10.2 192.168.10.254; >>range ip yg akan dibirikan ke client
  option domain-name-servers 192.168.10.1; >>dns resolver yg diberikan ke client
  option domain-name "forkits.com";
  option routers 192.168.10.1; >>gateway yg diberikan ke client
  option broadcast-address 192.168.10.255; >>ip brodcast dari network yg ada
  default-lease-time 600;
  max-lease-time 7200;
}
.....
.....
```

Gambar 16.4 Konfigurasi dhcp server

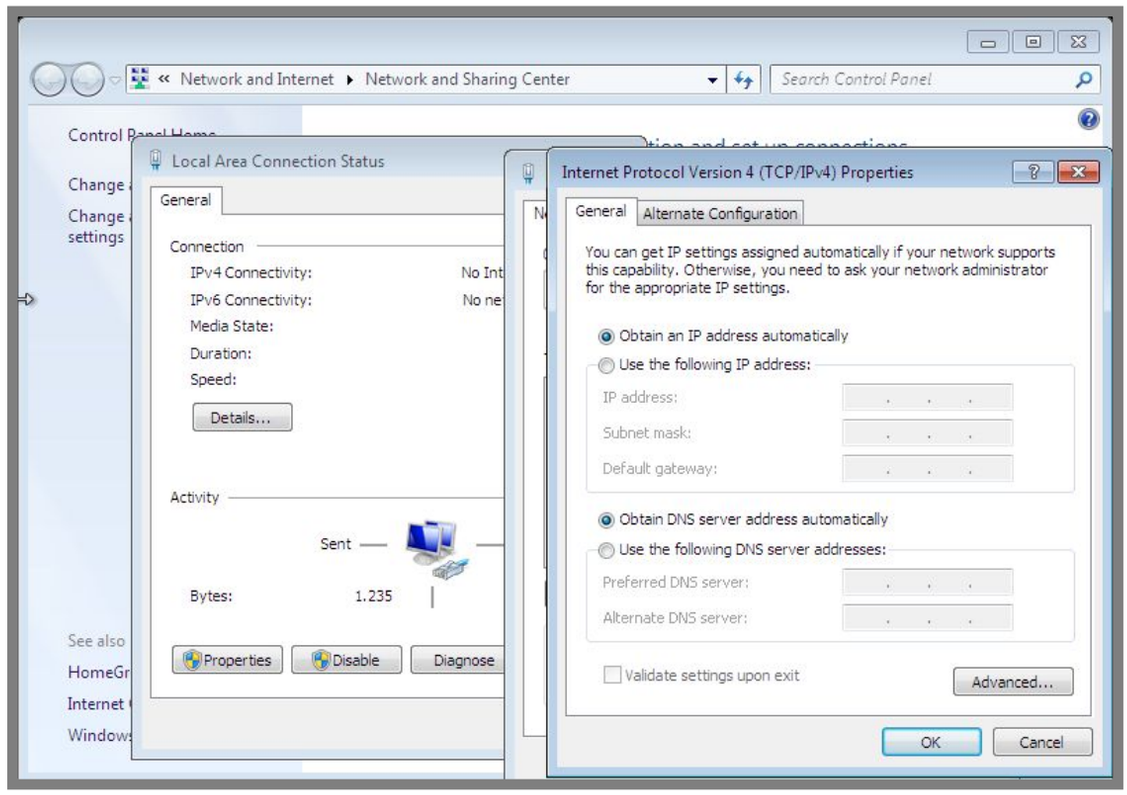
Perhatikan gambar diatas, kita melakukan perubahan pada bagian teks wana hijau. Lakukan perubahan sesuai dengan kondisi network yang ada di jaringan kita. Selanjutnya restart service dhcp

```
root@forkits:~# service isc-dhcp-server restart
[FAIL] Stopping ISC DHCP server: dhcpd failed!
[ ok ] Starting ISC DHCP server: dhcpd.
root@forkits:~#
```

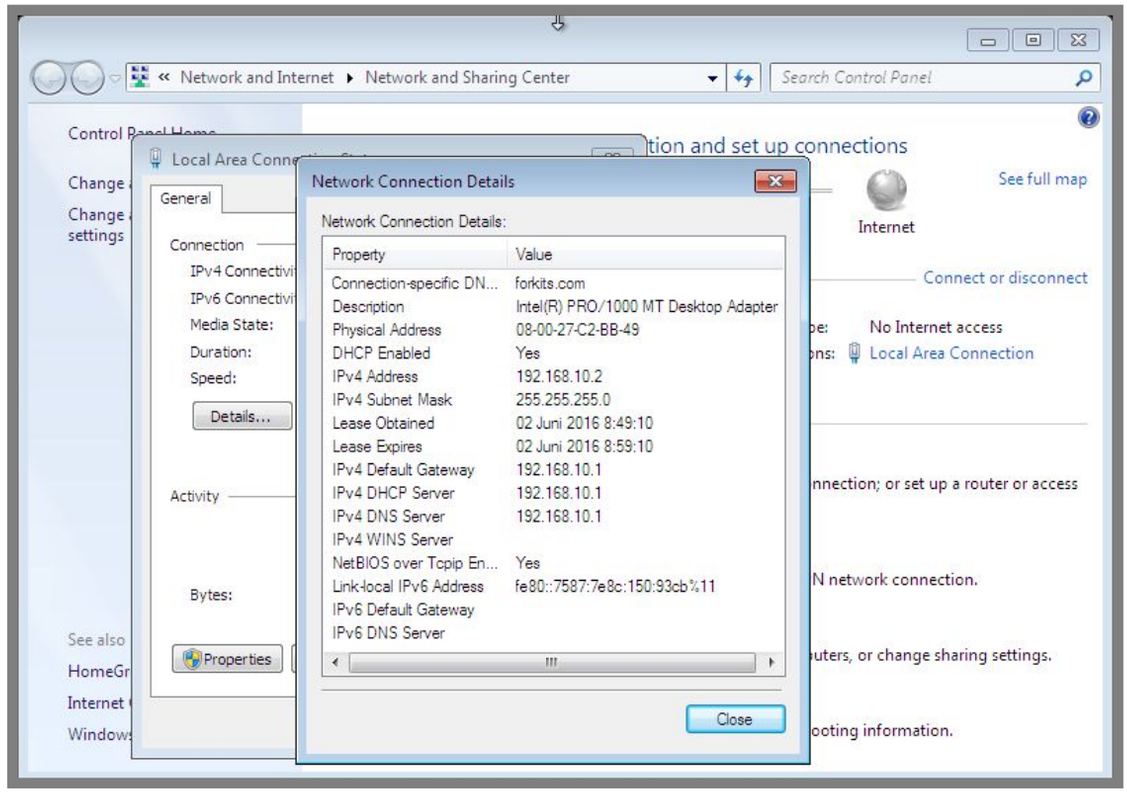
Gambar 16.5 Restart service dhcp server

Sampai saat ini kita telah selesai melakukan konfigurasi dhcp server. Selanjutnya kita bisa melakukan pengujian dari komputer client.

Karena nantinya komputer client akan mendapat ip address secara otomatis dari dhcp server, maka kita tidak perlu melakukan konfigurasi ip address secara manual di client. Kita cukup memilih obtain ip address automaticcally, perhatikan gambar berikut



Gambar 16.6 Konfigurasi dhcp client



Gambar 16.7 Hasil konfigurasi dhcp server

Perhatikan gambar diatas, terlihat bahwa saat ini komputer client telah mendapat ip address secara otomatis dari dhcp server, perhatikan parameter *ipv4 DHCP Server* yang menunjukkan ip address dari dhcp server

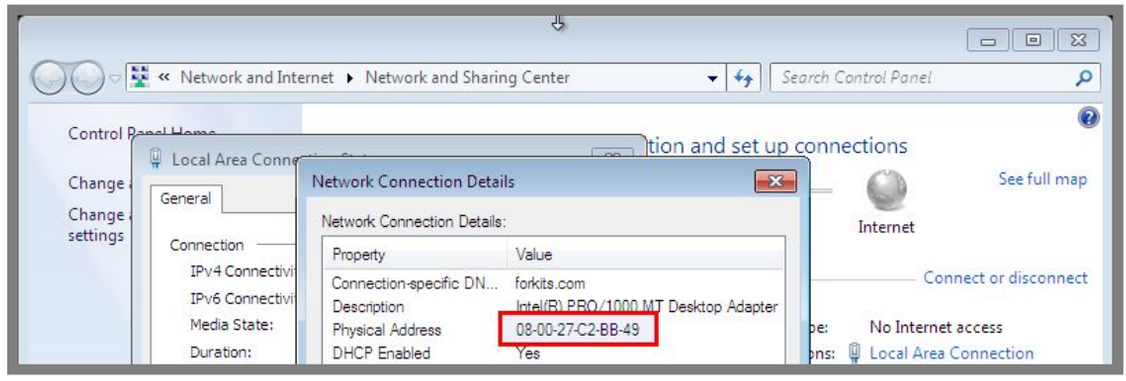
Konfigurasi Fixed IP Address

Sistem kerja dhcp server adalah menyewakan ip address kepada komputer client dengan selang waktu tertentu. Jika waktu sewa (lease time) tersebut habis, maka komputer client akan menyewa ip address lagi ke komputer server, begitu seterusnya. Namun tidak ada jaminan jika ip address yang didapat oleh komputer client selalu sama.

Ada kalanya kita diharuskan untuk mengkonfigurasi agar suatu client selalu mendapat ip address yang sama. Kita tidak perlu hawatir, ada sebuah fitur yang disebut fixed ip address pada isc-dhcp-server. Kita hanya perlu melakukan sedikit tambahan konfigurasi pada dhcp server.

Pada sub bab ini, kita akan praktik menggunakan topologi pada gambar 16.1. Diasumsikan komputer server telah dikonfigurasi dhcp server dan bisa berjalan dengan normal. Selanjutnya kita akan mengkonfigurasi dhcp server agar komputer client selalu mendapat ip address 192.168.10.100/24.

Hal pertama yang harus dilakukan adalah, kita harus mengecek mac address dari komputer client.



Gambar 16.8 Melihat mac address client

Perhatikan gambar diatas, terlihat bahwa mac address dari komputer client adalah 08:00:27:c2:bb:49. Kita akan membutuhkan mac address ini untuk konfigurasi pada dhcp server.

```
root@forkits:~# nano /etc/dhcp/dhcpd.conf
.....
.....
.....
# Fixed IP addresses can also be specified for hosts.   These addresses
# should not also be listed as being available for dynamic assignment.
# Hosts for which fixed IP addresses have been specified can boot using
# BOOTP or DHCP.   Hosts for which no fixed address is specified can only
# be booted with DHCP, unless there is an address range on the subnet
# to which a BOOTP client is connected which has the dynamic-bootp flag
# set.
host administrator {
   hardware ethernet 08:00:27:c2:bb:49;
   fixed-address 192.168.10.100;
}
.....
.....
.....
```

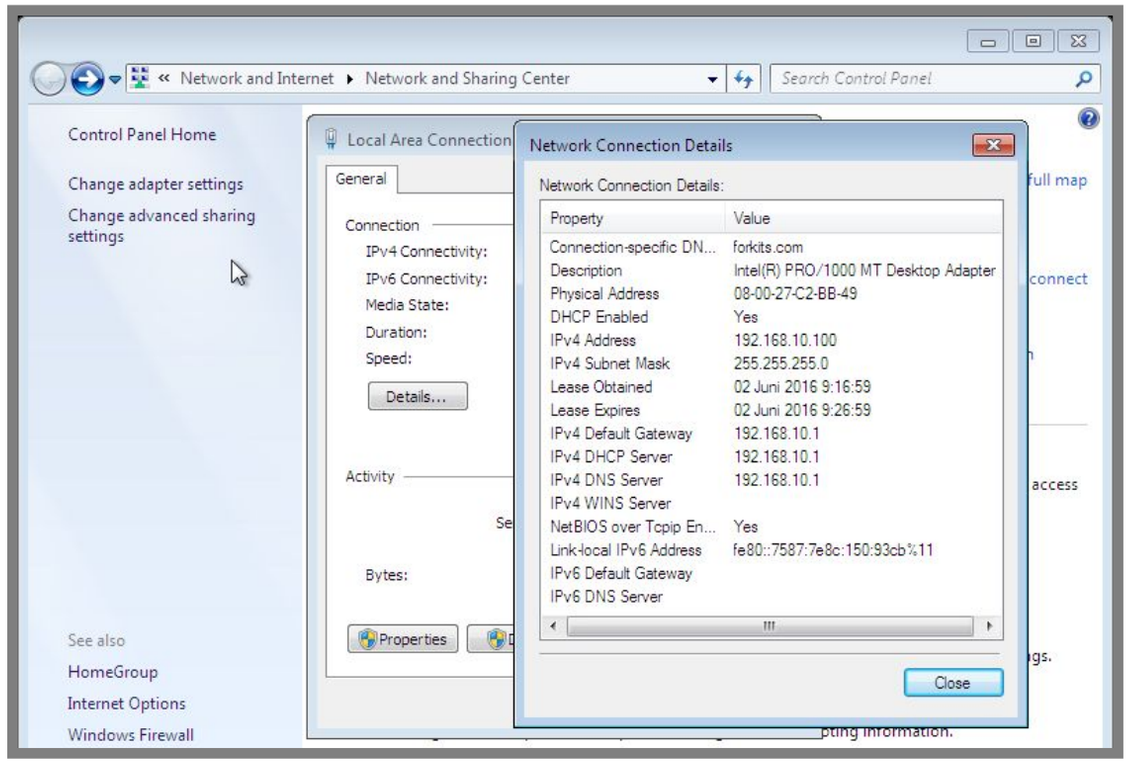
Gambar 16.29 Konfigurasi fiexed ip address

Selanjutnya restart service dhcp

```
root@forkits:~# service isc-dhcp-server restart
[ ok ] Stopping ISC DHCP server: dhcpd.
[ ok ] Starting ISC DHCP server: dhcpd.
root@forkits:~#
```

Gambar 16.30 Restart service dhcp server

Untuk melakukan pengujian dari komputer client, kita bisa coba untuk disable kemudian enable interface ethernet.

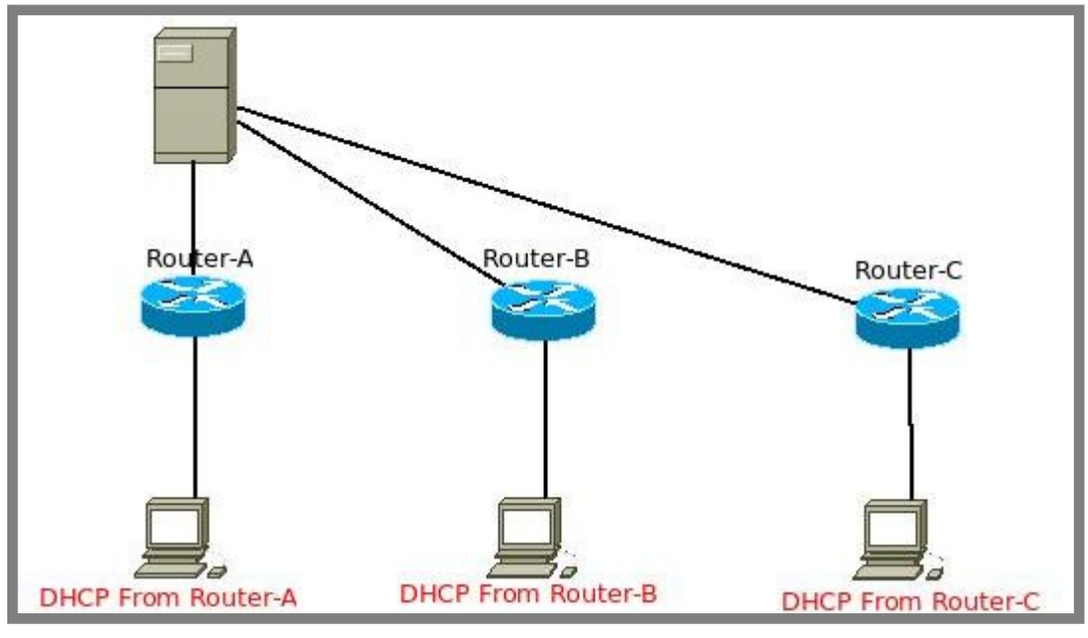


Gambar 16.31 Pengujian di komputer client

Perhatikan gambar diatas, terlihat bahwa komputer client mendapat ip address 192.168.10.100, sesuai dengan konfigurasi yang kita lakukan.

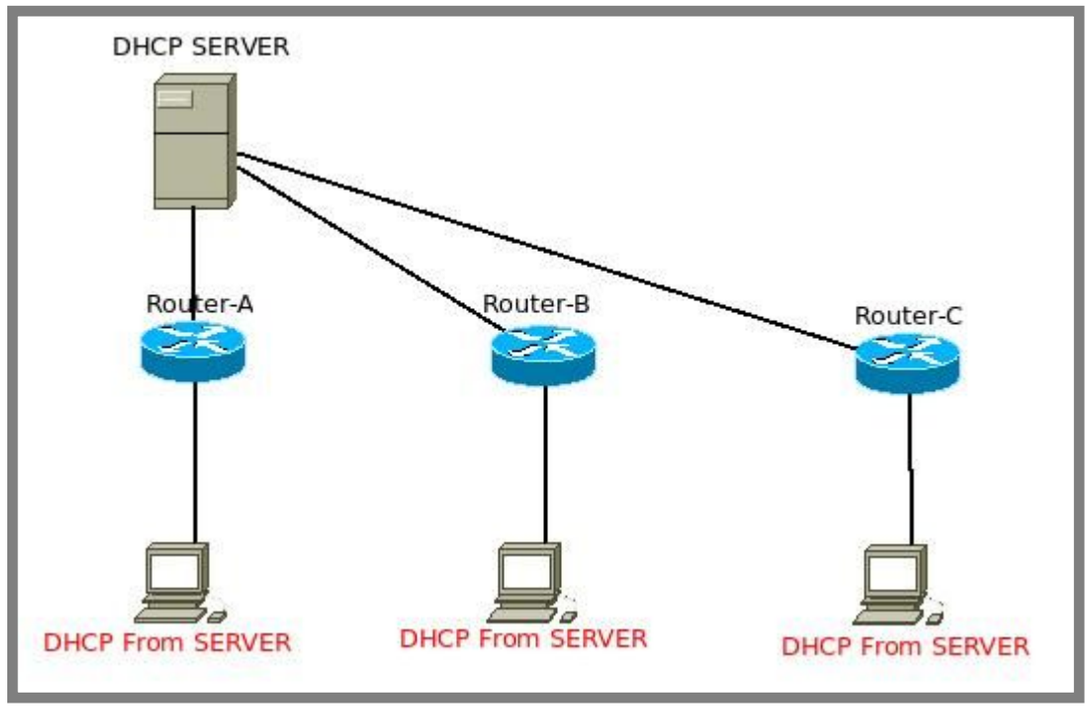
Konfigurasi DHCP Server for DHCP Relay

DHCP Relay merupakan sistem pemberian ip dhcp kepada client secara terpusat yang dilakukan oleh satu server. Perhatikan ilustrasi berikut



Gambar 16.32 Jaringan tanpa dhcp relay

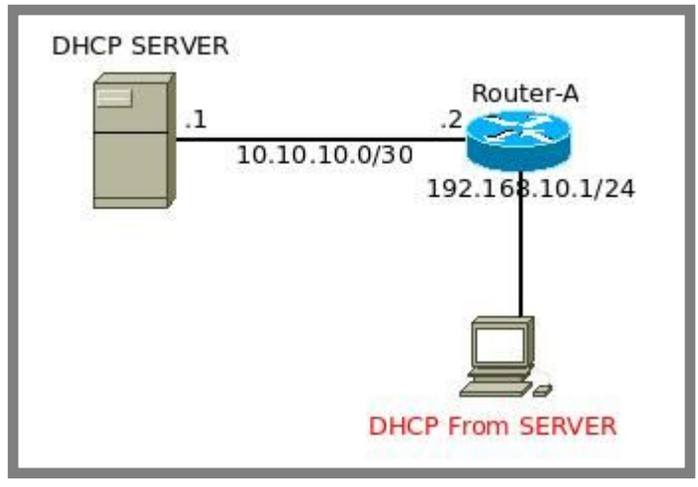
Perhatikan gambar diatas, terlihat bahwa masing-masing client akan mendapat ip dhcp dari routernya masing-masing. Hal ini tentu tidak efisien dalam hal mangement dan monitoring. Bayangkan, kita harus membuat dhcp server di tiga router, dan jika kita ingin memonitoring aktifitas client, kita harus melakukannya di tiga router juga. Karena itu, kita bisa memanfaatkan fitur dhcp relay, perhatikan ilustrasi berikut



Gambar 16.33 Jaringan dengan dhcp relay

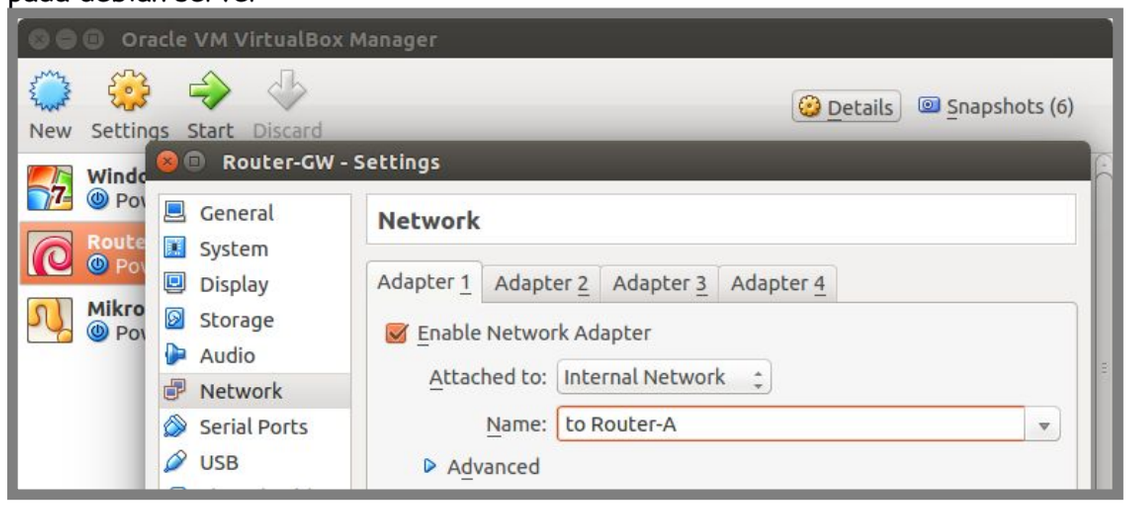
Perhatikan gambar diatas, terlihat bahwa client tidak mendapat dhcp dari routernya masing-masing, melainkan dari server. Hal ini akan memepermudah kita dalam melakukan mangement dan monitoring.

Nantinya server akan bertindak sebagai dhcp server, sedangkan Router-A, Router-B, dan Router-C hanya bertindak sebagai dhcp relay. Kita akan praktik membuat dhcp server sekaligus dhcp relay dengan topologi sebagai berikut



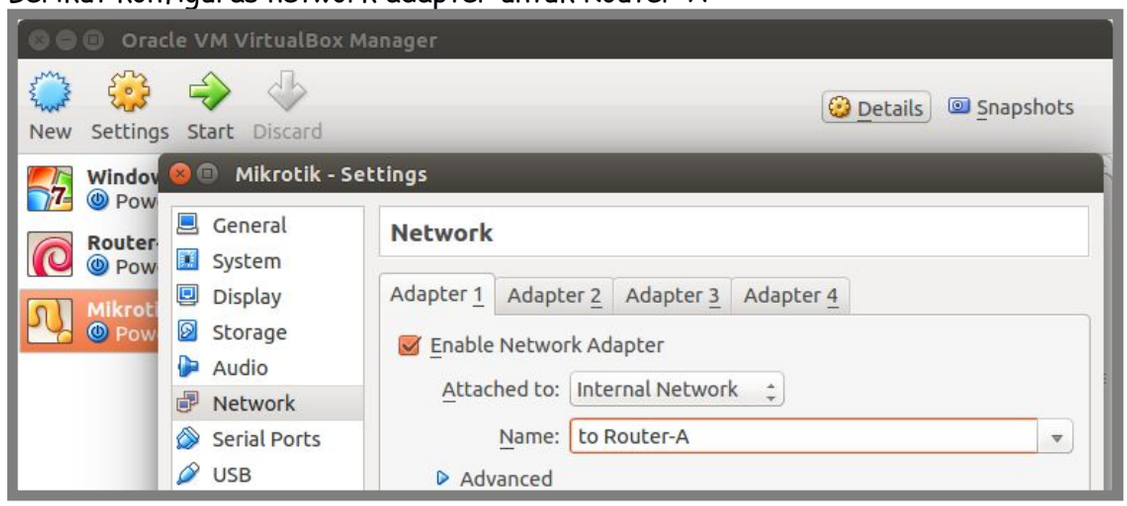
Gambar 16.34 Toplogi jaringan untuk praktik dhcp relay

Kita akan menggunakan guest os debian sebagai server, guest os mikrotik untuk router, dan guest os windows untuk client. Berikut konfigurasi network adapter pada debian server

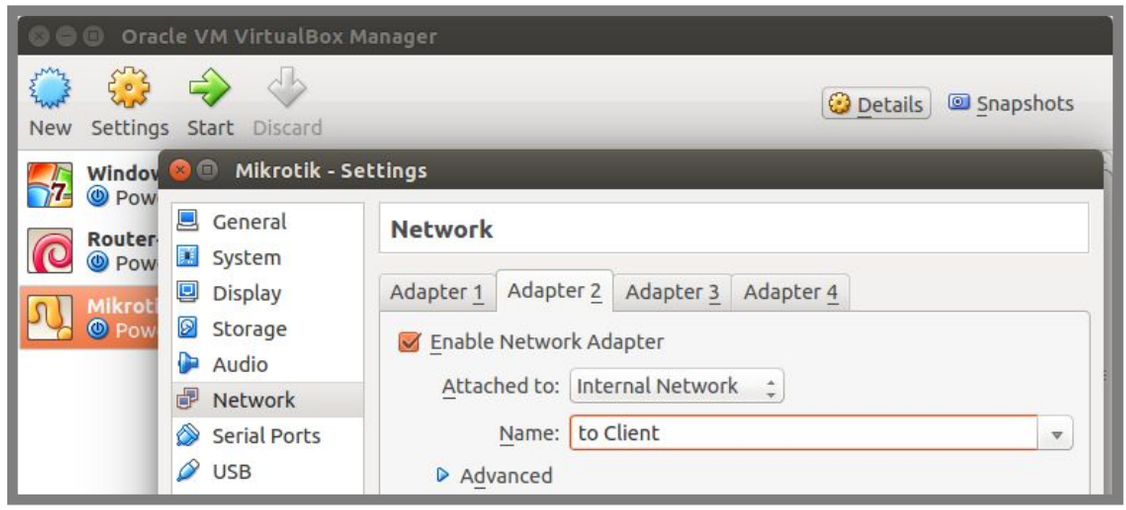


Gambar 16.35 Konfigurasi network adapter pada dhcp server

Berikut konfiguras network adapter untuk Router-A

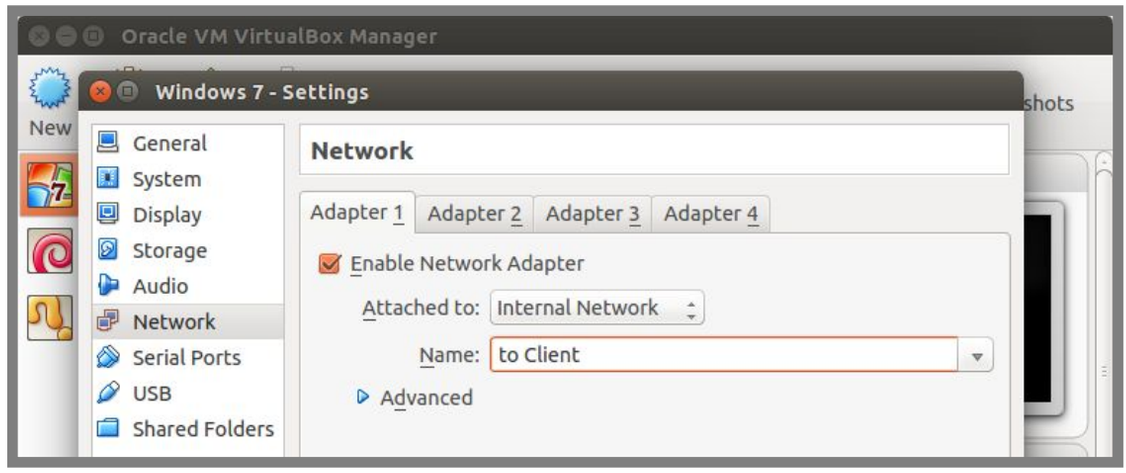


Gambar 16.36 Konfigurasi network adapter pada router



Gambar 16.37 Konfigurasi network adapter pada router

Sedangkan untuk client, berikut konfigurasi network adapternya



Gambar 16.38 Konfigurasi network adapter pada client

Diasumsikan bahwa di komputer server telah dikonfigurasi ip address sesuai dengan topologi pada gambar 16.34. Selanjutnya berikut konfigurasi yang perlu dilakukan pada komputer server

```
root@forkits:~# nano /etc/dhcp/dhcpd.conf
.....
.....
.....
subnet 10.10.10.0 netmask 255.255.255.252 {
   option broadcast-address 10.10.10.3;
   }

# A slightly different configuration for an internal subnet.
subnet 192.168.10.0 netmask 255.255.255.0 {
   range 192.168.10.2 192.168.10.254;
   option domain-name-servers 192.168.10.1;
   option domain-name "forkits.com";
   option routers 192.168.10.1;
   option broadcast-address 192.168.10.255;
   default-lease-time 600;
   max-lease-time 7200;
}
.....
.....
.....
```

Gambar 16.39 Konfigurasi dhcp server for relay

Perhatikan gambar diatas, terlihat bahwa kita harus menambahkan dua network pada konfigurasi dhcp. Network pertama adalah network dari interface server yang terhubung dengan router (10.10.10.0/30), sedangkan network kedua adalah network yang akan diberikan kepada client (192.168.10.0/24). Nantinya komputer

client akan mendapat ip address berdasarkan network kedua, yaitu 192.168.10.0/24. Selanjutnya restart service dhcp

```
root@forkits:~# service isc-dhcp-server restart
[ ok ] Stopping ISC DHCP server: dhcpd.
[ ok ] Starting ISC DHCP server: dhcpd.
root@forkits:~#
```

Gambar 16.40 Restart service dhcp server

Selanjutnya kita harus konfigurasi routing static di komputer server. Hal ini dikarenakan server harus mengetahui informasi tentang ip network yang menuju ke client

```
root@forkits:~# route add -net 192.168.10.0 netmask 255.255.255.0 gw
10.10.10.2
```

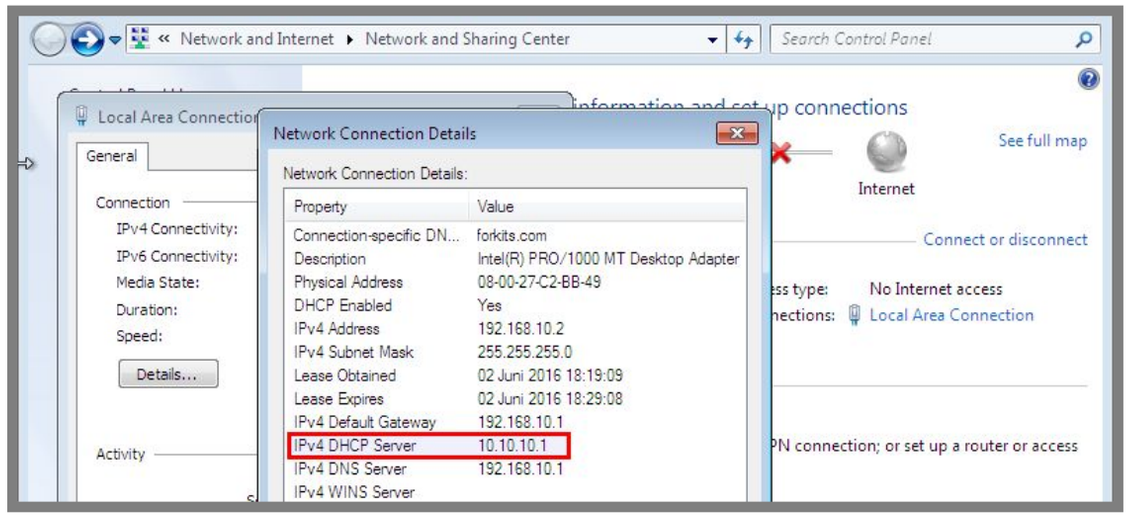
Gambar 16.41 Konfigurasi routing static pada dhcp server

Sampai saat ini kita sudah selesai melakukan konfigurasi pada server. Selanjutnya kita harus melakukan konfigurasi pada router mikrotik, berikut konfigurasi yang perlu dilakukan

```
[admin@MikroTik] > ip address add address=10.10.10.2/30
interface=ether1
[admin@MikroTik] > ip address add address=192.168.10.1/24
interface=ether2
[admin@MikroTik] > ip dhcp-relay add interface=ether2
dhcp-server=10.10.10.1 local-address=192.168.10.1 disabled=no
```

Gambar 16.42 Konfigurasi dhcp relay mikrotik

Terahir, untuk melakukan pengujian kita bisa melakukan obtain pada konfigurasi ip address di komputer client. Berikut hasil pengujian di komputer client



Gambar 16.43 Pengujian dari komputer client

Perhatikan parameter ipv4 dhcp server pada gambar diatas, terlihat bahwa yang menjadi dhcp server adalah 10.10.10.1 (komputer server). Sehingga dapat disimpulkan bahwa kita telah berhasil mengkonfigurasi dhcp server pada debian dan dhcp server pada router mikrotik.

---END OF CHAPTER---

Bab 17
Proxy Server

Proxy merupakan sebuah protocol dalam jaringan yang berfungsi untuk menyimpan halaman-halaman website yang pernah dikunjungi oleh client. Fungsinya adalah sebagai cache, yaitu jika sewaktu-waktu ada client yang mengakses halaman web yang sama, maka client tersebut tidak perlu mengakses halaman web yang ada diinternet, cukup mengakses halaman web yang sudah disimpann dalam proxy server. Hal ini tentu akan sangat menghemat penggunaan bandwidth dan waktu akses ke suatu website.

Selain itu, proxy juga bisa dimanfaatkan untuk keperluan filtering, yaitu dapat digunakan untuk memfilter situs-situs terlarang yang ada diinternet, bisa juga digunakan untuk memfilter client mana saja yang bisa mengakses internet, dan client mana saja yang tidak diperkenankan untuk mengakses internet.

Proxy juga dapat dimanfaatkan untuk melakukan management user, managemen waktu akses internet, dan managemen bandwidth. Management user artinya jika suatu saat ada client yang ingin mengakses internet, maka client tersebut harus memasukkan username dan password. Jika username dan password yang dimasukkan benar, maka client tersebut bisa berselancar di dunia internet, sebaliknya jika username dan password yang dimasukkan salah maka user tersebut tidak akan bisa berselancar di dunia internet.

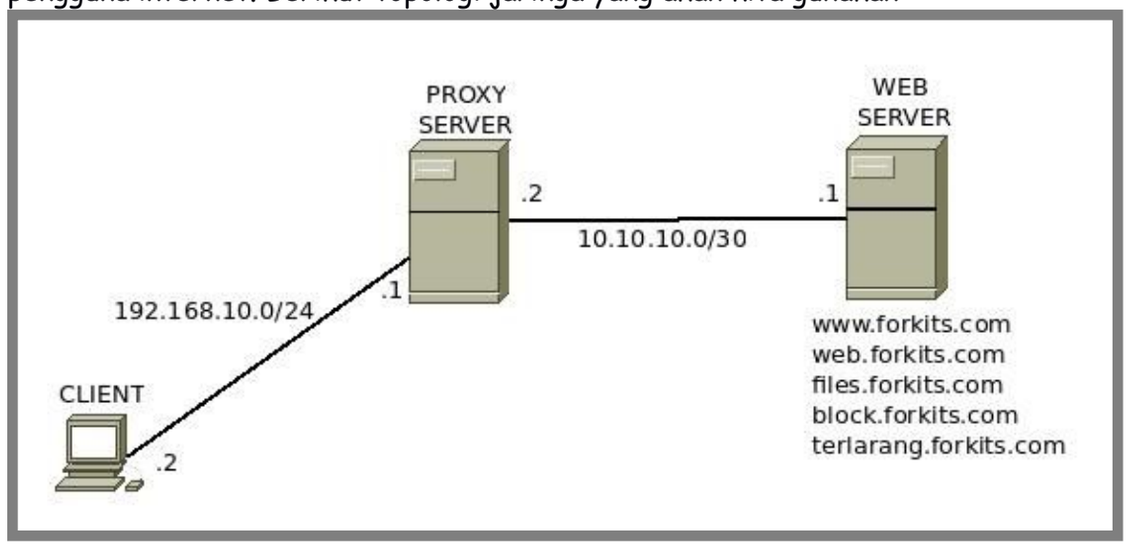
Mangement waktu akses internet artinya proxy dapat kita manfaatkan untuk membatasi akses ke internet pada waktu-waktu tertentu saja. Misal pada hari dan jam tertentu client tidak bisa mengakses internet, selebihnya client dibebaskan untuk mengakses internet.

Mangement bandwidth artinya kita bisa memanfaatkan proxy untuk mengatur berapa besar bandwidth yang dapat digunakan untuk client tertentu saat client tersebut mengakses internet. Management bandwidth yang baik dapat mengoptimalkan kinerja jaringan yang kita miliki.

Konfigurasi Proxy untuk Filtering

Kita akan membahas masing-msaing fungsi proxy yang telah dijelaskan diatas pada sub bab masing-masing. Pada sub bab ini kita akan fokus belajar konfigurasi proxy yang ditujukan untuk keperluan filtering.

Filtering yang dimaksud adalah filtering website-website terlarang dan filtering pengguna internet. Berikut topologi jaringa yang akan kita gunakan

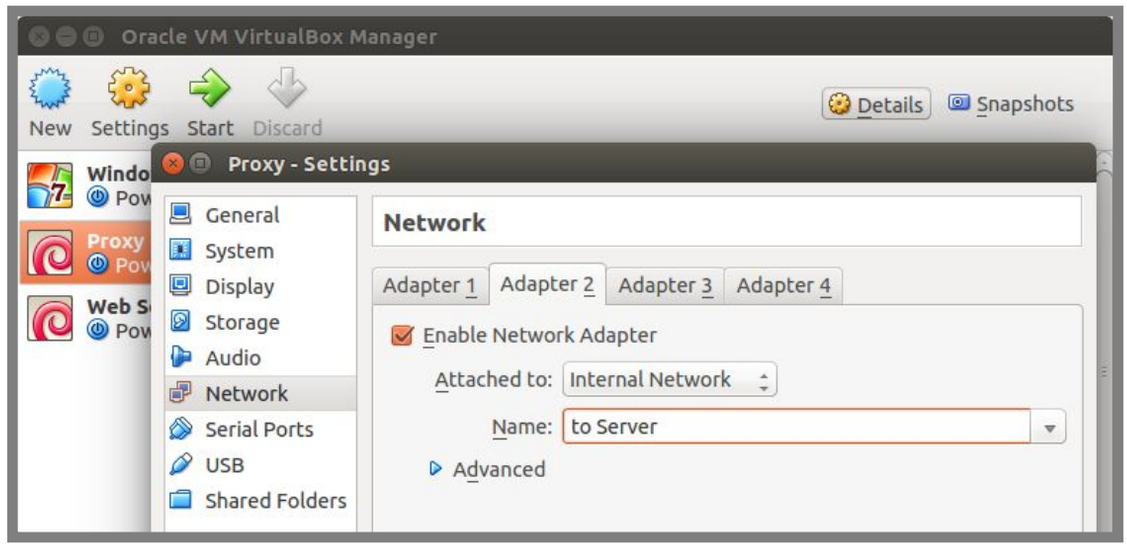


Gambar 17.1 Topologi jaringan untuk praktik proxy server

Pada skenario ini kita akan menggunakan guest os debian sebagai proxy server, begitu juga dengan web server, kita juga akan menggunakan guest os debian. Sedangkan sebagai komputer client kita akan menggunakan host os ubuntu.

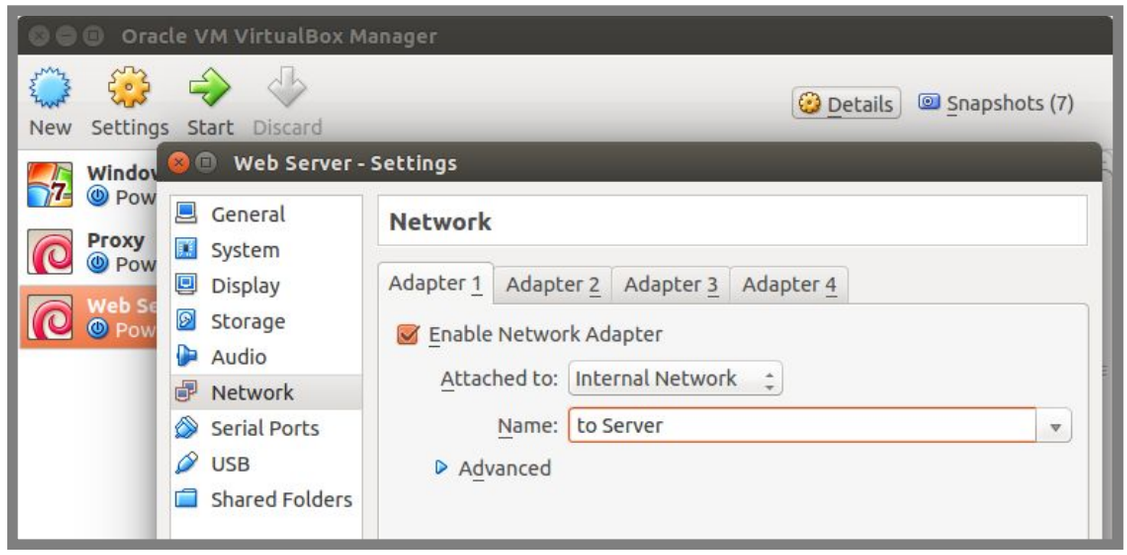
Berikut konfigurasi network adapter pada proxy server

Gambar 17.2 Konfigurasi network adapter pada proxy server



Gambar 17.3 Konfigurasi network adapter pada proxy server

Sedangkan untuk konfigurasi network adapter pada web server adalah sebagai berikut



Gambar 17.4 Konfigurasi network adapter pada web server

Diasumsikan pada web server, telah dikonfigurasi ip address sesuai topologi. Diasumsikan juga service dns server dan web server telah berjalan normal, dimana pada web server terdapat subdomain seperti yang ada pada topologi.

Selanjutnya kita hanya akan fokus pada konfigurasi di proxy server. Diasumsikan pula bahwa pada proxy server telah dikonfigurasi ip address sesuai topologi dan telah diaktifkan fungsi routing.

Kita akan membuat proxy server menggunakan aplikasi squid. Squid merupakan salah satu aplikasi yang dapat kita gunakan untuk membuat proxy server yang sangat populer. Berikut perintah yang dapat kita gunakan untuk menginstall squid di debian

```
root@Router-Proxy:~# apt-get install squid
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  squid-common squid-langpack
Suggested packages:
  squidclient squid-cgi logcheck-database resolvconf smbclient winbind
The following NEW packages will be installed:
  squid squid-common squid-langpack
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/1504 kB of archives.
After this operation, 4579 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 17.5 Installasi squid untuk proxy server

Diasumsikan nantinya tujuan kita adalah memblokir akses ke website http://block.forkits.com dan http://terlarang.forkits.com, sekaligus memblokir client dengan ip address 192.168.10.10-192.168.10.20.

Berikut konfigurasi yang perlu dilakukan untuk mewujudkan asumsi-asumsi diatas

```
root@Router-Proxy:~# nano /etc/squid/squid.conf
#  TAG: cache_mgr
#       Email-address of local cache manager who will receive
#       mail if the cache dies. The default is "webmaster".
cache_mgr admin@forkits.com      >> cari dengan kata kunci cache_mgr

visible_hostname www.forkits.com >> cari dengan kata kunci visible_hostname
#       If you want to present a special hostname in error messages, etc,
#       define this.   Otherwise, the return value of gethostname()

# And finally deny all other access to this proxy
#http_access deny all >> cari dengan kata kunci http_access2

#  TAG: http_access2
#       Allowing or Denying access based on defined access lists
#
#       Identical to http_access, but runs after redirectors. If not set
#       then only http_access is used.
```

Gambar 17.6 Konfigurasi proxy server

Masih di dalam file tersebut, lakukan penambahan konfigurasi sebagai berikut

```
acl Safe_ports port 488          # gss-http
acl Safe_ports port 591          # filemaker
acl Safe_ports port 777          # multiling http
acl Safe_ports port 631          # cups
acl Safe_ports port 873          # rsync
acl Safe_ports port 901          # SWAT
acl purge method PURGE
acl CONNECT method CONNECT
                                 >> cari dengan kata kunci acl connect
                                kemudian tambahkan baris warna hijau
acl lan src 192.168.10.0/24
acl user_nakal src 192.168.10.10-192.168.10.20
acl url dstdomain "/etc/squid/url"
http_access deny user_nakal
http_access deny url
http_access allow lan
```

Gambar 17.7 Konfigurasi proxy server

Perhatikan gambar diatas, terlihat baris pertama kita mendeklarasikan sebuah network 192.168.10.0/24 dengan nama "*lan*". Pada baris kedua kita mendeklarasikan ip range 192.168.10.10-192.168.10.20 dengan nama "*user_nakal*". Kita juga mendeklarasikan sebuah file url dengan nama "*url*" di */etc/squid/url* pada baris ketiga. Selanjutnya kita membuat policy untuk menolak "*user_nakal*" pada baris empat, begitu juga dengan baris kelima kita menolak "*url*". Sedangkan pada baris keenam kita mendeklarasikan policy untuk mengizinkan "*lan*".

Hati-hati dalam membuat policy-policy pada squid. Perlu diketahui bahwa squid membaca policy-policy tersebut dari atas kebawah. Selanjutnya kita harus membuat list url yang akan diblokir

```
root@Router-Proxy:~# nano /etc/squid/url
block.forkits.com
terlarang.forkits.com
```

Gambar 17.8 Dafatar url yang diblokir

Terahir restart service squid

```
root@Router-Proxy:~# service squid restart
[ ok ] Restarting Squid HTTP proxy: squid.
root@Router-Proxy:~#
```

Gambar 17.9 Restart service proxy server

Selanjutnya kita harus mengkonfigurasi dns resolver pada proxy server agar mengarah ke web server

```
root@Router-Proxy:~# nano /etc/resolv.conf
nameserver 10.10.10.1
```

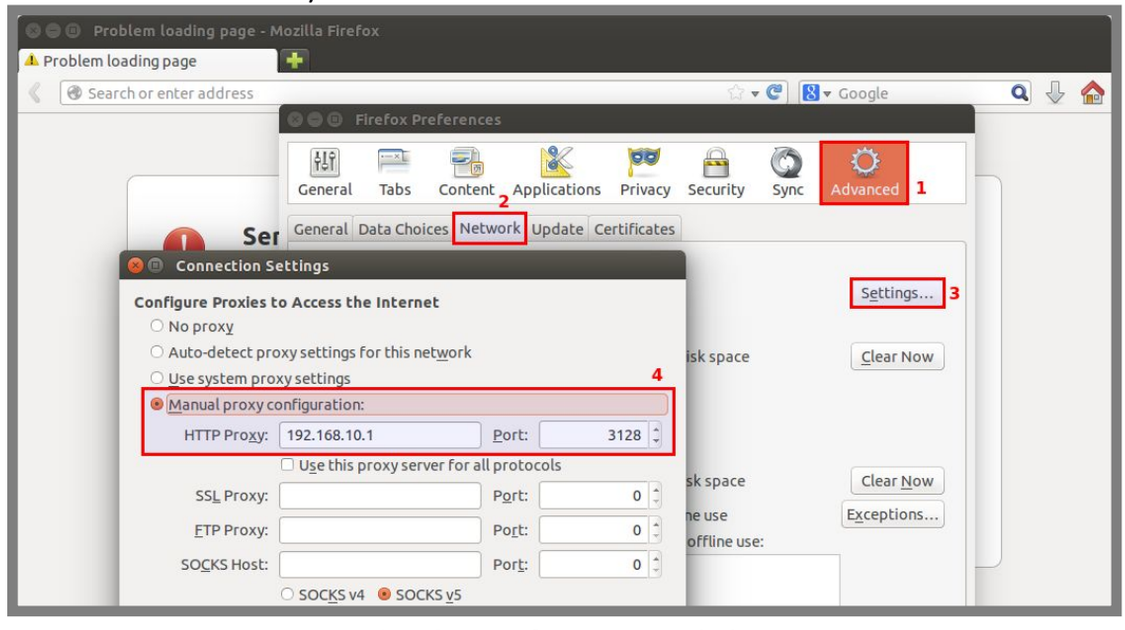
Gambar 17.10 Konfigurasi dns resolver

Sampai saat ini kita telah selesai mengkonfigurasi proxy server untuk keperluan filtering. Selanjutnya kita harus melakukan sedikit konfigurasi pada komputer client. Namun sebelumnya pastikan agar pada komputer client telah dikonfigurasi ip address sesuai topologi jaringan. Jangan lupa juga untuk mengarahkan gateway ke 192.168.10.1 dan dns resolver ke 10.10.10.1. Pastikan bahwa client telah bisa resolve www.forkits.com

```
admin@ubuntu:~$ cat /etc/resolv.conf
nameserver 10.10.10.1
admin@ubuntu:~$ nslookup forkits.com
Server:         10.10.10.1
Address:        10.10.10.1#53

Name:   forkits.com
Address: 10.10.10.1

admin@ubuntu:~$
```

Gambar 17.11 Client telah dapat meresolve domain milik web server

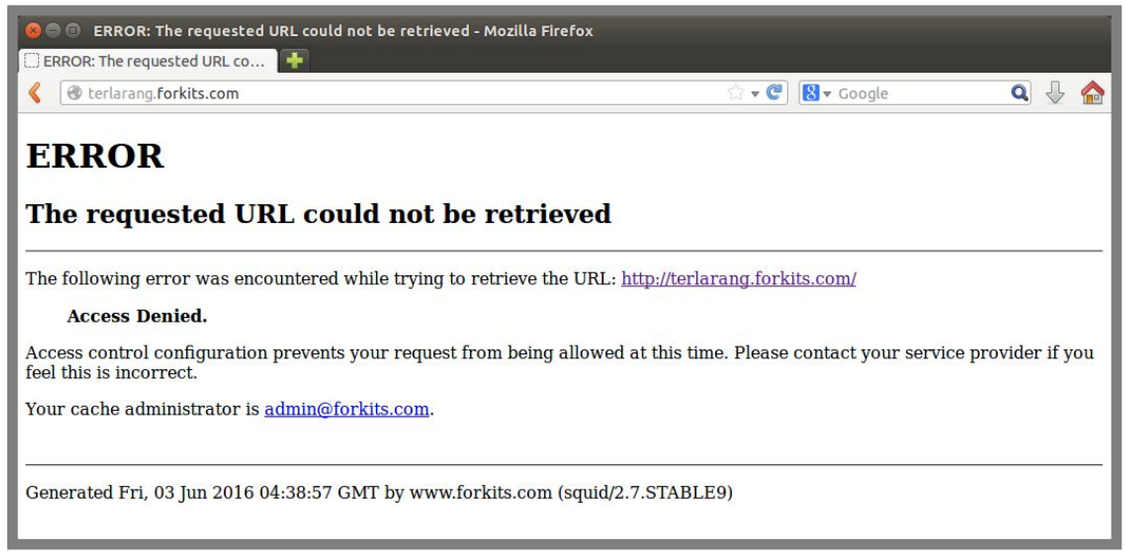
Jika komputer client sudah bisa meresolve domain milik web server seperti diatas, selanjutnya kita harus melakukan konfigurasi proxy di web browser miliki client, masuk ke menu *edit >> preference*



Gambar 17.12 Konfigurasi proxy di client

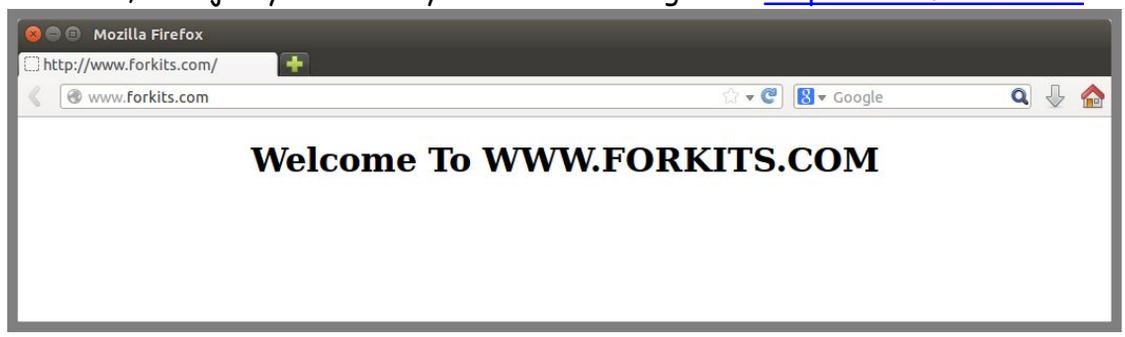
Berikut hasil pengujian saat mengakses website http://block.forkits.com dan http://terlarang.forkits.com

Gambar 17.13 Pengujian saat mengakses http://block.forkits.com



Gambar 17.14 Pengujian saat mengakses http://terlarang.forkits.com

Perhatikan bahwa saat ini client sudah tidak bisa mengakses kedua website tersebut, selanjutnya seharusnya client bisa mengakses http://www.forkits.com



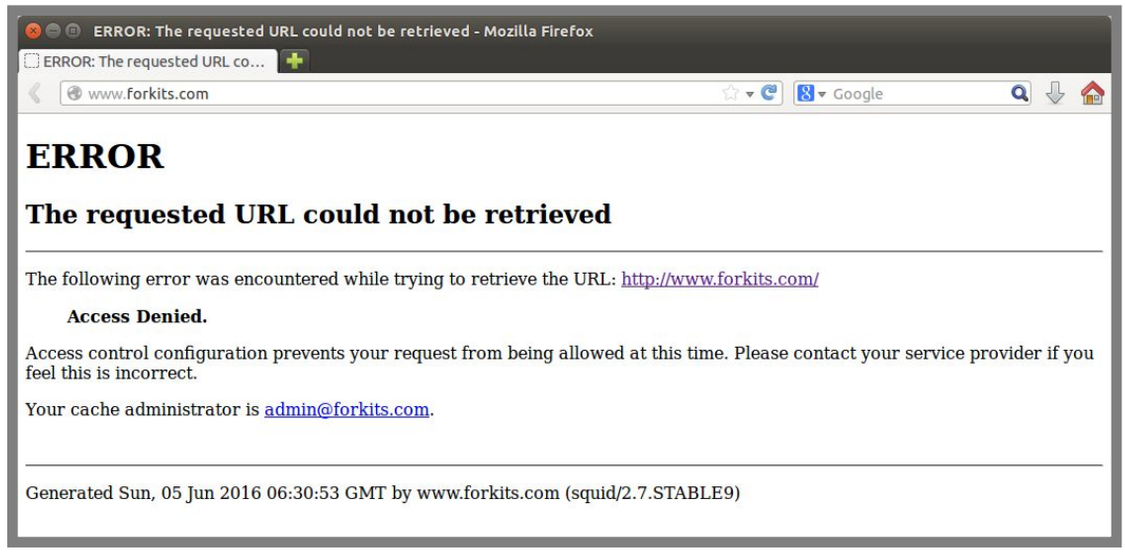
Gambar 17.15 Pengujian saat mengakses http://www.forkits.com

Sekarang kita akan mencoba mengganti ip address pada client menjadi ip address yang termasuk dalam rentang yang diblokir oleh proxy squid.

```
admin@ubuntu:~$ sudo ifconfig vboxnet0 192.168.10.12/24
[sudo] password for admin: (tak terlihat)
admin@ubuntu:~$ sudo route add default gw 192.168.10.1
```

Gambar 17.16 Mengganti ip address di client

Perhatikan gambar berikut yang menunjukkan hasil pengujian saat client mencoba mengakses http://www.forkits.com



Gambar 17.17 Pengujian saat mengakses http://www.forkits.com

Perhatikan bahwa hasilnya juga dideny (ditolak), padahal seharusnya http://www.forkits.com tidak termasuk dalam website yang diblokir. Hal ini dikarenakan kita menggunakan ip address yang termasuk dalam range yang diblokir oleh proxy squid.

Konfigurasi Proxy Untuk Managemen User

Fungsi ini memungkinkan kita untuk membatasi akses internet hanya oleh user tertentu saja yang memang kita beri username dan password untuk mengakses internet. User yang tidak kita beri username dan password tidak akan bisa mengakses internet.

Kita akan menggunakan toplogi jaringan yang sama dengan gambar 17.1, kita hanya akan melakukan sedikit tambahan konfigurasi pada proxy squid yang telah kita konfigurasi pada sub bab sebelumnya.

Berikut konfigurasi tambahan yang perlu dilakukan

```
root@Router-Proxy:~# nano /etc/squid/squid.conf
#
>>cari dengan kata kunci ncsa_auth
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd

acl lan src 192.168.10.0/24
acl user_nakal src 192.168.10.10-192.168.10.20/32
acl url dstdomain "/etc/squid/url"
acl user proxy_auth REQUIRED
http_access deny !user
http_access deny user_nakal
http_access deny url
http_access allow lan
```

Gambar 17.18 Konfigurasi proxy untuk authentikasi

Baris pertama (*auth_param basic....*) artinya kita mengkonfigurasi squid agar mendukung authentikasi dengan tipe basic. Sedangkan file authentikasi disimpan di */etc/squid/passwd*. Selanjutnya pada baris *acl user proxy_auth....*, artinya agar squid selalu meminta username dan password saat client akan mengakses internet. Sedangkan *http_access deny !user* artinya selain client yang memasukkan username dan password akan ditolak.

Selanjutnya kita harus membuat uesername dan password yang nantinya akan digunakan oleh client, misal kita akan membuat username dengan nama *usersquid* dan password *123456*

```
root@Router-Proxy:~# htpasswd -c /etc/squid/passwd usersquid
New password: (tak terlihat)
Re-type new password: (tak terlihat)
Adding password for user usersquid
root@Router-Proxy:~#
```

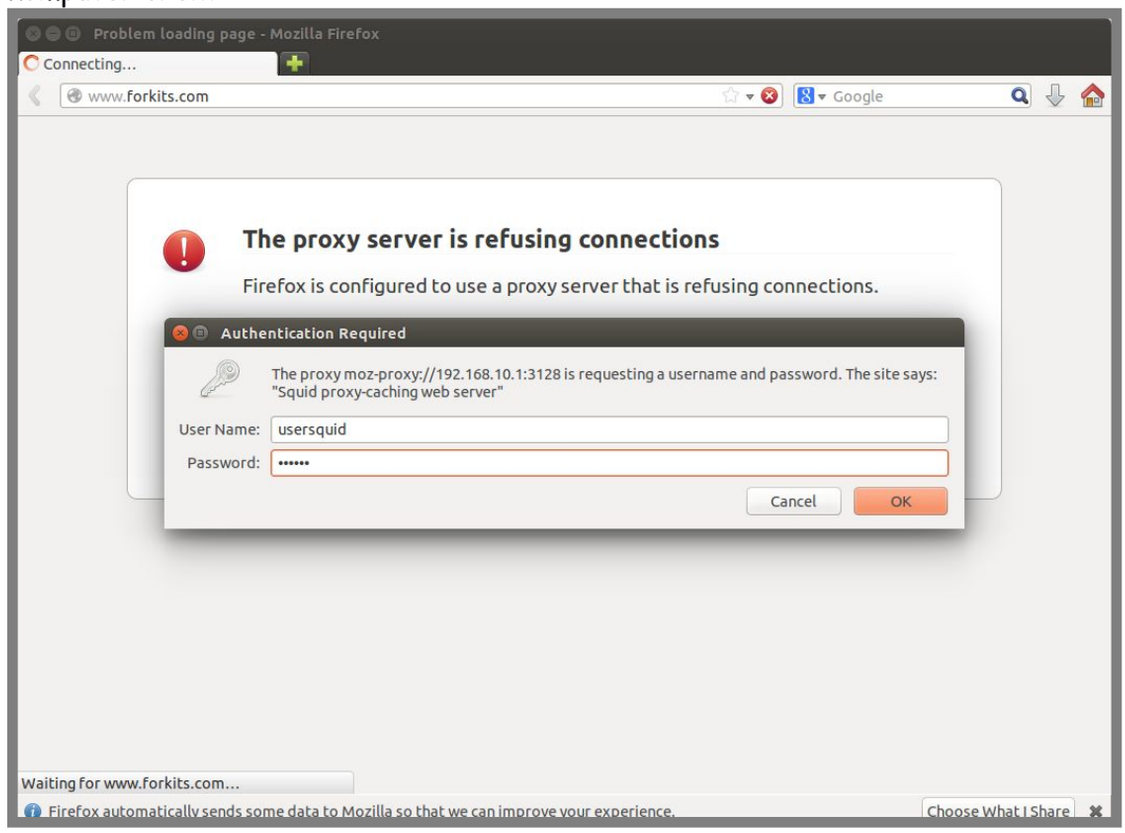
Gambar 17.19 Membuat user untuk proxy

Perlu diketahui bahwa untuk menjalankan perintah diatas (htpasswd), pada komputer proxy harus sudah terinstall paket apache2. Terahir restart service squid

```
root@Router-Proxy:~# service squid restart
[ ok ] Restarting Squid HTTP proxy: squid.
root@Router-Proxy:~#
```

Gambar 17.20 Restart service proxy server

Sampai saat ini kita sudah selesai melakukan konfigurasi proxy squid untuk keperluan managemen user. Selanjutnya kita bisa melakukan pengujian dari komputer client



Gambar 17.21 Pengujian dari komputer client

Perhatikan gambar diatas, terlihat bahwa saat client ingin mengakses http://www.forkits.com, maka client akan diminta oleh proxy untuk memasukkan username dan password.

Proxy Untuk Managemen Waktu Akses Internet

Kita telah membahas dua fungsi proxy pada sub bab sebelumnya, yaitu proxy untuk filtering dan managemen user. Pada sub bab ini kita akan membahas proxy untuk keperlua mangemen waktu akses internet.

Kita akan menggunakan topologi jaringan yang sama dengan pembahasan pada sub bab sebelumnya, yaitu topologi jaringan pada gambar 17.1. Nantinya kita akan melakukan konfigurasi agar client hanya bisa menggunakan internet pada jam kerja, yaitu hari senin-kamis pukul 09:00-16:00. Selain waktu tersebut, maka user tidak akan bisa menggunakan internet.

Selain itu, kita akan mengkonfigurasi proxy agar memperbolehkan administrator jaringan dengan ip address 192.168.10.2/24 untuk mengakses internet sepanjang waktu.

Berikut konfigurasi yang perlu dilakukan untuk mewujudkan skenario tersebut

```
root@Router-Proxy:~# nano /etc/squid/squid.conf
#
>>cari dengan kata kunci ncsa_auth
#auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd

acl lan src 192.168.10.0/24
acl adminstrator src 192.168.10.2
acl url dstdomain "/etc/squid/url"
acl jamkerja time MTWH 09:00-16:00
http_access deny url
http_access allow administrator
http_access deny lan !jamkerja
```

Gambar 17.22 Konfigurasi proxy untuk managemen waktu akses internet

Perhatikan gambar diatas, inti dari perubhan yang dilakukan adalah kita mendisable fungsi proxy sebagai managemen user, kemudian kita mengaktifkan fungsi proxy sebagai managemen waktu akses internet.

Perhatikan baris *acl administrator.....* Baris tersebut artinya kita membuat sebuah access control list untuk ip address administrator, yaitu 192.168.10.2. Selanjutnya kita juga membuat access control list untuk waktu jam kerja pada baris *acl jamkerja time.....* Arti dari baris tersebut adalah kita membuat access control list untuk hari senin-kamis (monday-thursday) jam 9 pagi sampai jam 4 sore (09:00-16:00). Berikut kata kunci huruf yang menandakan masing-masing hari dalam satu minggu

- S -> Sunday (Minggu)
- M -> Monday (Senin)
- T -> Tuesday (Selasa)
- W -> Wednesday (Rabu)
- H -> Thursday (Kamis)
- F -> Friday (Jumat)
- A -> Saturday (Sabtu)

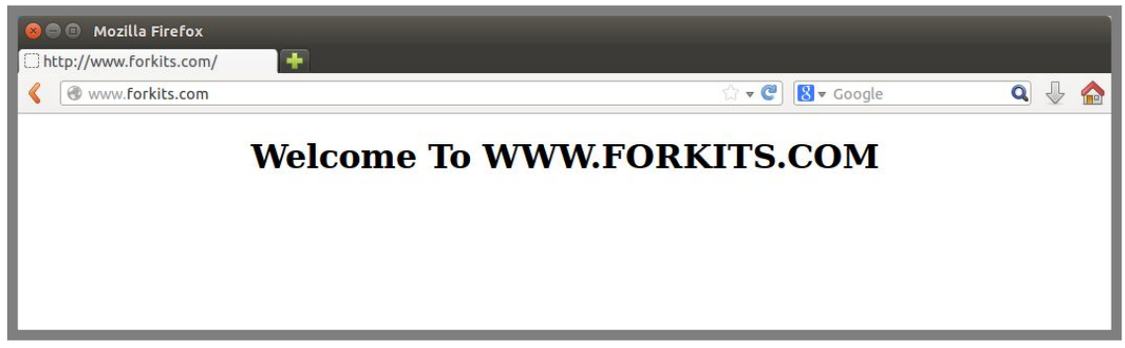
Selanjutnya kita membuat sebuah rule untuk memperbolehkan akses dari administrator (*http_access allow administrator*). Perhatikan bahwa rule tersebut kita letakkan setelah rule untuk menolak akses ke website-website terlarang, itu artinya administrator juga tidak akan bisa mengakses website-website teralrang tersebut. Selanjutnya kita juga membuat sebuah rule untuk menolak akses dari jaringan client (lan) selain pada jam kerja (*http_access deny lan !jamkerja*). Tanda negasi (!) memiliki arti *kecuali/selain.*

Selanjutnya restart service squid

```
root@Router-Proxy:~# service squid restart
[ ok ] Restarting Squid HTTP proxy: squid[....]   Waiting.....done.
. ok
root@Router-Proxy:~#
```

Gambar 17.23 Restart service proxy server

Berikut pengujian yang dilakukan dari komputer client yang menggunakan ip address 192.168.10.2/24 (komputer server). Saat saya melakukan pengujian, waktu menunjukkan pukul 08:42 (hari selasa). Itu artinya saat ini adalah jam diluar jam kerja.



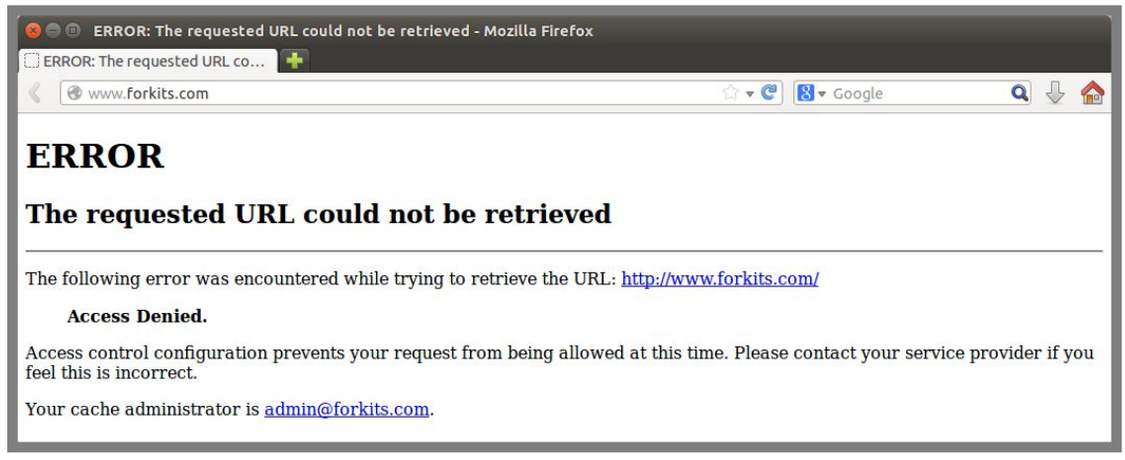
Gambar 17.24 Pengujian menggunakan administrator

Perhatikan gambar diatas, terlihat bahwa administrator tetap bisa melakukan akses internet walau diluar jam kerja. Selanjutnya kita coba ganti ip address pada komputer client agar tidak menggunakan ip administrator

```
admin@ubuntu:~$ sudo ifconfig vboxnet0 192.168.10.3/24
[sudo] password for admin: (tak terlihat)
admin@ubuntu:~$ sudo route add default gw 192.168.10.1
```

Gambar 17.25 Konfigurasi ip address client

Berikut hasil pengujian yang dilakukan oleh client saat menggunakan ip address 192.168.10.3



Gambar 17.26 Pengujian menggunakan client non administrator

Perhatikan gambar diatas, terlihat bahwa ada peringatan *Access Denied* saat kita mencoba melakukan akses dari ip address non administrator. Hal ini dikarenakan saat ini kita masih berada diluar jam kerja.

Konfigurasi Proxy Untuk Managemen Bandwidth

Fungsi terahir proxy yang akan kita bahas pada buku ini adalah proxy sebagai tool untuk melakukan managemen bandwidth. Kita tentu perlu melakukan managemen bandwidth yang baik pada jaringan agar jaringan yang kita miliki dapat beroperasi dengan baik.

Skenarionya adalah, jika ada client yang melakukan download file lebih dari 2 MB (2.000.000 Byte), maka bandwidthnya akan di drop menjadi 100 KB (100.000 Byte) saja. Namun perlu diketahui, managemen bandwidth ini hanya berlaku untuk protocol http saja, tidak berlaku pada protocol ftp, ataupun sftp.

Berikut konfigurasi-konfigurasi yang perlu dilakukan

```
root@Router-Proxy:~# nano /etc/squid/squid.conf
#
acl lan src 192.168.10.0/24
acl adminstrator src 192.168.10.2
acl url dstdomain "/etc/squid/url"
acl jamkerja time MTWH 09:00-16:00
http_access deny url
http_access allow administrator
http_access deny lan !jamkerja

delay_pools 1
delay_class 1 1
delay_parameters 1 100000/2000000
```

Gambar 17.27 Konfigurasi proxy untuk managemen bandwidth

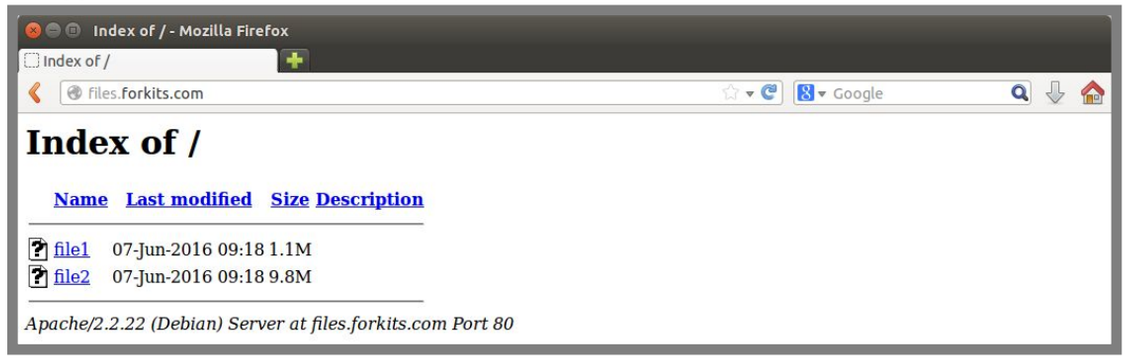
Selanjutnya restart service squid

```
root@Router-Proxy:~# service squid restart
[ ok ] Restarting Squid HTTP proxy: squid[....]   Waiting.....done.
. ok
root@Router-Proxy:~#
```

Gambar 17.28 Restart service proxy server

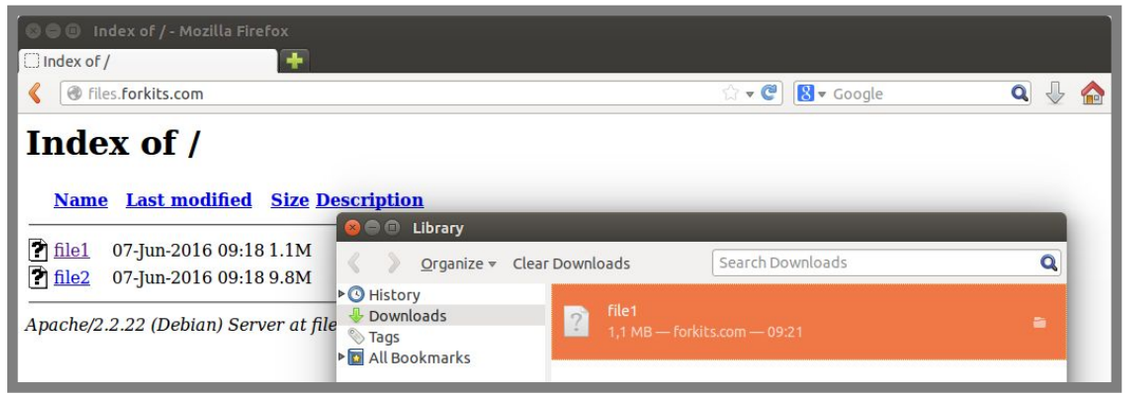
Untuk melakukan pengujian, kita akan menggunakan fitur virtual direktori pada web server. Jika teman-teman lupa apa itu virtual direktory, teman-teman bisa membacanya kembali pada bab web server.

Diasumsikan kita telah mempunyai file dengan ukuran 1 MB dan 10 MB pada virtual direktory web server (teman-teman bisa upload file tersebut ke web server menggunakan ftp maupun sftp).



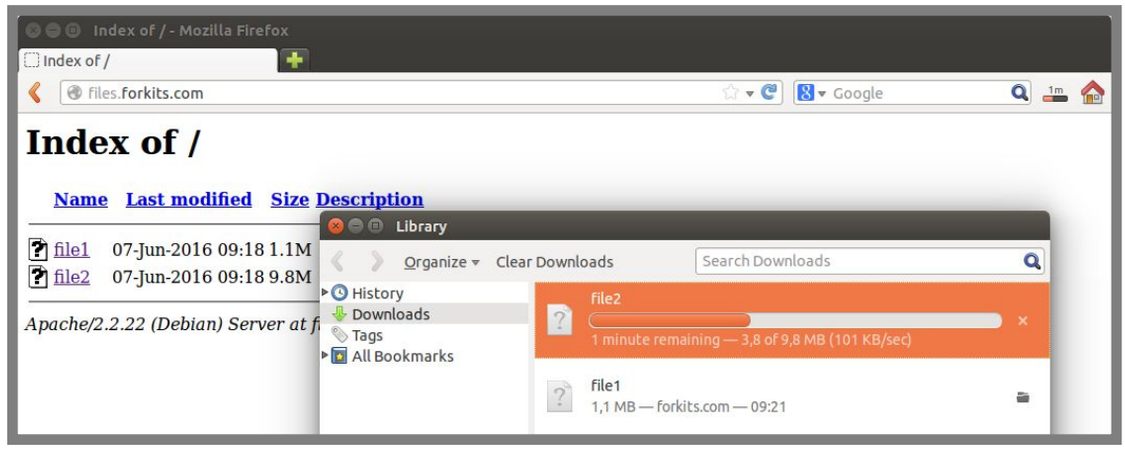
Gambar 17.29 Virtual direktori pada web server

Perhatikan gambar berikut yang menunjukka proses download file1 (1 MB)



Gambar 17.30 Download file 1 MB

Perhatikan gambar diatas, terlihat bahwa proses download akan langsung selesai, hal ini dikarenakan kita tidak melakukan limit bandwidth untuk aktifitas download yang kurang dari 2 MB. Selanjutnya berikut saat kita mencoba download file2 (10 MB)



Gambar 17.31 Download file 10 MB

Perhatikan gambar diatas, terlihat bahwa kita hanya akan mendapat bandwidth sekitar 100 KB saat kita mencoba mendownload file yang ukurannya lebih dari 2 MB. Hal ini tentu sudah sesuai dengan konfigurasi yang kita lakukan pada proxy.

Konfigurasi Transparent Proxy

Selama ini kita masih membuat manual proxy, artinya client harus mengkonfigurasi secara manual pada browser mereka masing-masing jika mereka ingin menggunakan proxy server.

Pada sub bab ini kita akan membahas bagaimana membuat transparent proxy, sehingga nantinya client tidak perlu melakukan konfigurasi apapun jika ingin menggunakann proxy. Begitu juga jika client tidak ingin menggunakan proxy, mereka tetap akan dipaksa agar menggunakan proxy secara otomatis. Sehingga mungkin saja client tidak tahu jika sebenarnya dia menggunakan proxy. Karena konfigurasi proxy murni hanya dilakukan pada proxy server.

Berikut konfigurasi yang perlu dilakukan pada squid

```
root@Router-Proxy:~# nano /etc/squid/squid.conf
#
# Squid normally listens to port 3128
http_port 3128 transparent >> cari dengan kata kunci http_port 3128
```

Gambar 17.32 Konfigurasi transparent proxy

Selanjutnya restart service squid

```
root@Router-Proxy:~# service squid restart
[ ok ] Restarting Squid HTTP proxy: squid[....]   Waiting.....done.
. ok
root@Router-Proxy:~#
```

Gambar 17.32 Restart service proxy server

Untuk membuat transparent proxy, kita harus membuat rule firewall untuk meredirect port http (80) ke port proxy (3128)

```
root@Router-Proxy:~# nano /etc/rc.local
#
# By default this script does nothing.
iptables -t nat -A PREROUTING -s 192.168.10.0/24 -p tcp --dport 80 -j
REDIRECT --to-port 3128
exit 0
```

Gambar 17.33 Firewall nat untuk transparent proxy

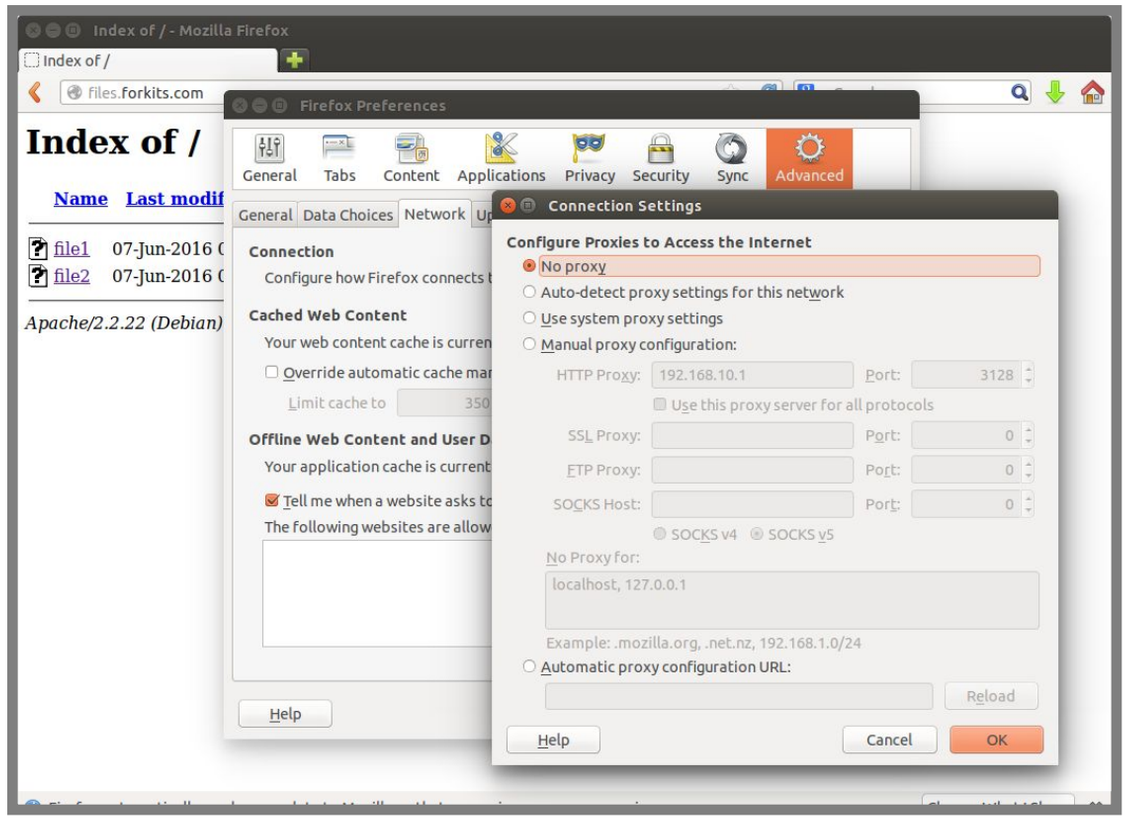
Perhatikan gambar diatas, arti dari script tersebut adalah kita menambahkan sebuah firewall nat dengan chain PREROUTING, src address 192.168.10.0/24 (ip network client), dengan tujuan protocol tcp port 80 (http), kemudian actionnya adalah REDIRECT ke port 3128 (proxy). Kita akan membahas lebih lanjut tentang firewall pada bab husus yang membahas firewall. Perlu diketahui bahwa script tersebut harus ditulis dalam satu baris.

Selanjutnya untuk menjalankan perubahan yang kita lakukan pada file rc.local diatas, kita harus merestart komputer, atau cukup dengan menjalankan perintah */etc/rc.local* seperti berikut

```
root@Router-Proxy:~# /etc/rc.local
root@Router-Proxy:~#
```

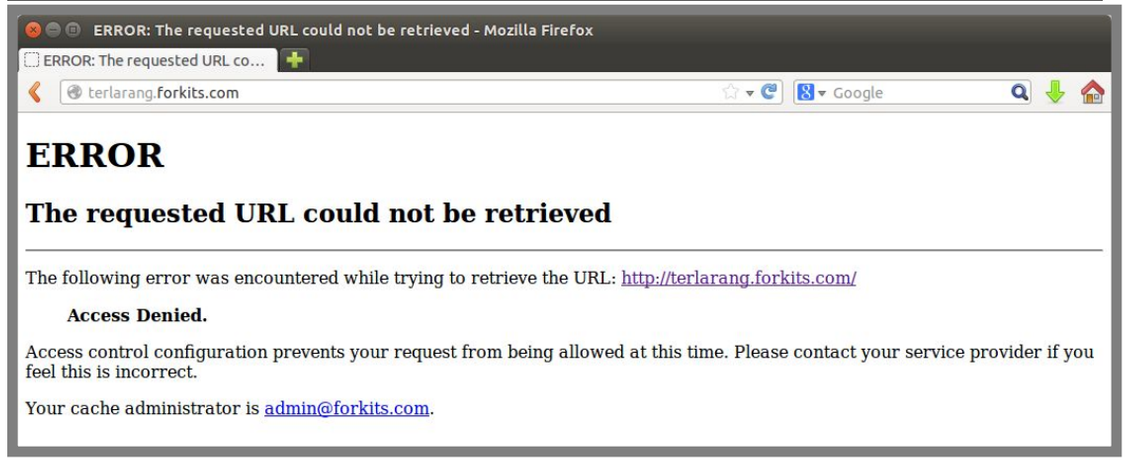
Gambar 17.34 Menjalankan konfigurasi firewall nat

Untuk melakukan pengujian, kita coba untuk menghilangkan konfigurasi manual proxy pada browser milik client



Gambar 17.35 Menonaktifkan konfigurasi manual proxy pada client

Saat ini seharusnya jika client mencoba untuk mengakses website-website terlarang yang diblokir oleh proxy, maka client akan menerima sebuah peringatan *Access Denied* dari proxy



Gambar 17.36 Pengujian dari komputer client

Perhatikan gambar diatas, terlihat bahwa kita tetap akan terblokir oleh proxy meskipun kita tidak melakukan konfigurasi manual proxy pada browser client. Hal ini cukup membuktikan bahwa kita telah membuat sebuah transparent proxy pada server, yang membuat client mau tidak mau dipaksa melewati proxy sebelum mengakses ke internet (web server).

Monitoring Proxy dengan Sarg

Sarg merupkan salah satu aplikasi berbasis web yang fungsinya untuk memantau aktifitas yang dilakukan oleh proxy server. Dengan aplikasi ini, kita bisa mengetahui siapa saja yang menggunakan proxy server, apa saja website yang dikunjungi oleh client, dll.

Berikut perintah yang dapat kita gunakan untuk menginstall aplikasi ini

```
root@Router-Proxy:~# apt-get install sarg
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  squidguard libapache2-mod-php5
The following NEW packages will be installed:
  sarg
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/252 kB of archives.
After this operation, 967 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Gambar 17.37 Installasi sarg untuk monitoring proxy

Setelah selesai melakukan installasi sarg, kita harus menjalankan sarg. Kita bisa menggunakan perintah berikut untuk menjalankan sarg

```
root@Router-Proxy:~# sarg-reports today
SARG: Period covered by log files: 07/06/2016-07/06/2016
root@Router-Proxy:~# sarg-reports daily
root@Router-Proxy:~# sarg-reports weekly
root@Router-Proxy:~# sarg-reports monthly
root@Router-Proxy:~#
```

Gambar 17.38 Menjalankann sarg

Sudah saya katakan sebelumnya, bahwa sarg merupakan aplikasi berbasis web. Jadi sistem kerja dari sarg tidak jauh berbeda dengan sistem kerja dari sebuah CMS. Maka dari itu kita harus membuat virtualhost untuk sarg ini.

```
root@Router-Proxy:~# cd /etc/apache2/sites-available/
root@Router-Proxy:/etc/apache2/sites-available# cp default sarg
root@Router-Proxy:/etc/apache2/sites-available# nano sarg
<VirtualHost *:80>
    ServerAdmin admin@forkits.com
    ServerName proxy.forkits.com
    DocumentRoot /var/lib/sarg
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/lib/sarg/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
```

Gambar 17.39 Konfigurasi virtualhost untuk sarg

Perhatikan gambar diatas, terlihat bahwa ServerName dari virtualhost tersebut adalah http://proxy.forkits.com. Domain ini harus sudah dibuat di web server dengan ip address mengarah ke proxy server. Sehingga nantinya client bisa meresolve domain tersebut.

```
admin@ubuntu:~$ nslookup proxy.forkits.com
Server:         10.10.10.1
Address:        10.10.10.1#53

Name:   proxy.forkits.com
Address: 192.168.10.1

admin@ubuntu:~$
```

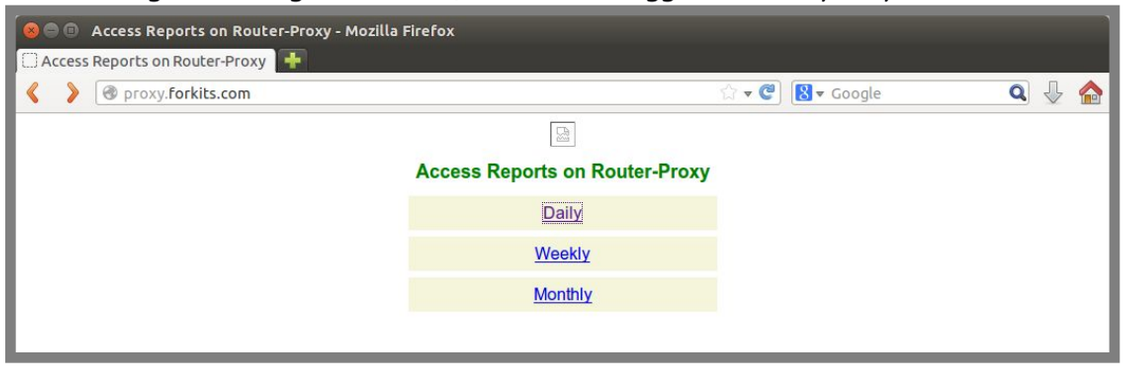
Gambar 17.40 Client bisa meresolve proxy.forkits.com

Perhatikan pula bahwa webdirectory dari virtualhost diatas adalah */var/lib/sarg,* hal ini dikarenakan file-file dari aplikasi sarg berada didirektory tersebut. Selanjutnya jangan lupa enable virtualhost tersebut dan restart apache

```
root@Router-Proxy:/etc/apache2/sites-available# a2ensite sarg
Enabling site sarg.
To activate the new configuration, you need to run:
  service apache2 reload
root@Router-Proxy:/etc/apache2/sites-available# service apache2 restart
[....] Restarting web server: apache2apache2: apache2: Could not reliably
determine the server's fully qualified domain name, using 127.0.0.1 for
ServerName
 ... waiting apache2:apache2: Could not reliably determine the server's fully
qualified domain name, using 127.0.0.1 for ServerName
. ok
root@Router-Proxy:/etc/apache2/sites-available#
```

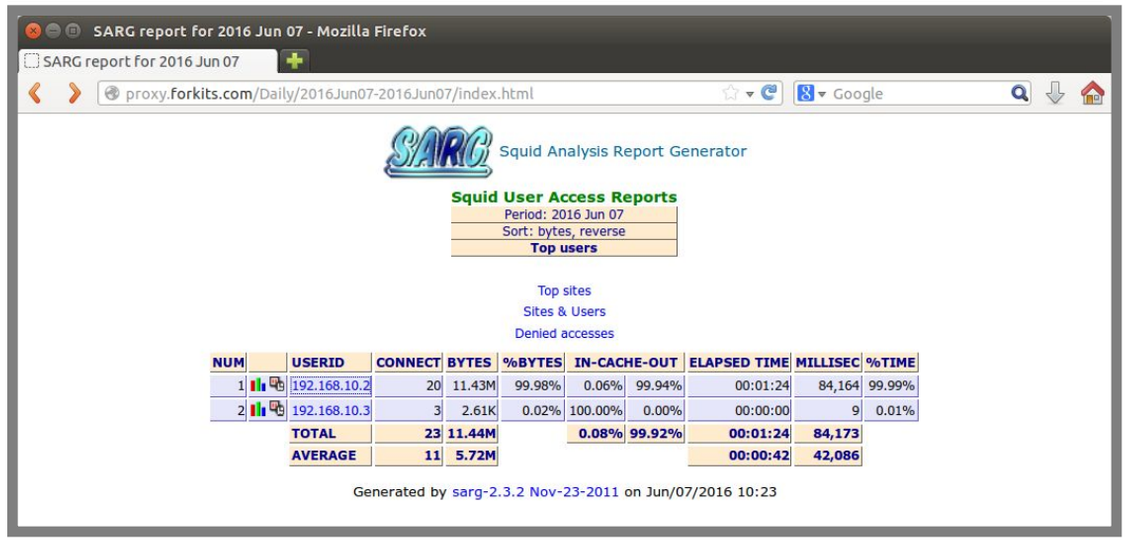
Gambar 17.41 Enable virtualhost untuk sarg dan restart service web server

Untuk mengakses sarg dari client, kita bisa menggunakan url *proxy.forkits.com*



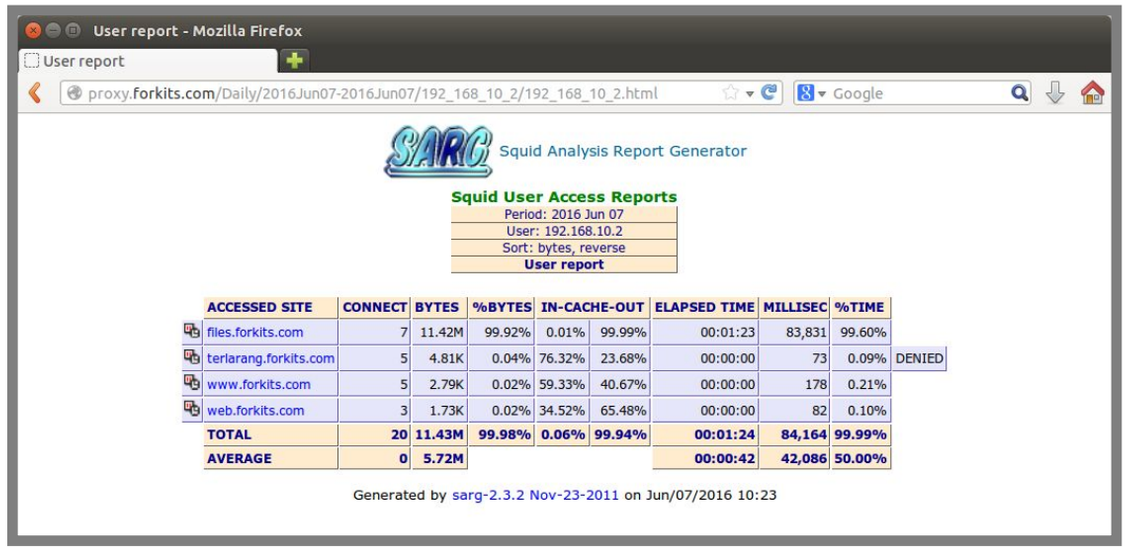
Gambar 17.42 Halaman utama sarg

Karena proxy yang kita konfigurasi masih satu hari, maka kita hanya bisa melihat report pada *Daily* (harian). Perhatikan bahwa hari ini ada dua user yang menggunakan proxy, yaitu user 192.168.10.2 dan 192.168.10.3



Gambar 17.43 Melihat siapa saja yang mengakses client

Sedangkan rincian website yang diakses oleh user 192.168.10.2 adalah sebagai berikut



Gambar 17.44 Melihat website apa saja yang dikunjungi salah satu client

Perhatikan bahwa kita bisa melihat informasi apa saja yang kita inginkan, mulai dari siapa saja yang menggunakan proxy, website apa saja yang diakses oleh user, berapa besar trafic dari masing-masing user, dll.

---END OF CHAPTER---

Bab 18
Linux Firewall

Firewall merupakan sebuah aturann yang berfungsi untuk menentukan paket mana saja yang dapat diterima atau ditolak. Untuk menjalankan fungsi tersebut, firewall akan memeriksa header dari suatu paket data.

Terdapat beberapa header yang pasti ada dalam suatu paket data, diantaranya yang akan sering kita gunakan adalah parameter src-address (ip address pengirim), dst-address (ip address tujuan/penerima), protocol, src-port (port sumber), dan dst-port (port tujuan).

Untuk membuat sebuah kebijakan firewall, entah itu accept (diterima) ataupun drop (ditolak), kita harus membuat rule-rule firewall yang mengandung prameter-parameter yang ada diheader setiap packet, seperti src-address, dst-address, protocol, src-port, dst-port, dll.

Selanjutnya firewall akan membaca rule-rule yang kita buat tadi. Jika ada paket yang cocok dengan rule yang kita buat tadi, maka firewall akan menerapkan kebijakan yang kita konfigurasikan kepada paket tersebut.

Terdapat tiga tabel yang ada pada firewall, yaitu mangle, filter, dan nat. Tabel mangel ditujukan untuk menandai suatu paket. Kita akan jarang menggunakan tabel ini pada debian. Kita akan sangat sering berjumpa dengan tabel ini jika menggunakan device yang memang dihususkan sebagai router, seperti mikrotik.

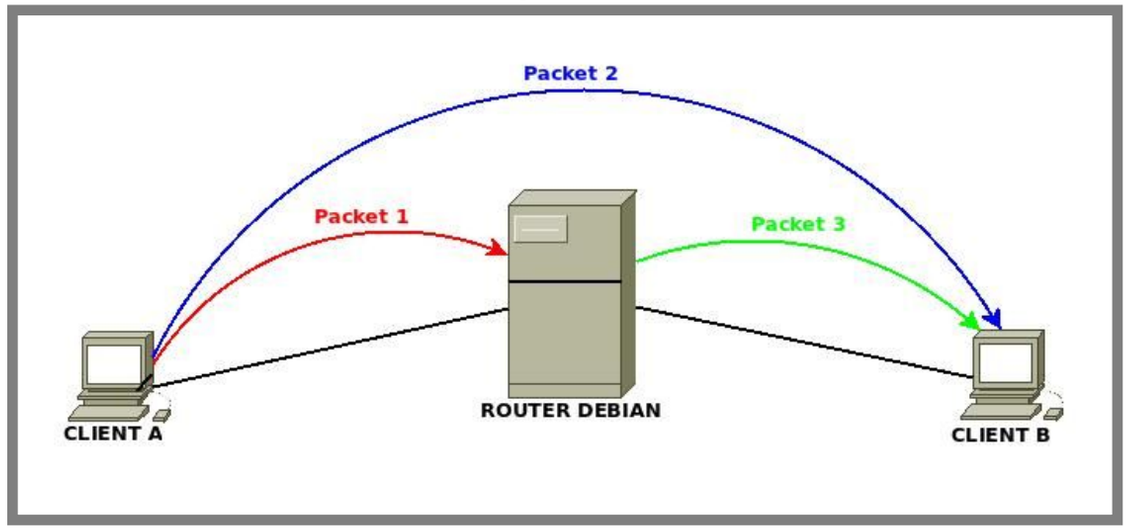
Selanjutnya tabel yang akan sering kita gunakan pada debian adalah tabel filter dan nat. Tabel filter digunakan untuk menentukan paket mana saja yang diaccept (diterima) atau didrop (ditolak). Sedangkan tabel nat digunakan untuk merubah parameter src-address (ip address pengirim) ataupun dst-address (ip address tujuan).

Selanjutnya kita akan membahas tabel filter dan tabel nat lebih mendalam pada sub bab yang berbeda.

Firewall Filter dengan Iptables

Telah disebutkan sebelumnya bahwa firewall filter digunakan untuk menentukan kebijakan terhadap suatu paket, apakah paket tersebut akan diijinkan berjalan pada jaringan (accept) ataukan paket tersebut akan ditolak (drop).

Firewall filter bekerja dengan parameter chain, chain digunakan untuk melihat asal dan tujuan dari paket yang diterima oleh firewall. Secara default, dalam firewall filter terdapat tiga chain, yaitu input, forward, dan output. Perhatikan ilustrasi berikut untuk memahami ketiga chain tersebut



Gambar 18.1 Ilustrasi chain pada firewall filter

Telah disebutkan sebelumnya bahwa chain digunakan untuk melihat asal dan tujuan dari paket yang diterima oleh firewall. Pada gambar 18.1, yang bertindak sebagai firewall adalah Router Debian. Selanjutnya kita akan membahas detail dari masing-masing paket yang diterima oleh firewal (router debian).

Paket 1 adalah paket yang berasal dari client A yang ditujukan untuk router debian, misal paket ping, ssh, http, dll. Paket ini akan ditangani oleh chain input pada firewall (router debian).

Paket 2 adalah paket yang berasal dari client A yang ditujukan oleh client B. Paket ini akan ditangani oleh chain forward pada firewall (router debian).

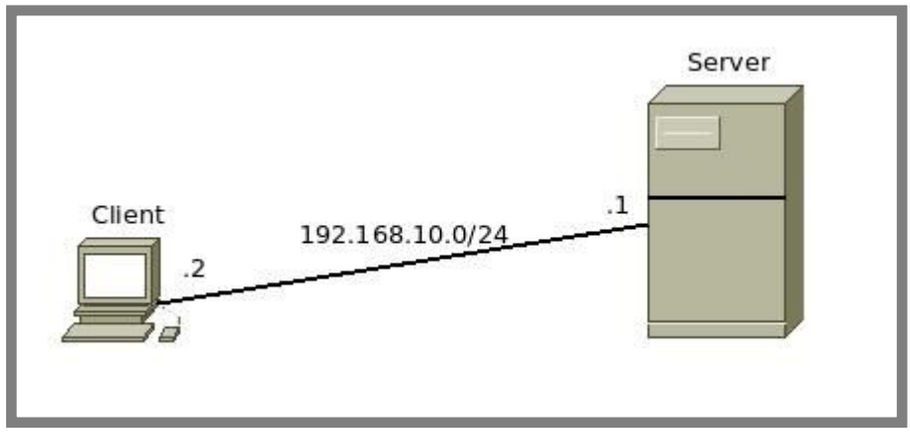
Paket 3 adalah paket yang berasal dari firewall itu sendiri (router debian) yang ditujukan oleh client B. Paket ini akan ditangani oleh chain output pada firewall (router debian).

Selanjutnya untuk lebih memahami ketiga chain tersebut, kita akan langsung praktik mengkonfigurasi firewall filter menggunakan contoh kasus. Namun perlu diketahui sebelumnya bahwa dalam mengkonfigurasikan firewall filter, kita bisa menggunakan dua taktik.

Taktik pertama yaitu membuang beberapa paket kemudian menerima semua paket. Sedangkan taktik kedua adalah menerima beberapa paket yang dibutuhkan kemudian membuang semua paket yang tidak dibutuhkan.

Perlu diketahui juga bahwa firewall filter membaca rule-rule dari atas kebawah, sehingga kita harus benar-benar memperhatikan urutan rule yang kita buat pada firewall filter.

Skenario 1 (input)



Gambar 18.2 Skenario jaringan pertama

Pada skenario ini, kita akan menggunakan firewall filter taktik kedua, yaitu menerima beberapa paket yang dibutuhkan, kemudian membuang semua paket yang tidak dibutuhkan. Dalam hal ini, paket yang dibutuhkan hanyalah paket ping.

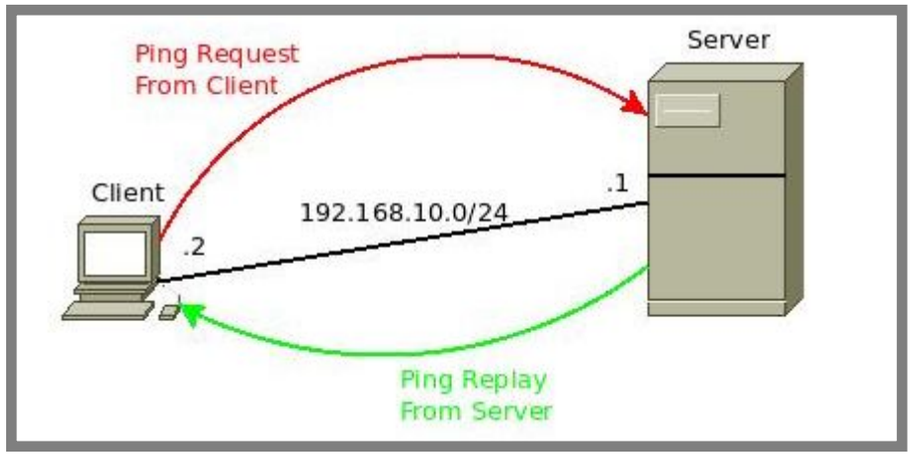
Sehingga nantinya client hanya bisa melakukan ping ke server dan tidak akan bisa melakukan komunikasi apapun ke server selain ping, entah itu remote access ssh, request dns, request web, dll.

Jika ada paket yang berasal dari client menuju server, tentu kita akan menggunakan chain input pada firewall (yang menjadi firewall adalah server).

Selanjutnya kita tentu tahu bahwa packet yang berjalan dalam jaringan mempunyai konsep request & replay. Yaitu jika client melakukan ping ke server (request), maka server akan membalas paket tersebut (reply). Sedangkan paket balasan dari server ke client ditangani oleh chain output.

Sehingga untuk memperbolehkan ping dari client ke server, kita perlu menambahkan rule firewall pada chain input dan output.

Perhatikan ilustrasi proses ping dari client ke server berikut ini



Gambar 18.3 Proses ping dari client ke server

Perhatikan gambar diatas, terlihat bahwa pertama-tama client akan melakukan ping request ke server, paket ini akan ditangani oleh chain input pada server. Setelah server menerima ping request dari client, maka server akan merespon paket tersebut dengan sebuah paket yang dinamakan ping replay, paket ini akan ditangani oleh chain output pada server.

Selanjutnya perhatikan tabel firewall filter pada debian berikut ini.

```
root@forkits:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@forkits:~#
```

Gambar 18.4 Tabel firewall filter

Perintah *iptables -L* dapat kita gunakan untuk melihat tabel firewall filter di debian. Perhatikan bahwa pada tabel filter terdapat tiga chain seperti yang kita telah bahas sebelumnya, yaitu input, forward, dan output.

Perhatikan atribut *policy ACCEPT* pada ketiga chain tersebut. Atribut tersebut disebut dengan default policy. Default policy adalah kebijakan terahir yang akan diterapkan kepada suatu paket jika tidak ada sama sekali rule firewall yang cocok dengan paket tersebut.

Dapat disimpulkan bahwa sistem kerja firewall adalah sebagai berikut: Firewall akan membaca rule-rule yang ada, jika nantinya suatu paket cocok dengan salah satu rule yang ada di firewall, maka firewall akan menerapkan kebijakan sesuai dengan rule yang cocok dengan paket tersebut. Namun jika sampai rule terahir

tetap saja tidak ada rule yang cocok dengan paket tersebut, maka paket tersebut akan dikenakan kebijakan sesuai dengan default policy yang ada.

Jika kita mengkonfigurasi default policy accept, maka artinya kita menggunakan firewall taktik pertama, ingat bahwa firewall taktik pertama adalah memblokir beberapa paket kemudian mengizinkan semua paket.

Sedangkan jika kita mengkonfigurasi default policy menjadi block, maka artinya kita menggunakan firewall taktik kedua, ingat bahwa firewall taktik kedua adalah mengizinkan beberapa paket kemudian memblokir semua paket.

Selanjutnya untuk menerapkan skenario 1, kita akan menggunakan taktik 2, sehingga kita harus merubah atribut policy pada chain input dan output menjadi drop. Kita tidak perlu merubah chain forward, karena kita tidak menggunakan chain forward pada skenario 1.

```
root@forkits:~# iptables -P INPUT DROP
root@forkits:~# iptables -P OUTPUT DROP
root@forkits:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
root@forkits:~#
```

Gambar 18.5 Merubah default policy pada firewall filter

Perhatikan gambar diatas, terlihat bahwa pada chain input dan output, policy nya sudah berubah menjadi drop.

Selanjutnya kita tinggal menambahkan rule firewall yang mengizinkan paket ping request dari client ke server dan juga paket ping replay dari server ke client.

```
root@forkits:~# iptables -A INPUT -s 192.168.10.2 -p icmp -j ACCEPT
root@forkits:~# iptables -A OUTPUT -d 192.168.10.2 -p icmp -j ACCEPT
root@forkits:~# iptables -L
Chain INPUT (policy DROP)
target     prot    opt source          destination
ACCEPT     icmp    --  192.168.10.2    anywhere

Chain FORWARD (policy ACCEPT)
target     prot    opt source          destination

Chain OUTPUT (policy DROP)
target     prot    opt source          destination
ACCEPT     icmp    --  anywhere        192.168.10.2
root@forkits:~#
```

Gambar 18.6 Menambahkan rule pada firewall filter

Perhatikan perintah pertama, digunakan untuk menambkan sebuah rule pada firewall filter chain input dengan parameter source address (-s) 192.168.10.2 dan protocol (-p) icmp dengan actionnya adalah diterima (accept). Rule ini artinya kita memperbolehkan paket ping request (icmp) yang berasal dari client.

Selanjutnya perintah kedua digunakan untuk menambahkan sebuah rule pada firewall filter chain output dengan parameter destination address (-d) 192.168.10.2 dan protocol icmp dengan action accept. Rule ini artinya kita memperbolehkan paket ping replay (icmp) yang berasal dari komputer server yang ditujukan untuk komputer client.

Saat ini, jika suatu saat ada sebuah paket ping request dari client menuju server, maka paket tersebut akan diperiksa oleh firewall filter, apakah ada rule yang cocok dengan paket tersebut, oh ternyata ada! Paket tersebut cocok dengan rule nomor satu dengan action accept, maka paket tersebut akan diterima oleh komputer server.

Kemudian jika suatu saat client mengirimkan paket ssh request ke client, maka paket tersebut juga akan diperiksa oleh firewall, apakah ada rule yang cocok dengan paket tersebut, oh ternyata tidak ada! Maka paket tersebut akan dikenakan kebijakan pada default policy, yaitu drop.

Perhatikan hasil pengujian ping request dan ssh request yang dilakukan oleh client berikut ini

```
admin@ubuntu:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.557 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.496 ms
```

Gambar 18.7 Pengujian ping dari client

Perhatikan gambar diatas, terlihat bahwa client berhasil melakukan ping ke server. Namun client tidak akan bisa melakukan request apapun ke server kecuali request ping tersebut. Berikut bukti saat client tidak bisa melakukan request ssh ke server

```
admin@ubuntu:~$ ssh 192.168.10.1 -l administrator
ssh: connect to host 192.168.10.1 port 22: Connection timed out
admin@ubuntu:~$
```

Gambar 18.8 Client gagal meremote server menggunakan ssh

Perhatikan bahwa client tidak bisa melakukan request ssh ke server, hal ini dikarenakan kita hanya memperbolehkan paket data icmp (ping) dari client ke server.

Selanjutnya tetap menggunakan skenario 1, namun kali ini kita hanya akan mengizinkan paket ssh dari client ke server. Sedangkan rule untuk mengizinkan paket ping yang telah kita buat tadi akan kita hapus. Berikut perintah yang dapat kita gunakan untuk menghapus rule yang ada pada firewall filter

```
root@forkits:~# iptables -D INPUT 1
root@forkits:~# iptables -D OUTPUT 1
root@forkits:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
root@forkits:~#
```

Gambar 18.9 Menghapus rule firewall filter

Perhatikan gambar diatas, terlihat bahwa kita bisa menggunakan option -D untuk menghapus rule yang ada pada firewall filter, diikuti dengan nomor urut dari rule yang ingin kita hapus. Karena pada chain input dan output hanya ada satu rule, maka rule tersebut mempunyai nomor 1 (satu). Perhatikan bahwa saat ini rule-rule yang kita buat tadi sudah tidak ada.

Selanjutnya kita bisa membuat rule yang digunakan untuk mengizinkan paket ssh dari client ke server. Tetap ingat konsep request & replay pada jaringan komputer. Sehingga kita tidak cukup jika hanya menambahkan rule untuk mengizinkan paket ssh dari client ke server saja, kita juga harus menambahkan rule untuk mengizinkan paket dari server ke client.

```
root@forkits:~# iptables -A INPUT -s 192.168.10.2 -p tcp --dport 22 -j
ACCEPT
root@forkits:~# iptables -A INPUT -d 192.168.10.2 -p tcp --sport 22 -j
ACCEPT
root@forkits:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source          destination
ACCEPT     tcp  --  192.168.10.2    anywhere        tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source          destination

Chain OUTPUT (policy DROP)
target     prot opt source          destination
ACCEPT     tcp  --  anywhere        192.168.10.2    tcp spt:ssh
root@forkits:~#
```

Gambar 18.10 Menambahkan rule untuk accept paket ssh

Perhatikan perintah pertama, digunakan untuk menambahkan rule pada firewall filter chain input dengan parameter source address 192.168.10.2, protocol tcp, dan dst-port 22 dengan action accept. Artinya rule ini memperbolehkan paket tcp port 22 (paket ssh) dari komputer client yang ditujukan untuk komputer server.

Selanjutnya perintah kedua digunakan untuk menambahkan rule pada firewall filter chain output dengan parameter destination address 192.168.10.2, protocol tcp, dan src-port 22 dengan action accept. Artinya rule ini memperbolehkan paket tcp port 22 (paket ssh) dari komputer server yang ditujukan untuk komputer client.

Untuk melakukan pengujian, pertama-tama kita akan mencoba melakukan request ping dari komputer client ke server.

```
admin@ubuntu:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
```

Gambar 18.11 Error ping dari komputer client ke server

Perhatikan bahwa tidak akan ada keluaran saat client mencoba ping ke server. Ini artinya client tidak bisa melakukan request paket ping (icmp) ke server. Namun seharusnya komputer client bisa melakukan request paket ssh ke server, perhatikan gambar berikut

```
admin@ubuntu:~$ ssh 192.168.10.1 -l administrator
administrator@192.168.10.1's password: (tak terlihat)
Linux forkits 3.2.0-4-486 #1 Debian 3.2.57-3 i686
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
administrator@forkits:~$
```

Gambar 18.12 Client berhasil melakukan request ssh ke server

Perhatikan gambar diatas, terlihat bahwa client berhasil meremote server menggunakan ssh.

Masih menggunakan topologi pada skenario 1, kita akan coba praktik menggunakan taktik pertama, yaitu menolak beberapa paket, kemudian mengizinkan semua paket. Dalam hal ini kita hanya ingin menolak paket ping yang berasal dari client, selebihnya kita akan mengizinkan semua paket yang berasal dari client.

```
root@forkits:~# iptables -F
root@forkits:~# iptables -P INPUT ACCEPT
root@forkits:~# iptables -P OUTPUT ACCEPT
root@forkits:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@forkits:~#
```

Gambar 18.13 Firewall filter taktik pertama

Perhatikan gambar diatas, perintah pertama digunakan untuk menghapus seluruh rule yang ada pada firewall (disarankan hati-hati dalam menggunakan perintah ini, karena rule yang sudah dihapus tidak akan bisa dikembalikan lagi).

Selanjutnya perintah kedua dan ketiga digunakan untuk merubah policy pada chain input dan output menjadi accept. Hal ini dikarenakan prinsip kerja dari taktik pertama adalah menolak beberapa paket kemudian menerima semua paket. Selanjutnya kita harus membuat rule firewall untuk menolak paket ping (icmp).

```
root@forkits:~# iptables -A INPUT -s 192.168.10.2 -p icmp -j DROP
root@forkits:~# iptables -A OUTPUT -d 192.168.10.2 -p icmp -j DROP
root@forkits:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot    opt source          destination
DROP       icmp    --  192.168.10.2    anywhere

Chain FORWARD (policy ACCEPT)
target     prot    opt source          destination

Chain OUTPUT (policy ACCEPT)
target     prot    opt source          destination
DROP       icmp    --  anywhere        192.168.10.2
root@forkits:~#
```

Gambar 18.14 Menambahkan rule untuk drop paket icmp

Perhatikan bahwa saat ini sudah ada sebuah rule yang menolak paket icmp dari client ke server maupun dari server ke client. Sedangkan default policy pada chain input dan output adalah accept.

Sehingga jika suatu saat client mengirimkan paket ping (icmp) ke server, maka paket tersebut akan diperiksa oleh firewall, apakah ada rule yang cocok dengan paket tersebut, oh ternyata ada! Paket ini cocok dengan rule nomor 1, maka paket ini akan dikenakan kebijakan sesuai dengan rule tersebut, yaitu drop.

Begitu juga saat client mencoba melakukan request dns ke server, maka paket tersebut akan diperiksa oleh firewall, apakah ada rule yang cocok dengan paket tersebut, oh ternyata tidak ada! Maka paket tersebut akan dikenakan kebijakan sesuai dengan default policy yang ada, yaitu accept.

Perhatikan pengujian berikut yang menunjukkan client tidak bisa melakukan ping request ke server

```
admin@ubuntu:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
```

Gambar 18.15 Client gagal melakukan request icmp ke server

Perhatikan gambar diatas, terlihat bahwa client gagal melakukan ping ke komputer server. Namun seharusnya client bisa melakukan request dns ke server, perhatikan gambar berikut

```
admin@ubuntu:~$ cat /etc/resolv.conf
nameserver 192.168.10.1
admin@ubuntu:~$ nslookup forkits.com
Server:         192.168.10.1
Address:        192.168.10.1#53

Name:   forkits.com
Address: 192.168.10.1

admin@ubuntu:~$
```

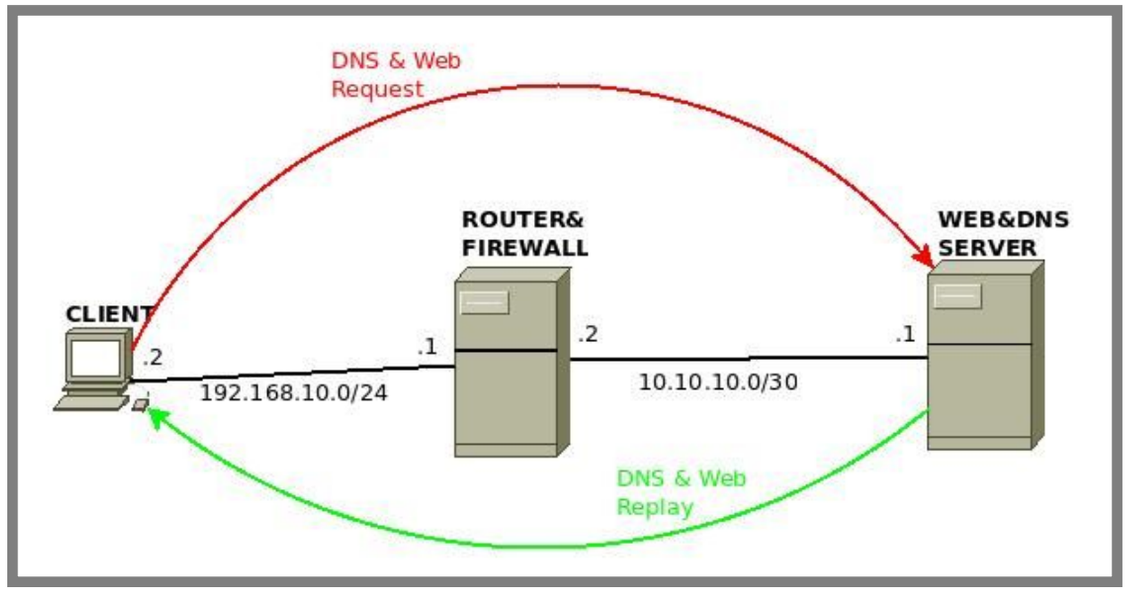
Gambar 18.16 Client berhasil melakukan request dns ke server

Perhatikan gambar diatas, terlihat bahwa client berhasil melakukan request dns ke komputer server. Selanjutnya perhatikan gambar berikut

Gambar 18.17 Client berhasil melakukan request http ke server

Perhatikan gambar diatas, terlihat bahwa client berhasil melakukan request http ke server.

Skenario 2 (forward)



Gambar 18.18 Topologi jaringan skenario 2

Pada skenario 2 ini, kita akan belajar mengkonfigurasi firewall filter dengan chain forward. Perhatikan topologi diatas, terlihat bahwa terdapat suatu paket dns & web request dari client menuju server, selain itu juga ada paket dns & web replay dari server menuju client. Tentunya kedua paket ini akan ditangani oleh firewall filter chain forward oleh komputer router (router & firewall).

Pada skenario ini diasumsikan bahwa client hanya diperbolehkan melakukan request dns dan web kepada server, sehingga nantinya client tidak akan bisa melakukan ping ke server, namun bisa mengakses website milik server dari web browser.

Dari skenario diatas, kita bisa menyimpulkan bahwa taktik yang paling efektif untuk kita gunakan adalah taktik kedua, yaitu mengizinkan beberapa paket kemudian menolak semua paket.

Dalam hal ini kita bisa merubah default policy menjadi drop, kemudian menambahkan beberapa rule pada chain forward, yaitu rule untuk paket request dari client dan rule untuk paket replay dari server).

Untuk lebih jelasnya, kita akan langsung praktik konfigurasi firewall pada komputer router. Namun sebelumnya diasumsikan bahwa komputer router dan server sudah dikonfigurasi sesuai dengan topologi diatas. Mulai dari ip address, hingga service yang dibutuhkan (fungsi router, dns server, web server, dll).

Sebelumnya perlu diketahui bahwa dns berjalan pada protocol udp dan tcp port 53, sedangkan web (http) berjalan pada protocol tcp port 80.

Berikut konfigurasi firewall yang perlu kita lakukan pada komputer router

```
root@Router:~# iptables -P FORWARD DROP
root@Router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@Router:~#
```

Gambar 18.19 Merubah default policy forward

Perhatikan gambar diatas, terlihat bahwa kita merubah default policy pada chain forward menjadi drop. Selanjutnya kita harus menambahkan rule-rule firwall untuk memperbolehkan paket dns & web request dari client

```
root@Router:~# iptables -A FORWARD -s 192.168.10.2 -d 10.10.10.1 -p
udp --dport 53 -j ACCEPT
root@Router:~# iptables -A FORWARD -s 192.168.10.2 -d 10.10.10.1 -p
tcp --dport 53 -j ACCEPT
root@Router:~# iptables -A FORWARD -s 192.168.10.2 -d 10.10.10.1 -p
tcp --dport 80 -j ACCEPT
root@Router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  192.168.10.2         10.10.10.1    udp dpt:domain
ACCEPT     tcp  --  192.168.10.2         10.10.10.1    tcp dpt:domain
ACCEPT     tcp  --  192.168.10.2         10.10.10.1    tcp dpt:http

Chain OUTPUT (policy ACCEPT)
```

Gambar 18.20 Menambahkan rule untuk paket request dari client

Langkah diatas masih menambahkan rule untuk paket request dari client, selanjutnya kita harus menambahkan rule untuk paket replay dari server. Berikut perintah yang dapat kita gunakan

```
root@Router:~# iptables -A FORWARD -s 10.10.10.1 -d 192.168.10.2 -p
udp --sport 53 -j ACCEPT
root@Router:~# iptables -A FORWARD -s 10.10.10.1 -d 192.168.10.2 -p
tcp --sport 53 -j ACCEPT
root@Router:~# iptables -A FORWARD -s 10.10.10.1 -d 192.168.10.2 -p
tcp --sport 80 -j ACCEPT
root@Router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  192.168.10.2         10.10.10.1      udp dpt:domain
ACCEPT     tcp  --  192.168.10.2         10.10.10.1      tcp dpt:domain
ACCEPT     tcp  --  192.168.10.2         10.10.10.1      tcp dpt:http
ACCEPT     tcp  --  10.10.10.1           192.168.10.2    udp spt:domain
ACCEPT     tcp  --  10.10.10.1           192.168.10.2    tcp spt:domain
ACCEPT     tcp  --  10.10.10.1           192.168.10.2    tcp spt:http

Chain OUTPUT (policy ACCEPT)
```

Gambar 18.21 Menambahkan rule untuk paket replay dari server

Sampai saat ini kita sudah selesai melakukan konfigurasi firewall di komputer router. Perlu diperhatikan bahwa pada skenario ini, kita sama sekali tidak melakukan konfigurasi firewall pada komputer server. Berikut hasil saat client mencoba melakukan ping ke server

```
admin@ubuntu:~$ ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
```

Gambar 18.22 Error saat client ping ke server

Perhatikan bahwa kita tidak bisa melakukan ping dari client ke server, namun kita seharusnya bisa melakukan request dns dan web dari client ke server

```
admin@ubuntu:~$ cat /etc/resolv.conf
nameserver 10.10.10.1
admin@ubuntu:~$ nslookup forkits.com
Server:         10.10.10.1
Address:        10.10.10.1#53

Name:   forkits.com
Address: 10.10.10.1

admin@ubuntu:~$
```

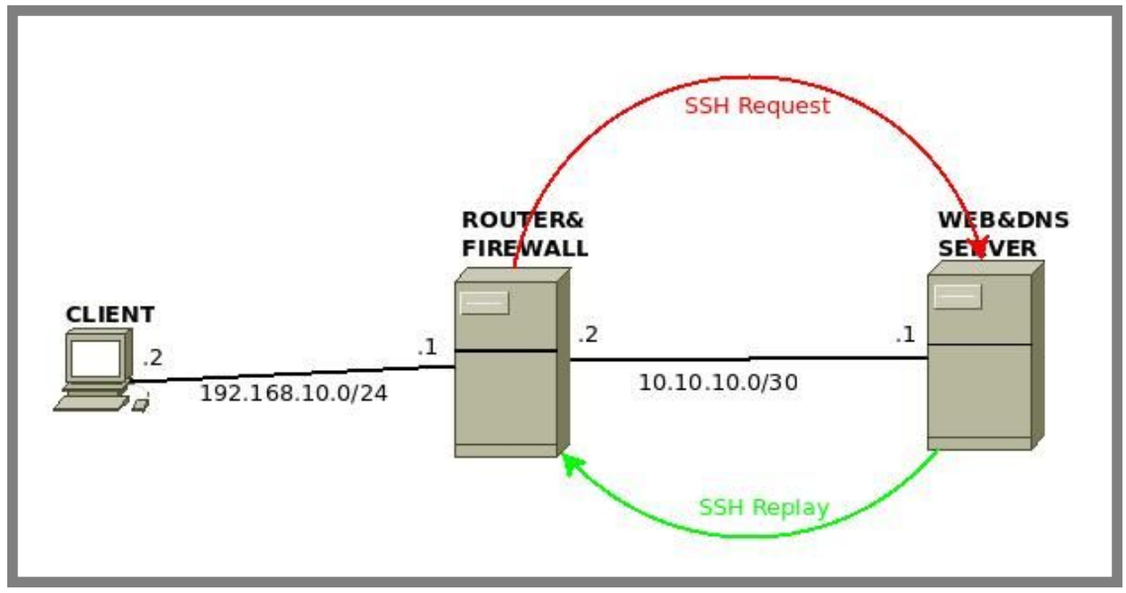
Gambar 18.23 Client berhasil melakukan request dns ke server

Perhatikan gambar diatas, terlihat bahwa kita berhasil melakukan request dns dari client ke server, meskipun kita tidak bisa ping dari client ke server. Selanjutnya kita coba request http dari client ke server

Gambar 18.24 Client berhasil melakukan request http ke server

Perhatikan gambar diatas, terlihat bahwa kita juga berhasil melakukan request http dari client ke server. Hal ini cukup membuktikan bahwa kita sudah berhasil mengkonfigurasi firewall sesuai dengan skenario yang ada.

Skenario 3 (output)



Gambar 18.25 Topologi jaringan skenario 3

Kita akan praktik menggunakan topologi jaringan yang sama dengan skenario 2. Namun skenario kerjanya akan sedikit berbeda.

Pada skenario 3 ini, tugas kita adalah mengkonfigurasi firewall pada router agar nantinya router tidak bisa melakukan ping ke server, namun router bisa melakukan request ssh, dns, web, dll ke server.

Dari skenario diatas, kita bisa menyimpulkan bahwa taktik yang paling efektif untuk digunakan adalah taktik pertama, yaitu menolak beberapa paket kemudian menerima semua paket.

Untuk menerapkan skenario tersebut, kita akan bekerja pada chain output dan input. Chain output akan menangani paket request dari router ke server, sedangkan chain input akan menangani paket replay dari server ke router.

Kita tentu sudah tahu bahwa default policy pada chain output dan input adalah accept. Default policy ini sudah cocok dengan taktik yang akan kita gunakan, yaitu taktik pertama. Sehingga kita tidak perlu melakukan perubahan pada default policy.

Kita hanya perlu menambahkan rule pada chain output dan input untuk memblokir paket ping dari router ke server dan dari server ke router. Kita bisa menggunakan perintah berikut

```
root@Router:~# iptables -A OUTPUT -d 10.10.10.1 -p icmp -j DROP
root@Router:~# iptables -A INPUT -s 10.10.10.1 -p icmp -j DROP
root@Router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp  --  10.10.10.1          anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     tcp --   192.168.10.2         10.10.10.1       udp dpt:domain
ACCEPT     tcp --   192.168.10.2         10.10.10.1       tcp dpt:domain
ACCEPT     tcp --   192.168.10.2         10.10.10.1       tcp dpt:http
ACCEPT     tcp --   10.10.10.1           192.168.10.2     udp spt:domain
ACCEPT     tcp --   10.10.10.1           192.168.10.2     tcp spt:domain
ACCEPT     tcp --   10.10.10.1           192.168.10.2     tcp spt:http

Chain OUTPUT (policy ACCEPT)
target     prot opt source                 destination
DROP       icmp --  anywhere               10.10.10.1
root@Router:~#
```

Gambar 18.26 Menambahkan rule untuk drop paket icmp

Perhatikan gambar diatas, terlihat bahwa kita menambahkan satu rule pada chain output untuk menangani paket icmp dari router ke server dan satu rule pada chain input untuk menangani paket icmp dari server ke router.

Untuk pengujian, pertama kita akan mencoba melakukan ping dari router ke server

```
root@Router:~# ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
```

Gambar 18.27 Error ping dari router ke server

Perhatikan gambar diatas, terlihat bahwa kita tidak bisa melakukan ping dari router ke server. Selanjutnya kita akan mencoba melakukan request dns dari router ke server

```
root@Router:~# cat /etc/resolv.conf
nameserver 10.10.10.1
root@Router:~# nslookup forkits.com
Server:         10.10.10.1
Address:        10.10.10.1#53

Name:   forkits.com
Address: 10.10.10.1

root@Router:~#
```

Gambar 18.28 Router berhasil request dns ke server

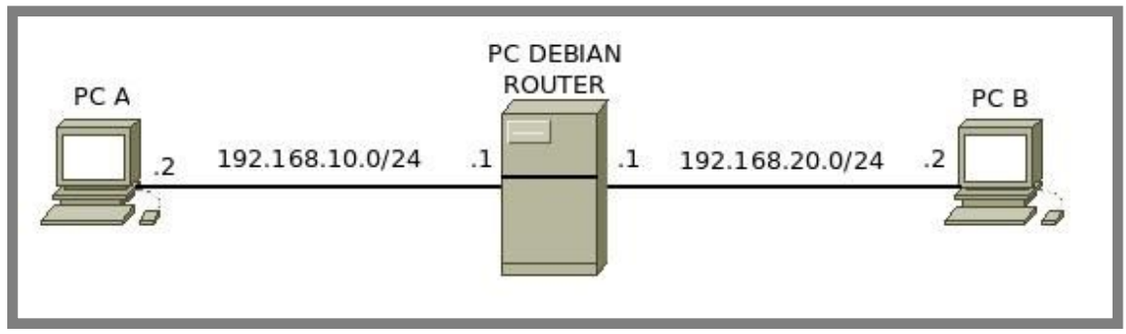
Perhatikan bahwa kita berhasil melakukan request dns dari router ke server. Meskipun router tidak bisa ping ke server.

Firewall NAT dengan Iptables

Firewall NAT merupakan sebuah aturan yang berfungsi untuk merubah ip address sumber maupun ip address tujuan. Tidak sebatas pada ip address saja, namun kita juga dapat merubah port sumber dan juga port tujuan menggunakan firewall nat ini.

Terdapat dua chain dalam firewall nat, yaitu chain src-nat dan dst-nat. Chain src-nat digunakan untuk merubah ip address pengirim maupun port pengirim. Sedangkan chain dst-nat digunakan untuk merubah ip address tujuan maupun port tujuan.

Untuk lebih memahami konsep firewall nat, pertama-tama perhatikan ilustrasi berikut.



Gambar 18.29 Penerapan routing

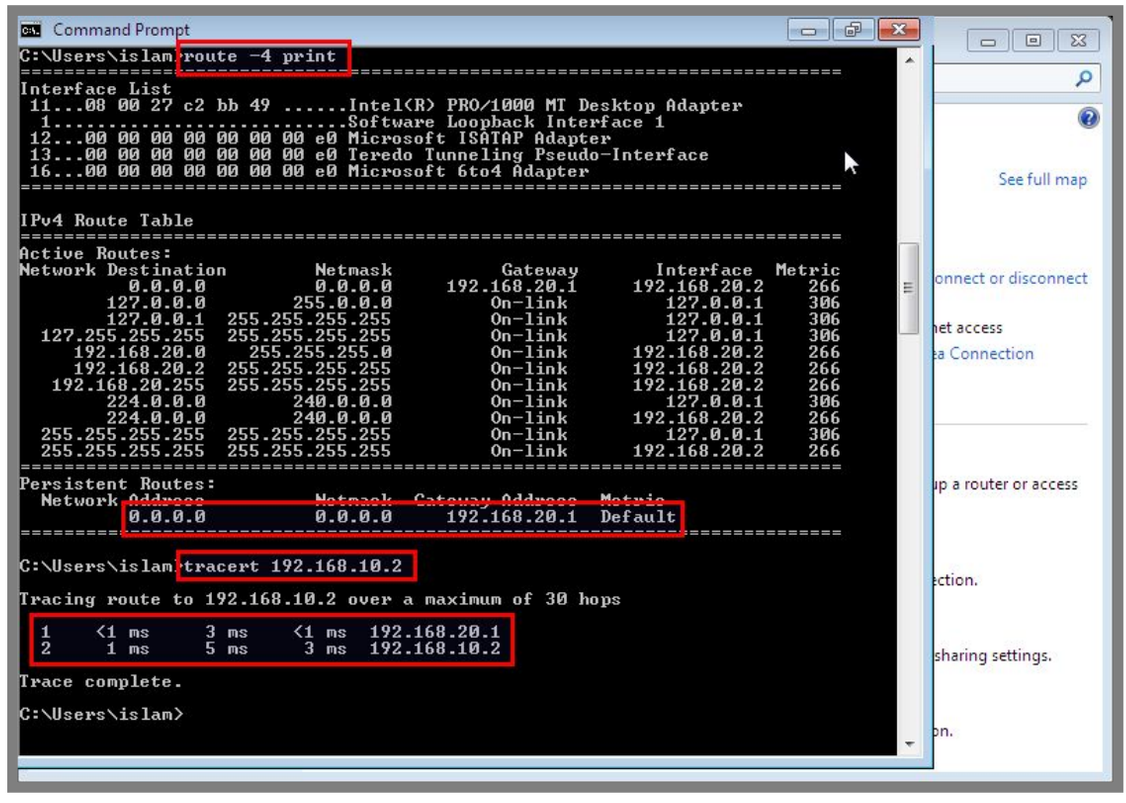
Gambar diatas merupakan topologi jaringan yang kita gunakna pada pembahasan materi routing (gambar 14.2). Saya sarankan teman-teman untuk membaca kembali bab tentang konfigurasi router jika masih belum terlalu memahami konsep routing.

Pada bab 14 (bab konfigurasi router), kita melakukan konfigurasi gateway pada kedua komputer client, yaitu PC A dan PC B (ditunjukkan pada gambar 14.11 dan gambar 14.12).

Sehingga setelah melakukan konfigurasi gateway pada kedua komputer, maka PC A akan mengenal PC B, begitu juga PC B juga akan mengenal PC A. Perhatikan gambar 18.30 dan gambar 18.31 berikut

```
admin@ubuntu:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.10.1    0.0.0.0         UG    0      0        0 vboxnet0
192.168.10.0    *               255.255.255.0   U     0      0        0 vboxnet0
admin@ubuntu:~$ traceroute 192.168.20.2
traceroute to 192.168.20.2 (192.168.20.2), 30 hops max, 60 byte packets
 1  192.168.10.1 (192.168.10.1)  0.520 ms  0.498 ms  0.488 ms
 2  192.168.20.2 (192.168.20.2) 2.539 ms  *         *
admin@ubuntu:~$
```

Gambar 18.30 Traceroute dari PC A ke PC B

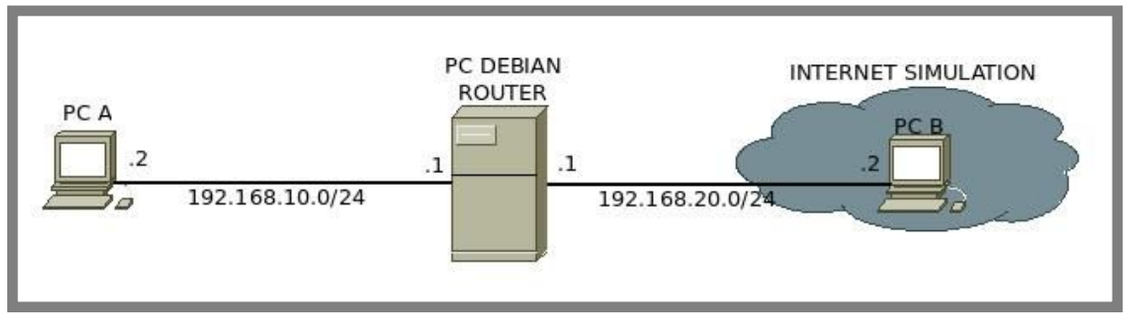


Gambar 18.31 Traceroute dari PC B ke PC A

Note : Kedua gambar diatas diambil setelah melakukan konfigurasi ip address pada PC A dan PC B pada bab 14 (gambar 14.11 dan gambar 14.12).

Perhatikan kedua gambar diatas, terlihat bahwa kedua komputer telah mengetahui keberadaan remote networknya. Yaitu PC A telah mengetahui bahwa untuk mencapai PC B dia harus melewati gateway 192.168.10.1, begitu juga PC B juga telah mengetahui bahwa untuk mencapai PC A dia harus melewati gateway 192.168.20.1.

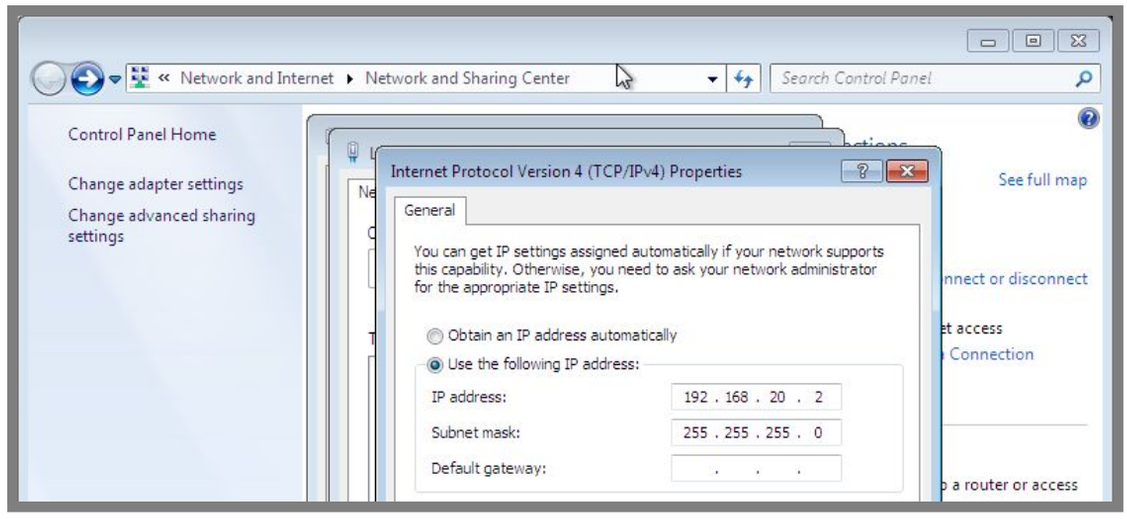
Penjelasan diatas adalah konsep routing, yaitu pada PC A dan PC B dikonfigurasikan gateway agar mengetahui remote networknya masing-masing. Namun dalam penerapan di dunia internet, ada kalanya bahwa pada PC B tidak boleh dikonfigurasikan gateway. Perhatikan ilustrasi berikut



Gambar 18.32 Simulasi konsep NAT

Perhatikan gambar diatas, terlihat bahwa PC B berada pada sebuah cloud yang menandakan internet simulation. Sehingga nantinya kita tidak boleh melakukan konfigurasi gateway pada PC B. Hal ini dikarenakan dunia internet tidak akan mengetahui keberadaan local network yang berada di belakang router kita. Dunia internet hanya akan mengenal ip public yang kita konfigurasikan pada interface yang mengarah ke internet. Dalam hal ini, yang menjadi ip public pada router kita adalah 192.168.20.1. Sehingga dapat disimpulkan bahwa yang bisa melakukan ping ke internet (PC B) hanyalah pc router, sedangkan PC A tidak akan bisa krane internet (PC B) tidak mengetahui keberadaan jaringan local (PC A) yang berada di belakang pc router.

Untuk membuktikannya, kita akan menghapus konfigurasi gateway pada PC B



Gambar 18.33 Menghapus gateway pada PC B

Saat ini seharusnya yang bisa melakukan ping ke internet (PC B) hanya pc router saja. Perhatikan pengujian ping dari PC A dan pc router berikut

```
admin@ubuntu:~$ ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
```

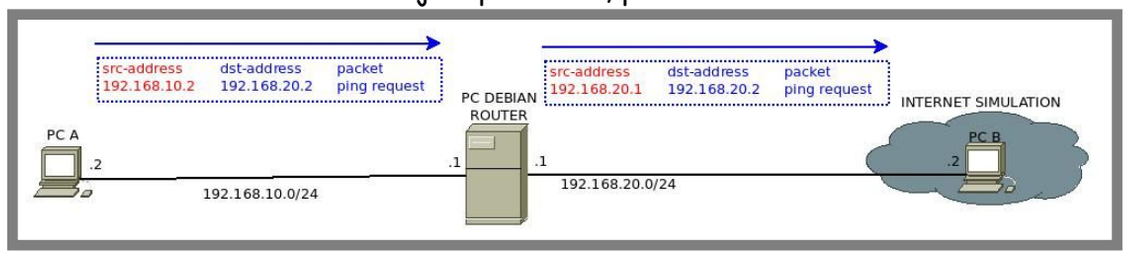
Gambar 18.34 Ping dari PC A ke PC B

```
root@forkits:~# ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_req=1 ttl=128 time=4.02 ms
64 bytes from 192.168.20.2: icmp_req=2 ttl=128 time=1.34 ms
64 bytes from 192.168.20.2: icmp_req=3 ttl=128 time=2.20 ms
```

Gambar 18.35 Ping dari pc router ke PC B

Perhatikan kedua gambar diatas, terlihat bahwa PC A gagal melakukan ping ke PC B sedangkan pc router berhasil melakukan ping ke PC B. Hal ini dikarenakan bahwa PC B hanya mengetahui keberadaan ip public milik router, PC B tidak mengetahui keberadaan jaringan local yang berada di belakang router.

Nah, sekarang bagaimana caranya agar PC A bisa ping ke PC B?? Kita akan menggunakan NAT dengan chain src-nat. Kita akan melakukan perubahan ip address sumber dari PC A menjadi pc router, perhatikan ilustrasi berikut



Gambar 18.36 Konsep src-nat

Perhatikan gambar diatas, terlihat bahwa saat PC A melakukan ping request ke PC B, maka parameter-parameternya adalah sebagai berikut

- Src-address : 192.168.10.2
- Dst-address : 192.168.20.2
- Packet : ping request

Namun setelah paket tersebut melewati router, maka parameter-parameternya akan berubah menjadi seperti ini

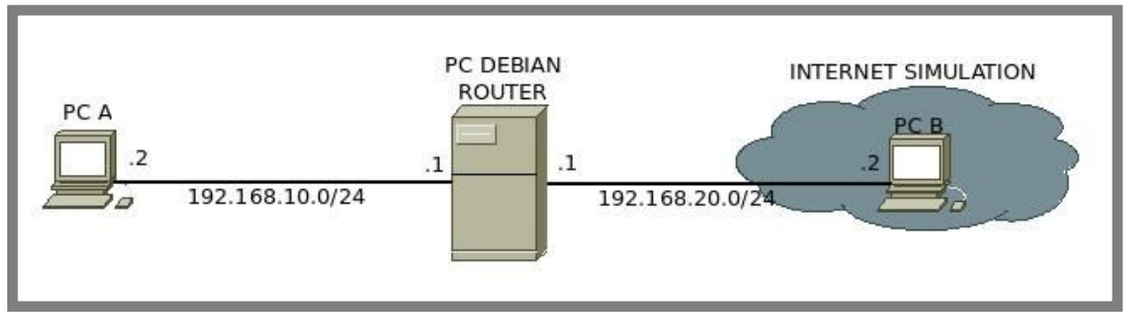
- Src-address : 192.168.20.1
- Dst-address : 192.168.20.2
- Packet : ping request

Perhatikan bahwa router akan melakukan perbuahan parameter src-address pada paket tersebut. Parameter src-address yang mulanya adalah ip address PC A, setelah melewati router, maka src-address akan berubah menjadi ip public milik pc router. Sehingga nantinya PC B akan mengetahui bahwa yang melakukan ping ke dirinya adalah pc router meskipun sebenarnya yang melakukan ping adalah PC A. Hal ini dikarenak pc router telah melakukan manipulasi parameter src-address pada paket tersebut.

Itulah konsep dari NAT, yaitu melakukan perubahan ip address, entah itu ip address sumber (src-nat) maupun ip address tujuan (dst-nat). Selanjutnya kita akan membahas kedua chain tersebut pada sub bab yang berbeda.

Skenario 1 (src-nat)

Untuk prakti pada skenario pertama ini, kita akan menggunakan topologi yang telah kita gunakan untuk membahas konsep NAT diatas,



Gambar 18.37 Topologi skenario 1

Untuk belajar src-nat pada sub bab ini, diasumsikan bahwa pc router telah dikonfigurasi sebagai router seperti pada bab 14. Begitu juga denga PC A dan PC B, kedua PC tersebut juga harus sudah dikonfigurasi ip address sesuai topologi. Hanya saja pada PC A harus dikonfigurasi gateway 192.168.10.1 sedangkan pada PC B kita tidak boleh mengkonfigurasi gateway.

Selanjutnya kita hanya perlu membuat sebuah rule pada firewall nat chain src-nat untuk melakukan perubahan ip address sumber. Kita bisa menggunakan perintah seperti berikut

```
root@forkits:~# iptables -t nat -A POSTROUTING -s 192.168.10.2 -j
SNAT --to 192.168.20.1
root@forkits:~#
```

Gambar 18.37 Konfigurasi src-nat

Pada gambar diatas, terlihat bahwa kita menambahkan sebuah rule pada firewall nat dengan parameter src-address=192.168.10.2 dan dengan actionnya adalah SNAT ke ip address 192.168.20.1. Ini artinya kita menginginkan agar jika suatu

saat ada paket yang berasal dari 192.168.10.2 maka src-addressnya akan dirubah menjadi 192.168.20.1

Untuk melihat rule apa saja yang ada di firewall nat, kita bisa menggunakan perintah berikut

```
root@forkits:~# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  192.168.10.2         anywhere        to:192.168.20.1
root@forkits:~#
```

Gambar 18.38 Melihat tabel nat

Sampai saat ini, seharusnya PC A sudah bisa ping ke PC B, berikut pengujian yang dilakukan

```
admin@ubuntu:~$ ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=127 time=57.6 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=127 time=5.41 ms
64 bytes from 192.168.20.2: icmp_seq=3 ttl=127 time=6.38 ms
```

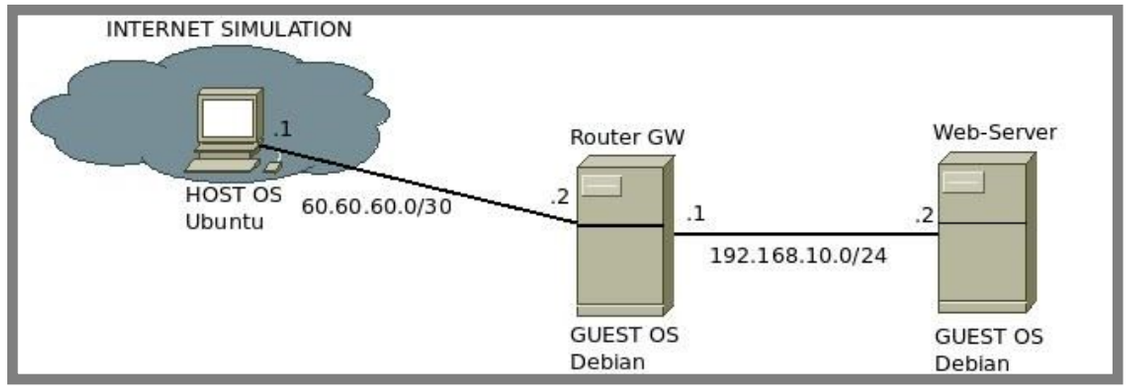
Gambar 18.39 PC A berhasil melakukang ping ke PC B

Perhatikan gambar diatas, terlihat bahwa memang benar jika PC A sudah bisa melakukan ping ke PC B. Hal ini cukup membuktikan bahwa kita berhasil melakukan konfigurasi src-nat pada pc router.

Skenario 2 (dst-nat)

Pada skenario ini, kita akan belajar melakukan konfigurasi firewall nat dengan chain dst-nat. Telah dijelaskan sebelumnya bahwa chain dst-nat digunakan untuk melakukan perubahan pada paremeter dst-address.

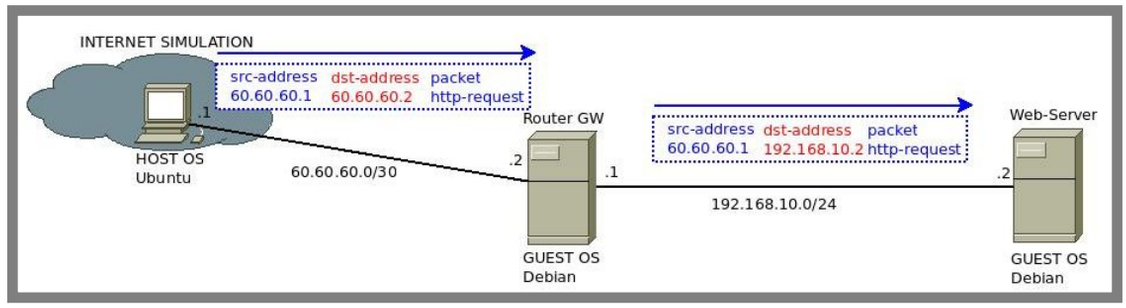
Berikut topologi jaringa yang akan kita gunakan pada skenario ini



Gambar 18.40 Topologi jaringan skenario 2

Perhatikan gambar diatas, kita bisa tahu bahwa internet (host os ubuntu) tidak akan bisa mengakses web server. Hal ini dikarenakan web server berada pada jaringan local dibelakang router gw. Namun bagaimana jika kita ingin agar website yang ada di web server bisa diakses dari internet?? Ada beberapa hal yang perlu kita lakukan untuk mewujudkan hal tersebut.

Hal pertama yang harus kita lakukan adalah memetakan domain yang dimiliki web server ke ip public dari router gw. Selanjutnya kita harus membuat sebuah rule firewall nat dengan chain dst-nat, agar jika suatu saat ada paket http-request dengan parameter dst-address 60.60.60.2 (ip public milik router gw), maka router akan merubah parameter dst-address menjadi 192.168.10.2 (ip address web server). Untuk lebih jelasnya, perhatikan ilustrasi berikut



Gambar 18.41 Konsep implementasi dst-nat

Kita dapat melihat pada gambar diatas bahwa saat komputer client pada internet simulaiton melakukan http-request, parameter dst-address nya adalah 60.60.60.2. Hal ini menunjukkan bahwa paket tersebut ditujukan untuk router gw. Namun pada router gw, ditanamkan sebuah rule firwall nat dengan chain dst-nat untuk merubah parameter dst-address menjadi 192.168.10.2, yaitu ip address web server.

Pada skenario ini, kita asumsikan bahwa web server sudah dikonfigurasi ip address sesuai topologi, jangan lupa untuk mengkonfigurasi gateway ke 192.168.10.1. Diasumsikan pula bahwa router gw sudah dikonfigurasi sebagai router gateway sehingga web server sudah bisa melakukan ping ke internet simulation (lihat sub bab konfigurasi router gateway pada bab 14).

Pada skenario ini, kita hanya akan fokus pada paket http-request. Sehingga kita tidak akan menggunakan domain (dns server) pada skenario ini. Nantinya untuk mengakses website, kita hanya akan menggunakan ip address.

Jika ausmsi-asumsi diatas sudah terpenuhi, selanjutnya kita hanya tinggal menambahkan sebuah rule firewall pada firewall nat dengan chain dst-nat agar jika suatu saat ada paket http-request dengan parameter dst-address 60.60.60.2, maka paket tersebut akan diteruskan ke 192.168.10.2 (dirubah parameter dst-addres nya). Berikut perintah yang dapat kita gunakan

```
root@Router-GW:~# iptables -t nat -A PREROUTING -p tcp -d
60.60.60.2 --dport 80 -j DNAT --to 192.168.10.2:80
root@Router-GW:~#
```

Gambar 18.42 Menambahkan rule dst-nat pada router-gw

Setelah melakukan langkah diatas, seharusnya saat ini website milik web server sudah bisa diakses dari internet simulation menggunakan ip public milik router gw, perhatikan hasil pengujian berikut

Gambar 18.43 Pengujian dari internet simulation

Perhatikan gambar diatas, terlihat bahwa website milik web server sudah bisa diakses menggunakan ip address 60.60.60.2 (ip public milik router-gw).

Menyimpan Konfigurasi Iptables

Sampai saat ini, konfigurasi iptables yang kita lakukan hanya bersifat sementara. Artinya jika suatu saat komputer direstart, maka konfigurasi iptables akan hilang. Karena itu, kita perlu belajar bagaimana agar konfigurasi iptables yang telah kita lakukan tidak hilang walau komputer direstart.

Pada sub bab ini kita tidak akan menggunakan topologi jaringan tertentu, karena kita hanya akan belajar bagaimana cara menyimpan rule-rule iptables yang ada pada suatu server. Diasumsikan bahwa pada suatu server terdapat beberapa rule firewall, maka kita bisa melakukan langkah-langkah berikut untuk menyimpannya

```
root@Router-Proxy:~# iptables-save >> /etc/iptables.conf
root@Router-Proxy:~#
```

Gambar 18.44 Menyimpan rule-rule iptables

Perintah diatas digunakan untuk menyimpan rule-rule iptables pada file */etc/iptables.conf*. Selanjutnya kita harus melakukan konfigurasi pada */etc/rc.local* agar rule-rule firewall yang telah kita simpan tadi bisa diload saat komputer pertama kali dinyalakan

```
root@Router-Proxy:~# nano /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
iptables-restore < /etc/iptables.conf
exit 0
```

Gambar 18.45 Konfigurasi agar rule-rule iptables diload saat komputer booting

Sampai saat ini seharusnya rule-rule firewall tidak akan hilang walau komputer server direstart.

---END OF CHAPTER---

Bab 19
Redundant Array of Independent Disk

Redundant Array of Independent Disk (RAID) adalah sebuah teknologi penyimpanan yang menggunakan kombinasi beberapa harddisk yang digabung menjadi satu dengan tujuan meningkatkan kapasitas penyimpanan dan juga untuk fungsi backup. Artinya jika suatu saat ada data pada suatu harddisk yang hilang, maka data tersebut masih tersimpan pada harddisk yang lain, sehingga kita bisa melakukan resotre data tersebut.

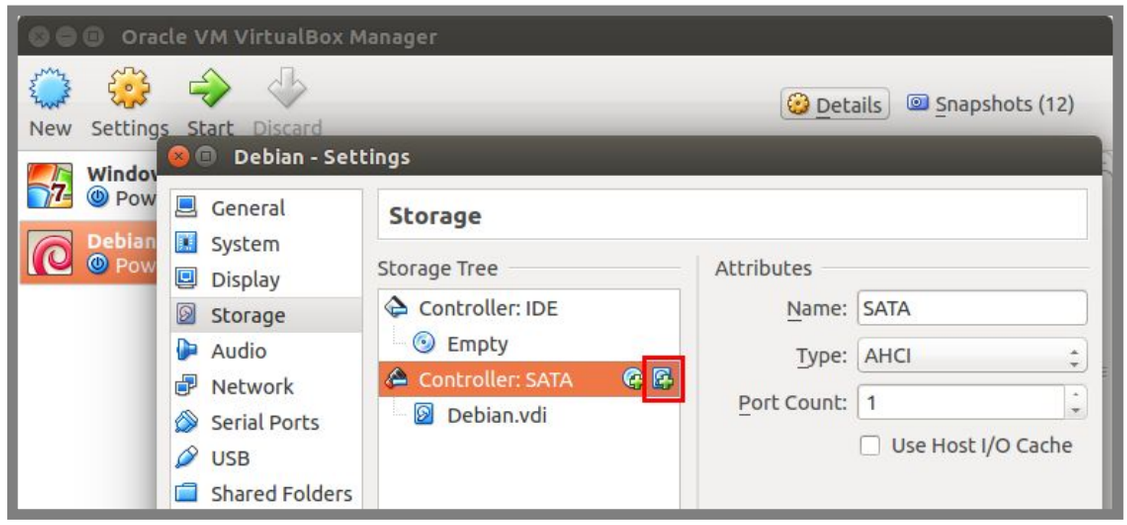
Terdapat beberapa level pada RAID yang umum digunakan, masing-masing level mempunyai karakteristik, kelebihan, dan kekuarangan masing-masing. Kita akan membahas detail tentang karakteristik, kelebihan, dan kekurangan masing-masing level pada beberapa sub bab. Kita juga akan membahas bagaimana cara mengkonfigurasi RAID pada debian.

RAID Level 0

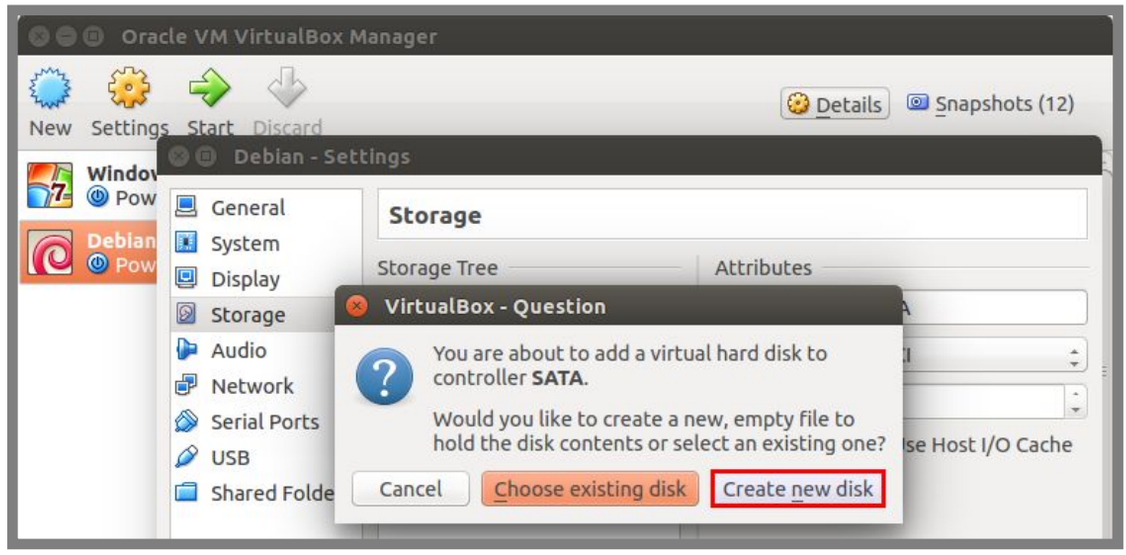
RAID level 0 menggunakan metode penggabungan, artinya jika kita memiliki dua harddisk dengan kapasitas masing-masing 500GB, kita bisa menggabung kedua harddisk tersebut menjadi sebuah harddisk dengan kapasitas 1TB.

Kelebihan dari raid level ini adalah performa pembacaan dan penulisan data ke harddisk akan lebih cepat, karena penyimpanan data akan disebar lintas harddisk. Sedangkan kelemahan dari raid level ini adalah tidak adanya mekanisme deteksi kesalahan, sehingga jika ada salah satu harddisk yang rusak, maka data yang disimpan dalam harddisk tersebut akan hilang dan tidak terbaca.

Untuk praktik mengkonfigurasi raid level 0 pada debian, kita akan menggunakan dua harddisk. Kita bisa menambahkan harddisk virtual di virtualbox dengan langkah-langkah berikut



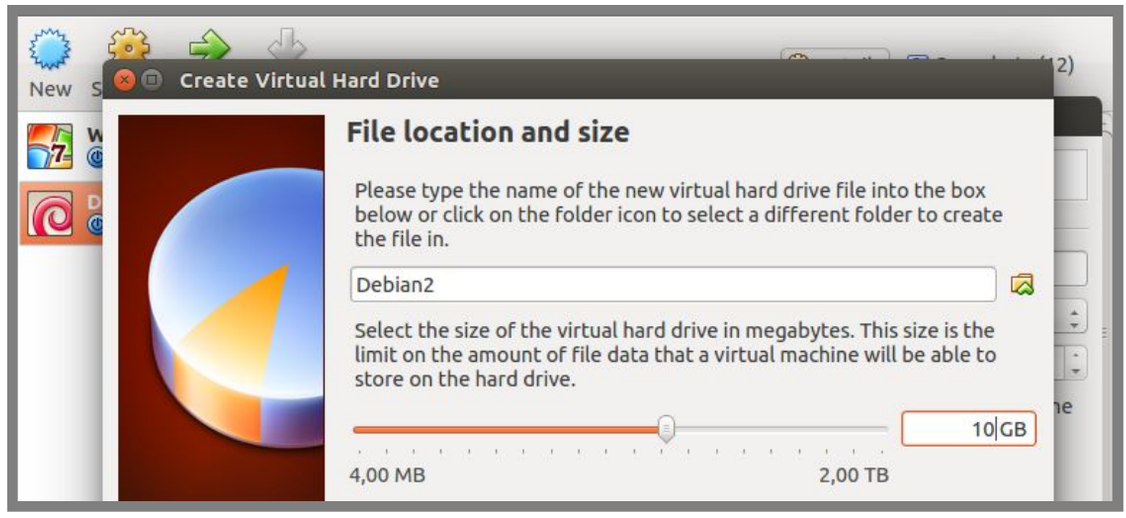
Gambar 19.1 Menambahkan harddisk virtual



Gambar 19.2 Membuat virtual harddisk baru

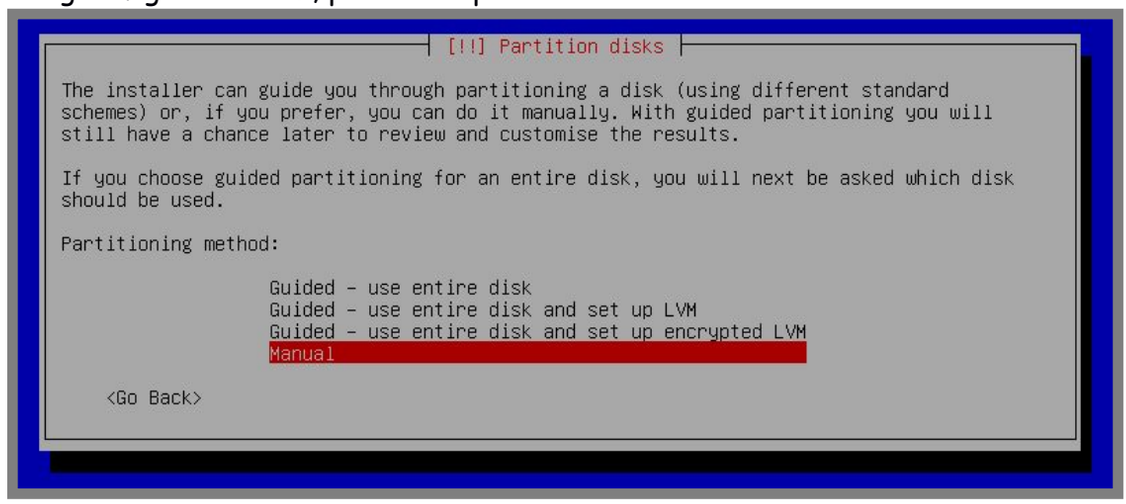
Gambar 19.3 Memilih tipe harddisk virtual

Gambar 19.4 Memilih sistem penyimpanan harddisk virtual



Gambar 19.5 Menentukan nama dan kapasitas harddisk virtual

Setelah selesai menambahkan harddisk, selanjutnya kita bisa melakukan install debian dengan menggunakan RAID level 0. Untuk langkah-langkah installasi debian, bisa dibacah pada bab 2, kita hanya akan fokus pada bagian partisi saja. Untuk mengkonfigurasi RAID, pilih mode partisi manual

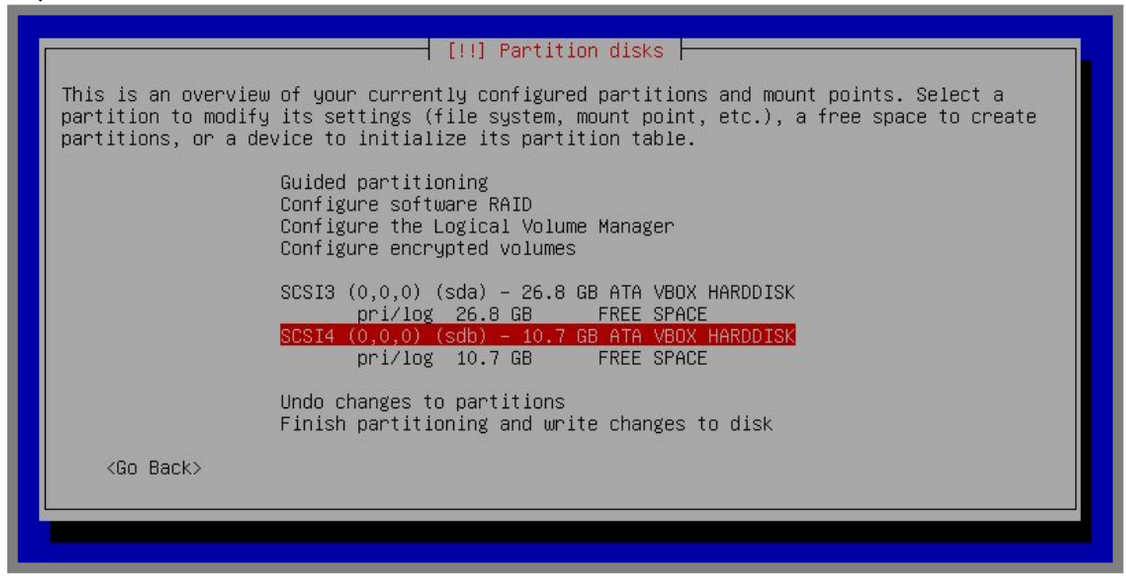


Gambar 19.6 Partisi metode manual

Buat tabel partisi baru pada kedua harddisk

Gambar 19.7 Membuat tabel partisi baru

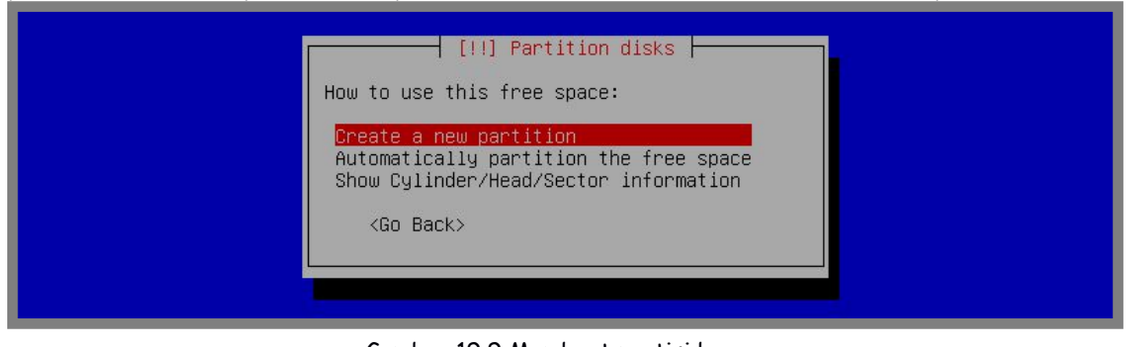
Setelah membuat tabel partisi baru pada kedua harddisk, seharusnya tampilannya seperti berikut



Gambar 19.8 Hasil tabel partisi baru

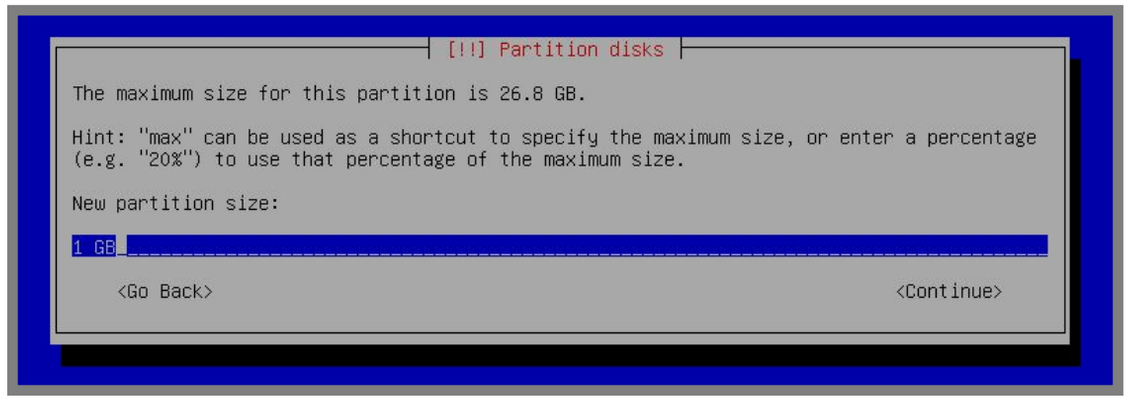
Kali ini, kita ingin membuat partisi swap dengan kapasitas 2GB. Jadi kita nanti akan membuat sebuah partisi pada harddisk 1 dengan kapasitas 1GB dan juga pada harddisk 2 dengan kapasitas 1GB. Selanjutnya kita akan menggabungkan kedua partisi tersebut.

Selanjutnya kita akan menggunakan semua sisa kapasitas harddisk yang ada untuk partisi root. Pilih pada *free space* harddisk 1, kemudian *create a new partition*



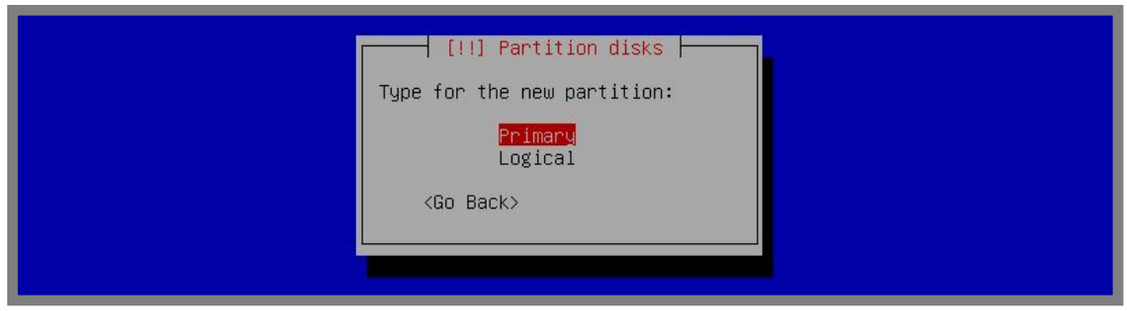
Gambar 19.9 Membuat partisi baru

Masukkan kapasitas partisi yang ingin dibuat 1GB



Gambar 19.10 Menentukan kapasitas partisi baru

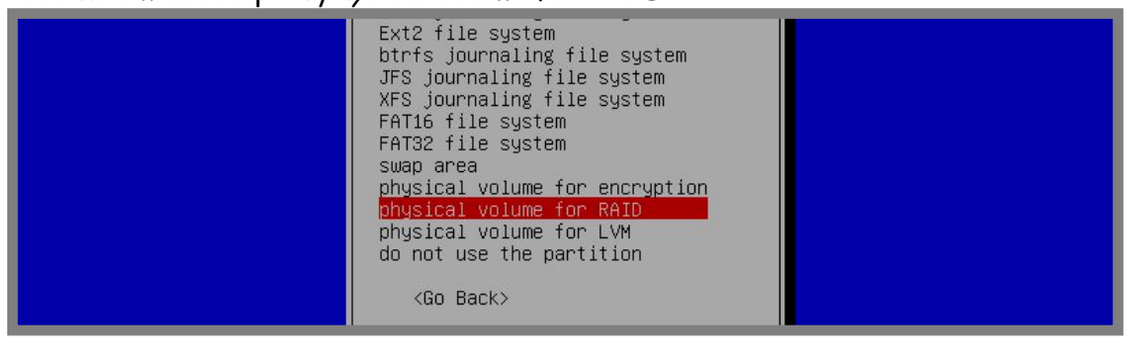
Pilih Primary



Gambar 19.11 Menentukan type partisi

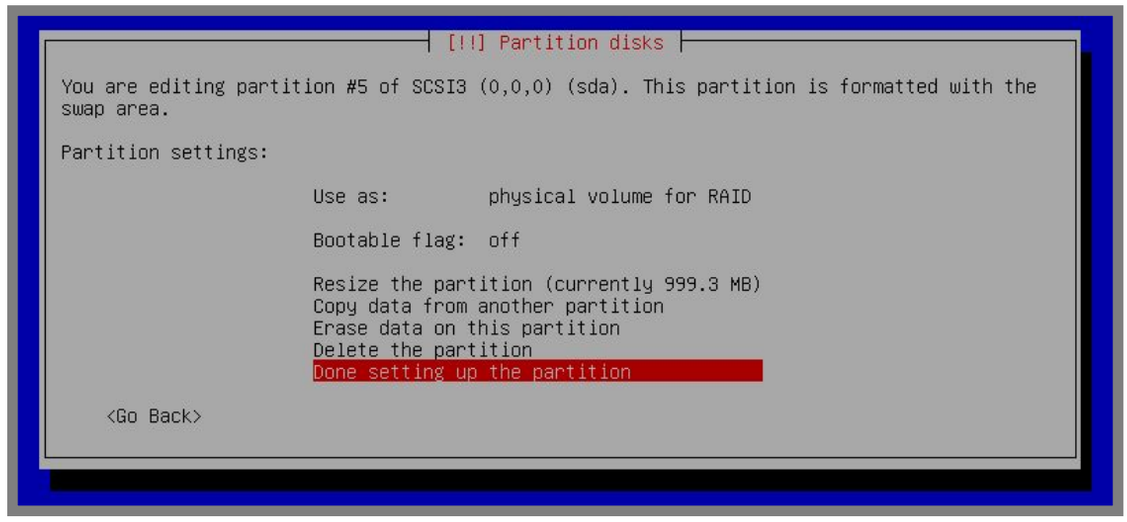
Gambar 19.12 Pilih Beginning

Pada kolom *use as* pilih *physical volume for RAID*



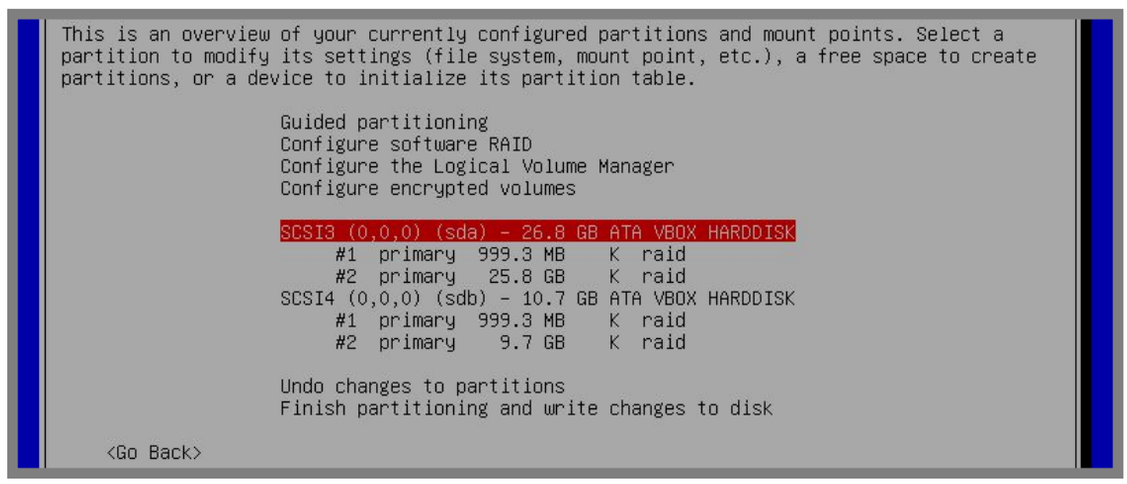
Gambar 19.12 Memilih tipe partisi raid

Kemudian pilih *Done setting up the partition*



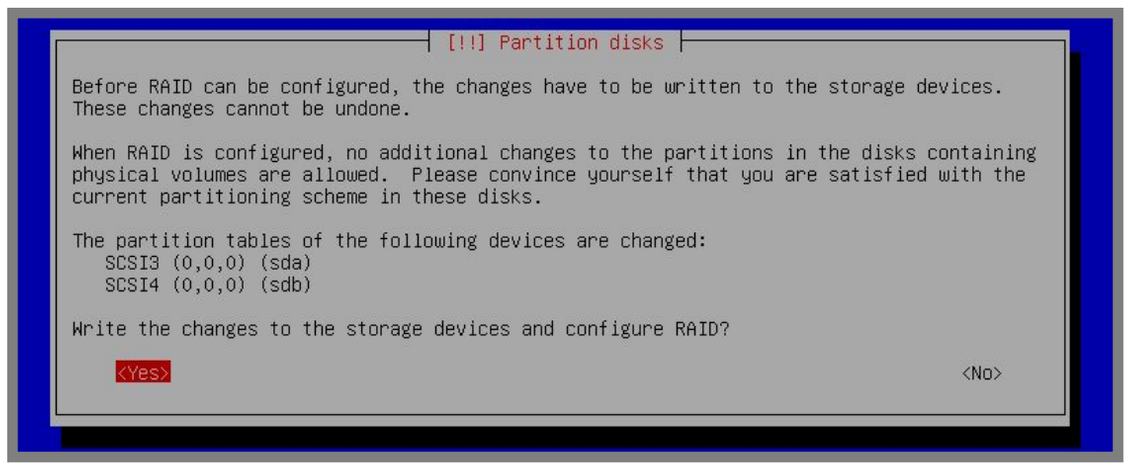
Gambar 19.13 Selesai membuat partisi baru

Lakukan langkah-langkah diatas pada free space di harddisk 1 dan juga harddisk 2, sehingga hasilnya akan nampak seperti berikut



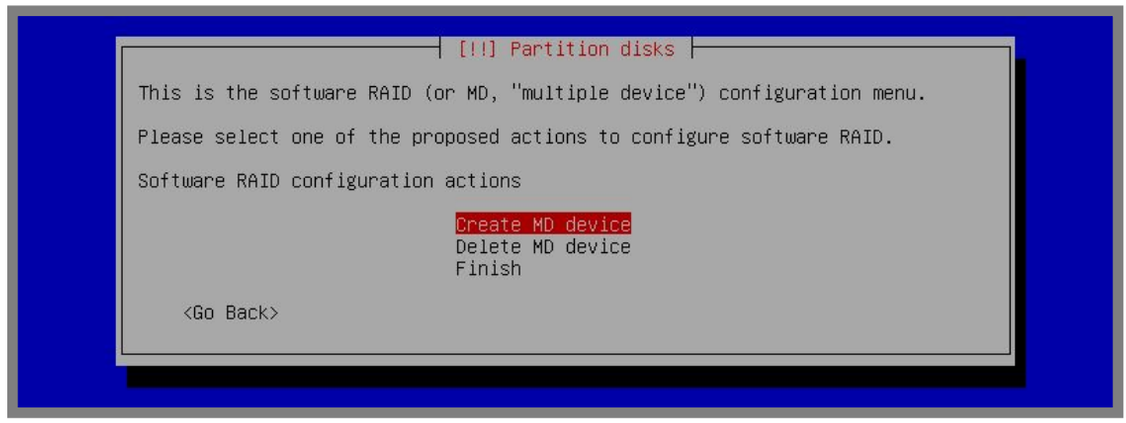
Gambar 19.14 Hasil pembuatan partisi

Selanjutnya pilih *Configure software RAID* dan pilih *Yes*



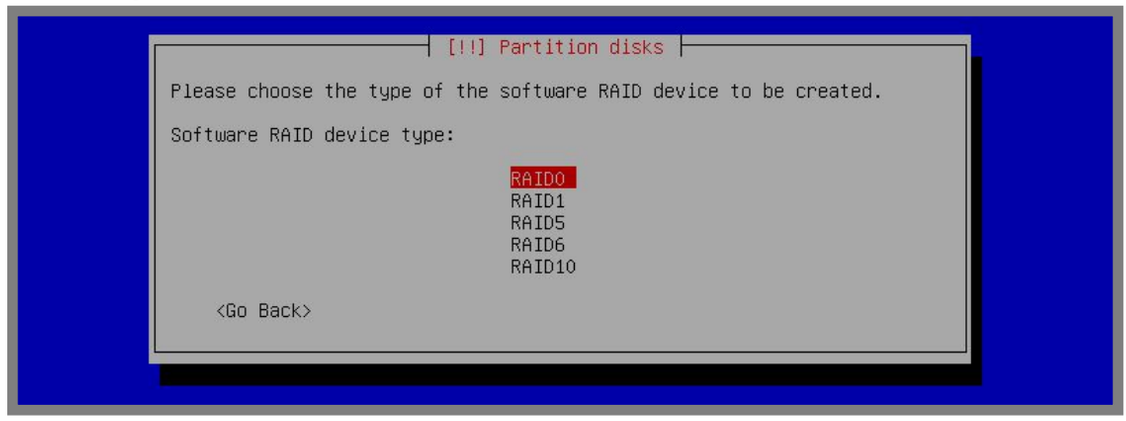
Gambar 19.15 Konfigurasi software raid

Pilih *create md device*



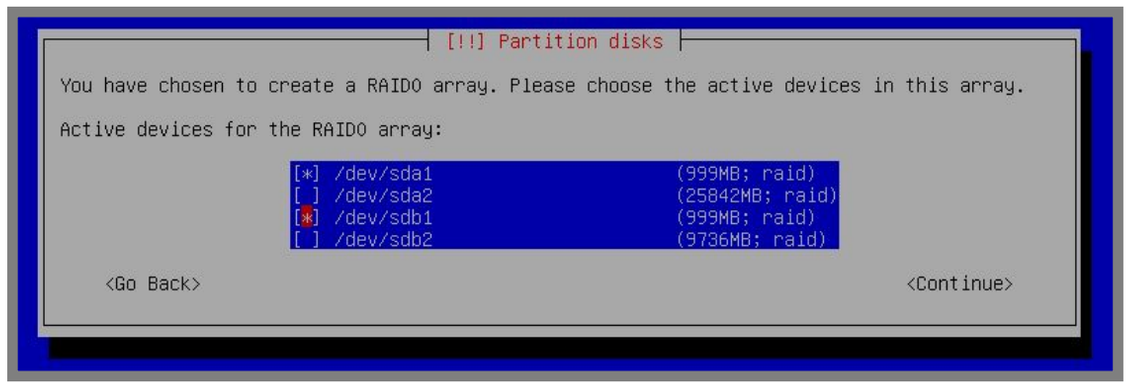
Gambar 19.16 Proses konfigurasi software raid

Pilih *RAID0*



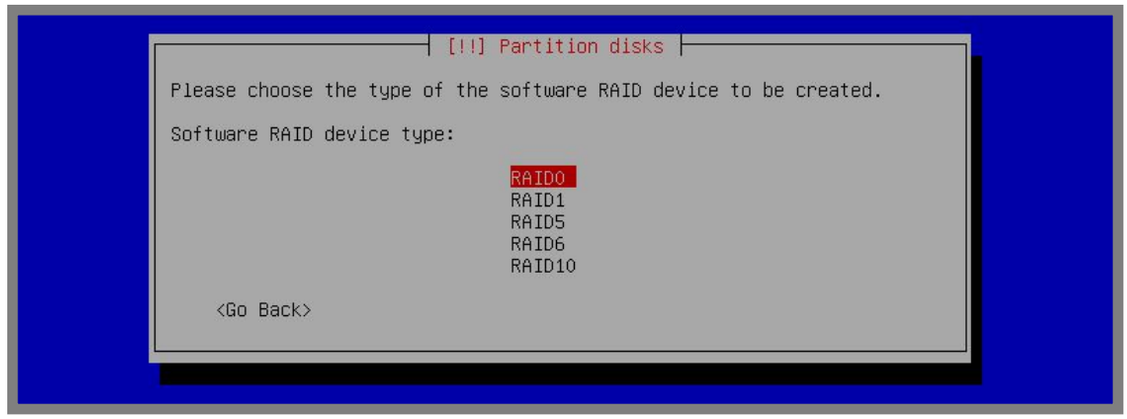
Gambar 19.17 Pilih RAID0

Pilih pada partisi yang ukurannya 1GB, kedua partisi ini nantinya akan digabung menjadi 2GB untuk menjadi partisi swap



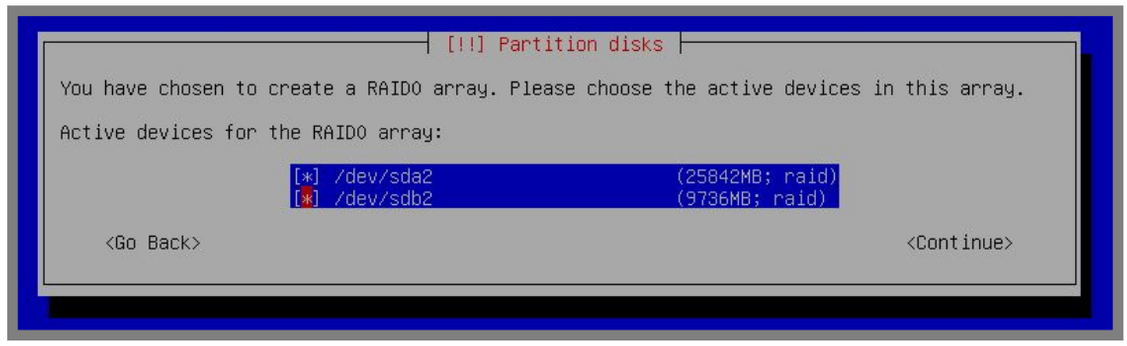
Gambar 19.18 Pilih partisi untuk swap

Kemudian *create MD device* lagi dan pilih *RAID0*



Gambar 19.19 Membuat md device yang kedua

Sekarang centang pada partisi yang akan digabung untuk menjadi partisi root

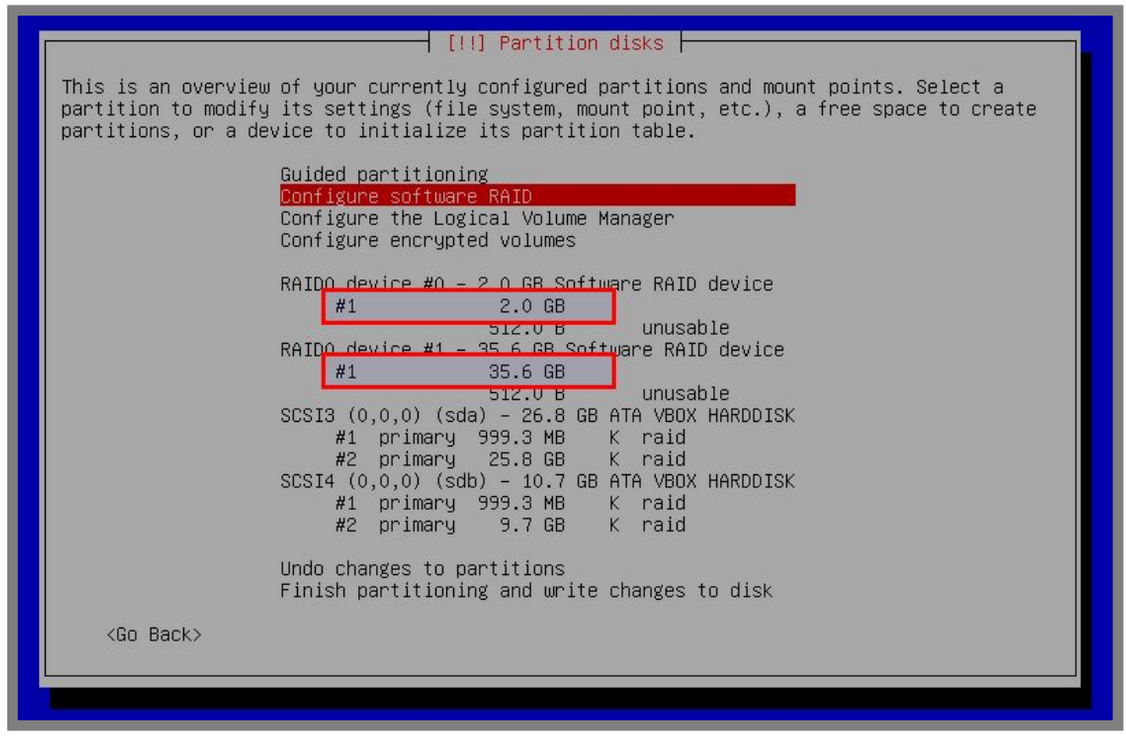


Gambar 19.20 Pilih partisi untuk root

Selanjutnya pilih finish

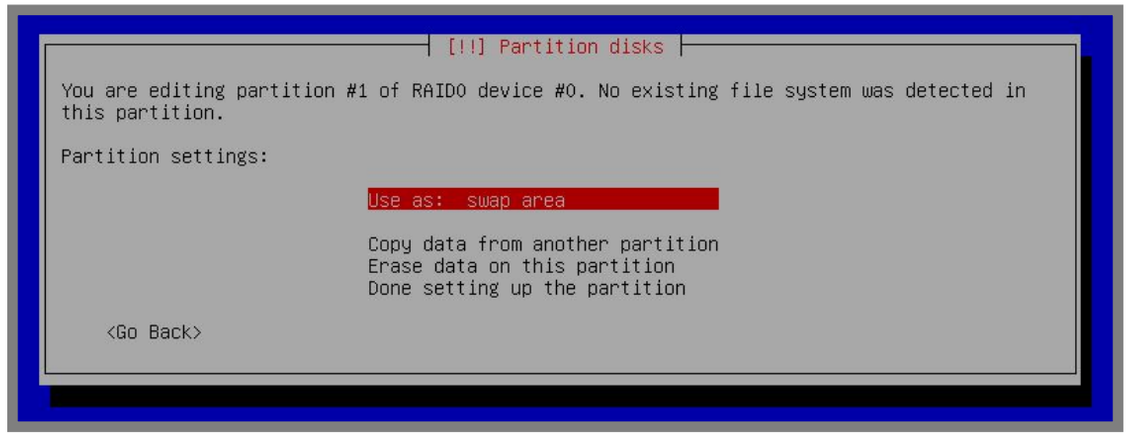
Gambar 19.21 Selesai konfigurasi software raid

Hasilnya akan seperti ini



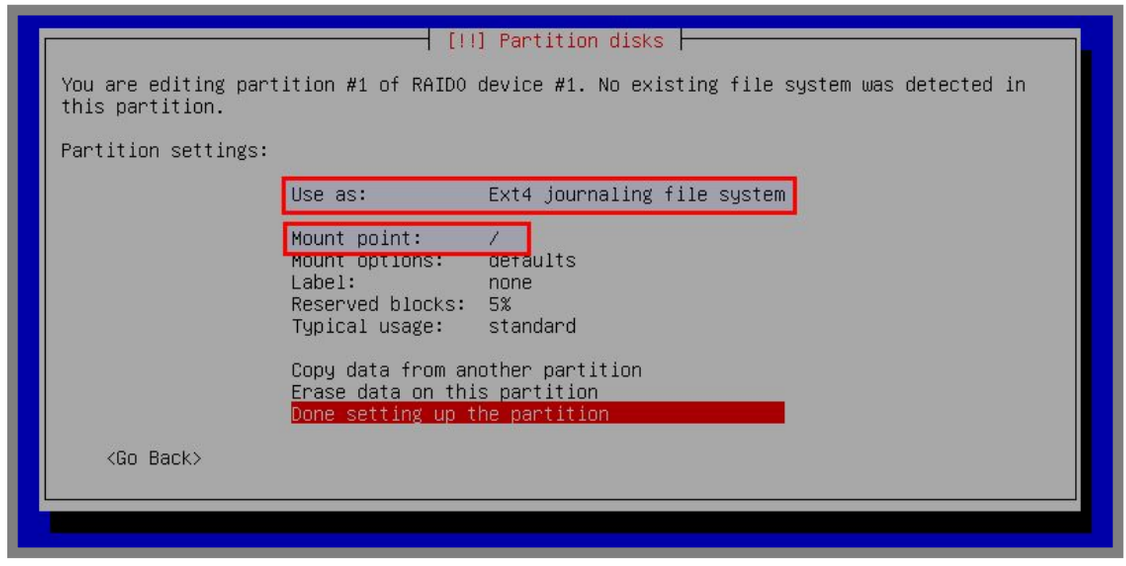
Gambar 19.22 Hasil konfigurasi software raid

Perhatikan pada partisi 2GB dan 35,6GB. Kedua partisi ini adalah gabungan dari dua harddisk yang kita miliki. Nantinya partisi 2GB akan kita gunakan sebagai swap area dan partisi 35,6GB akan kita gunakan untuk root. Pilih pada 2GB, kemudian pada kolom *use as* pilih *swap area*



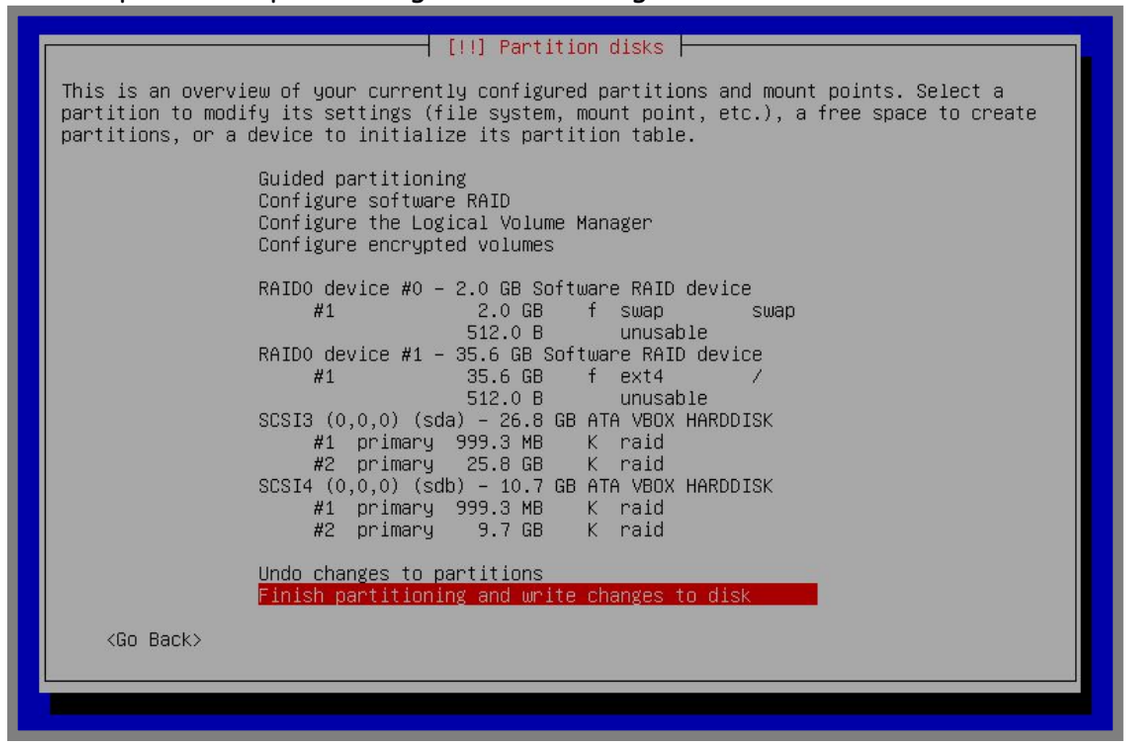
Gambar 19.23 Konfigurasi partisi swap

Selanjutnya pilih *Done setting up the partition* kemudian lakukan langkah yang sama pada partisi 35,6GB, hanya saja pada kolom *use as* kita akan memilih *Ext4 journaling file system* kemudian pada bagian *Mount point* kita pilih root (/).



Gambar 19.24 Konfigurasi partisi root

Terahir pilih finish partitioning and write changes to disk



Gambar 19.25 Selesai konfigurasi partisi

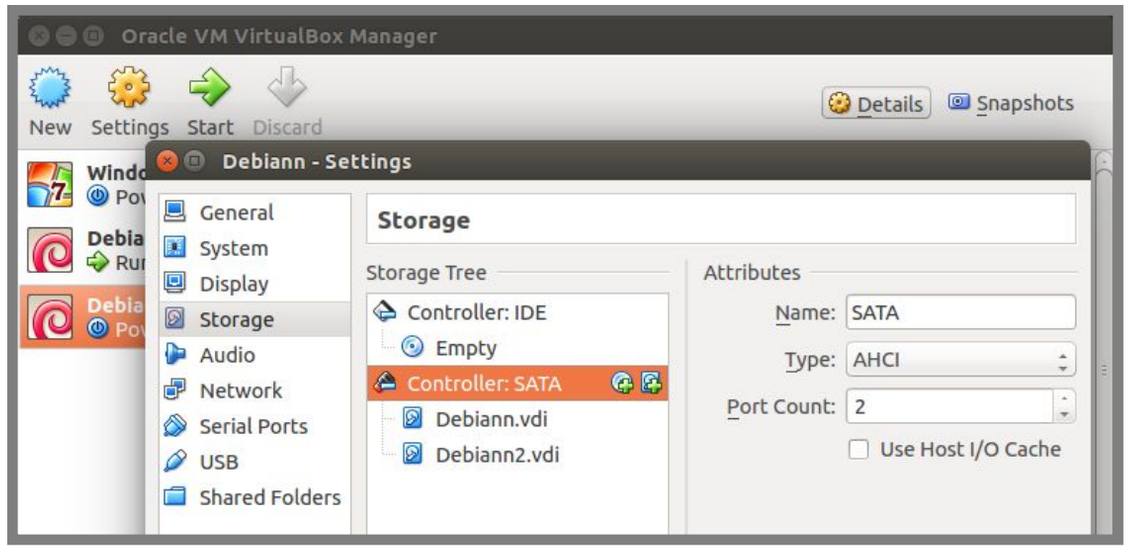
Sampai saat ini kita sudah selesai melakukan partisi menggunakan RAID level 0 pada debian. Selanjutnya kita bisa melanjutkan proses installasi seperti biasa.

RAID Level 1

RAID pada level ini akan menggunakan prinsip mirroring, artinya data akan disimpan kedalam dua atau lebih harddisk yang ada. Kelebihan dari raid level ini adalah bahwa data disimpan kedalam dua harddisk sekaligus, sehingga jika suatu saat harddisk 1 rusak, maka data masih ada didalam harddisk 2.

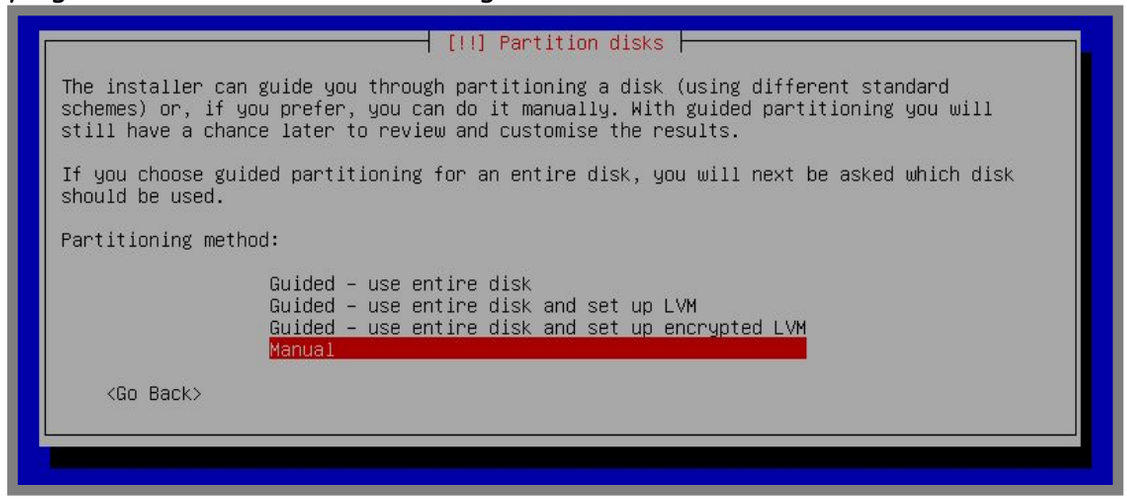
Sedangkan kelemahan dari raid level ini adalah tidak menambah kapasitas harddisk, jadi misal kita mempunyai dua harddisk dengan ukuran masing-masing 10GB, maka kapasitasnya tidak menjadi 20GB, melainkan tetap 10GB. Selain itu, proses penulisan data keharddisk juga akan relatif lebih lama, hal ini dikarenakan data pada harddisk 1 harus disinkronkan dengan data pada harddisk 2.

RAID pada level ini membutuhkan minimal dua harddisk. Diasumsikan kita telah memiliki dua harddisk dengan ukuran masing-masing 10GB divirtual machine.



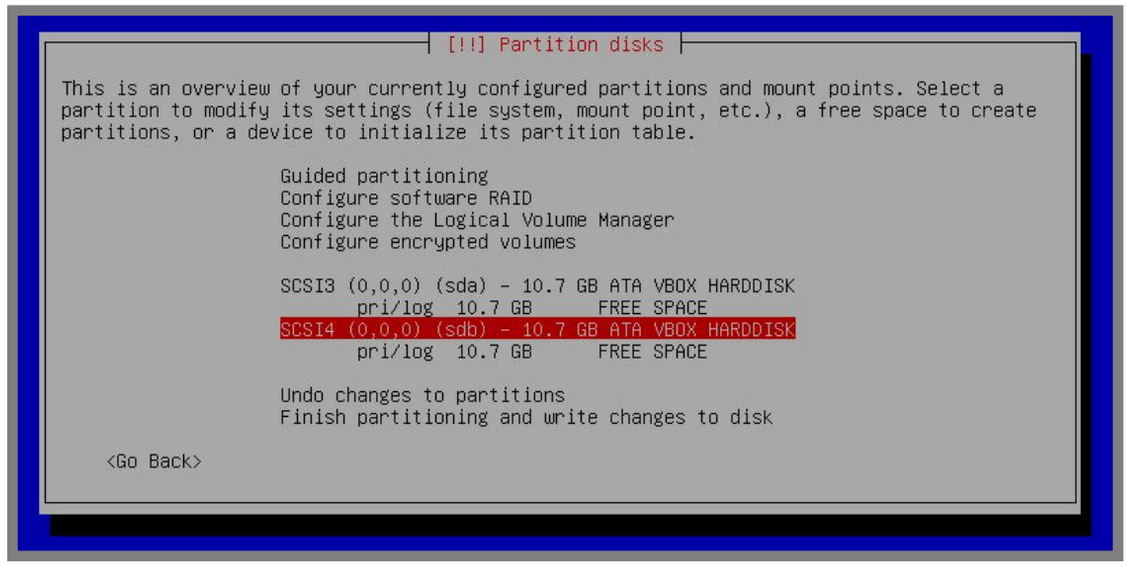
Gambar 19.26 Harddisk untuk praktik raid 1

Selanjutnya silahkan booting dengan installaer debian dan berikut proses partisi yang harus kita lakukan untuk konfigurasi raid 1



Gambar 19.27 Metode partisi manual

Hal pertama yang harus dilakukan adalah membuat tabel partisi baru sehingga hasilnya akan nampak seperti berikut

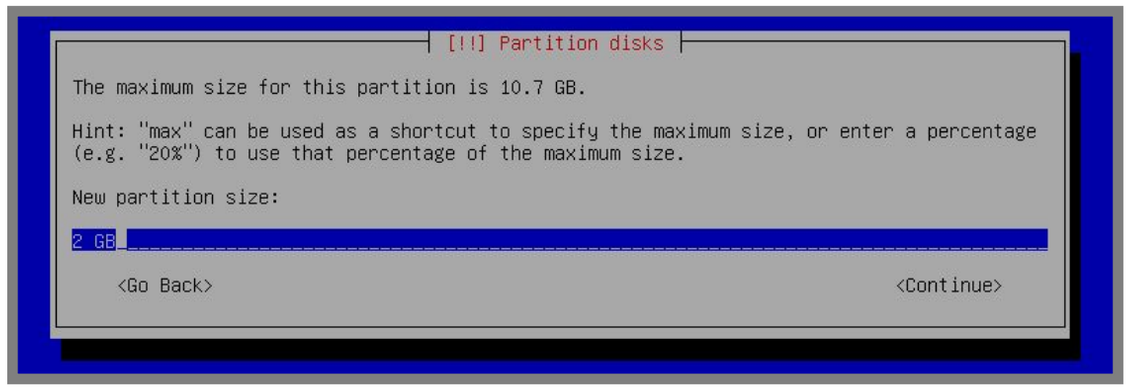


Gambar 19.28 Membuat tabel partisi baru

Sudah dikatakan sebelumnya bahwa raid pada level ini hanya berfungsi untuk mirroring dan tidak menambah kapasitas harddisk. Nantinya kita akan membuat paritisi swap sebesar 2GB dan sisanya untuk root. Buat partisi baru pada free space harddisk 1

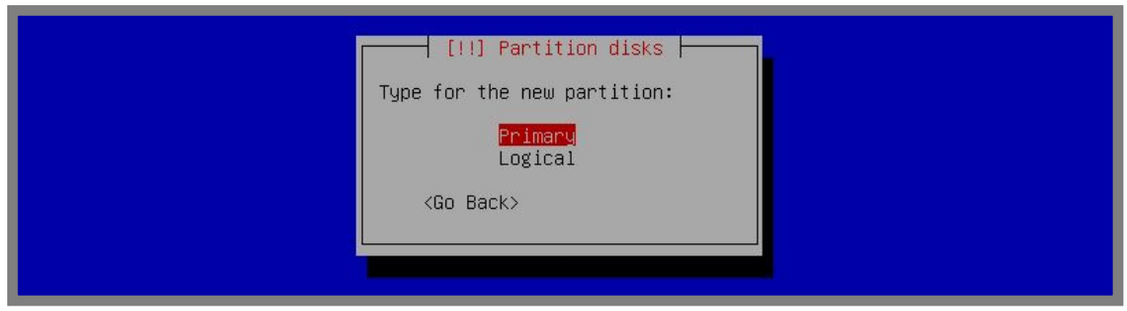
Gambar 19.29 Membuat partisi baru

Konfigurasi space dengan 2GB untuk partisi swap



Gambar 19.30 Kapasitas harddisk untuk swap

Pilih primary

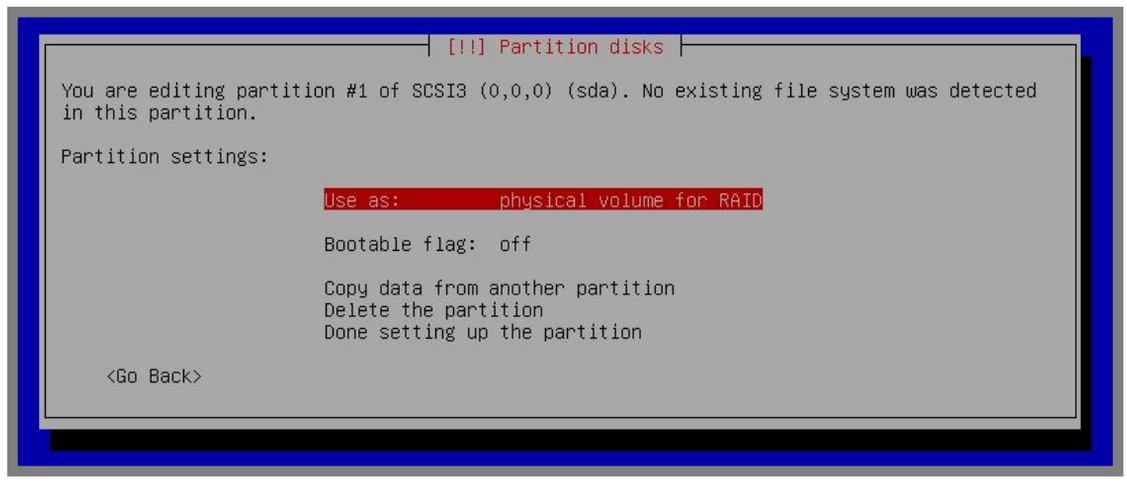


Gambar 19.31 Memilih tipe harddisk primary

Pilih beginning

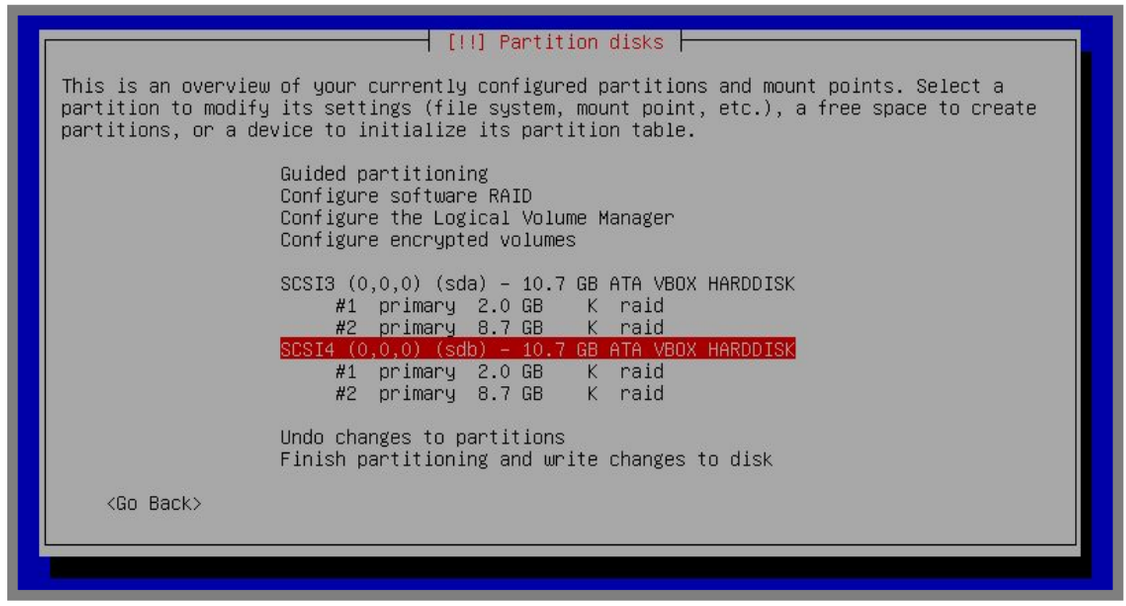
Gambar 19.32 Pilih beginning

Pada kolom use as pilih *physical volume for RAID*



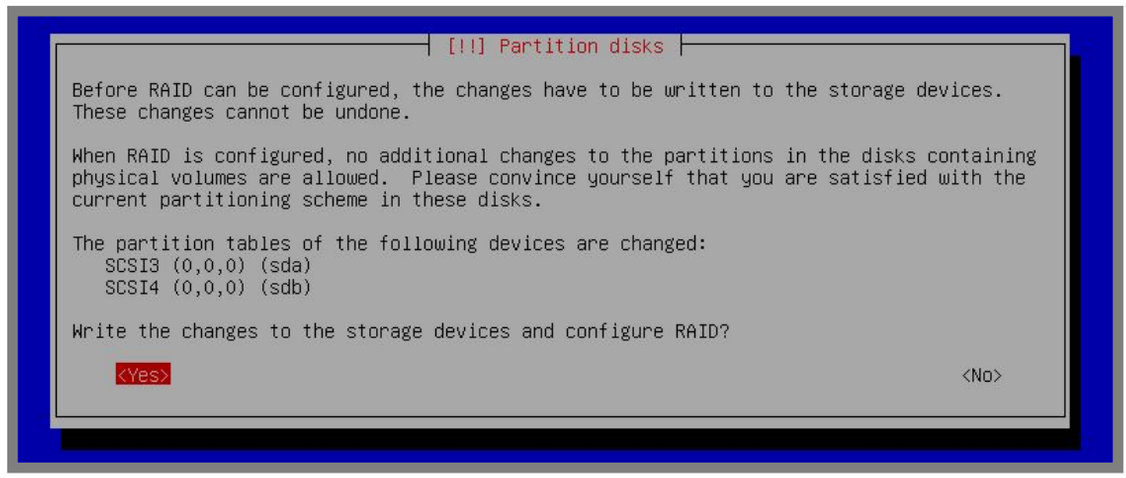
Gambar 19.33 Memilih tipe partisi raid

Selanjutnya pilih *Done setting up the partition* kemudian lakukan langkah yang sama seperti diatas pada free space di harddisk 1, lakukan juga pada free space harddisk 2 sehingga hasilnya akan nampak seperti berikut



Gambar 19.34 Hasil konfigurasi partisi untuk raid

Selanjutnya pilih *Configure software RAID* dan pilih *yes*

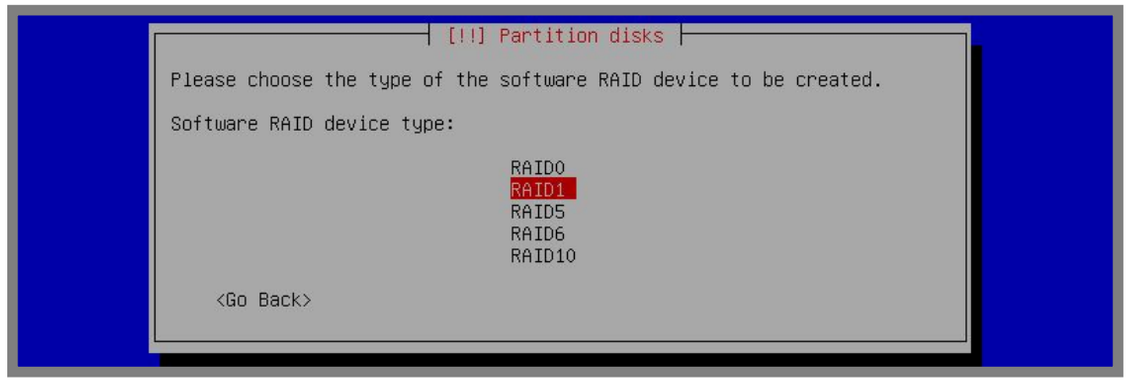


Gambar 19.35 Verifikasi konfigurasi software raid

Pilih *Create MD Device*

Gambar 19.36 Membuat MD device

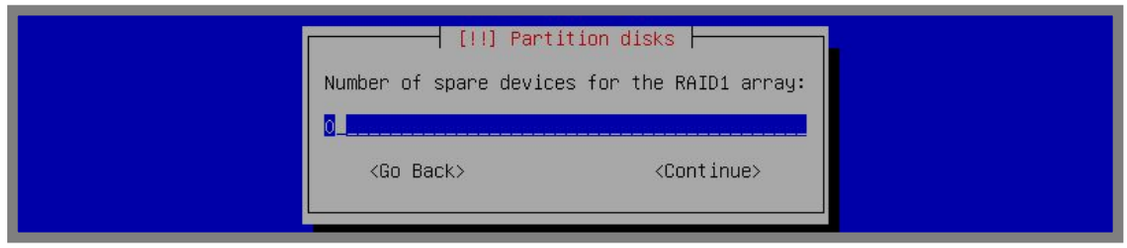
Pilih *RAID1*



Gambar 19.37 Memilih raid level 1

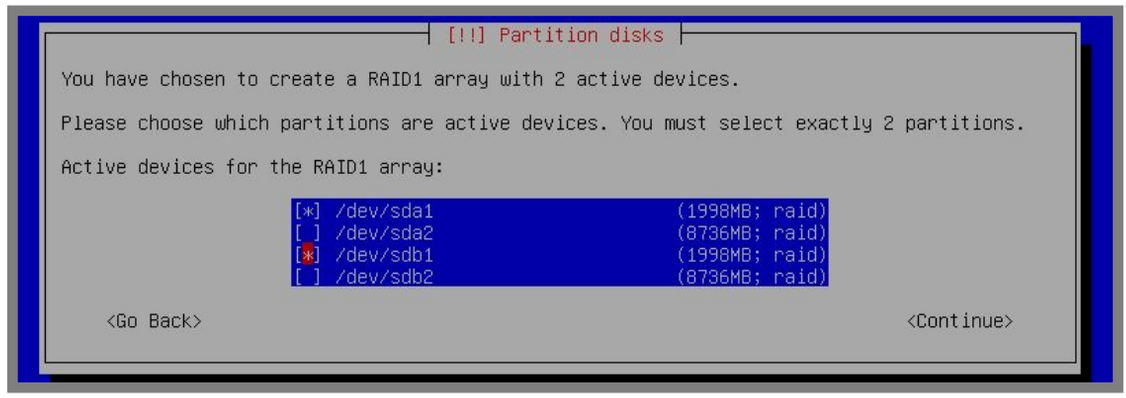
Masukkan jumlah harddisk yang kita gunakan

Gambar 19.37 Menentukan jumlah harddisk yang digunakan



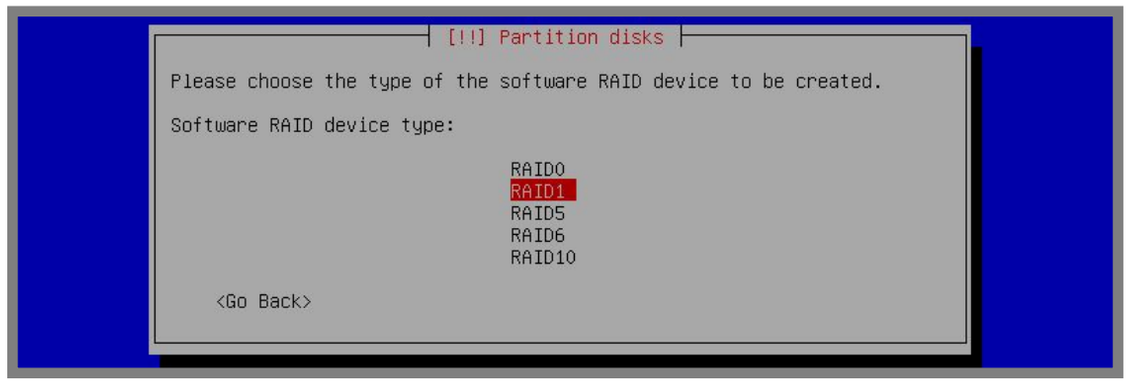
Gambar 19.38 Proses konfigurasi raid 1

Centang pada partisi yang akan kita gunakan untuk swap



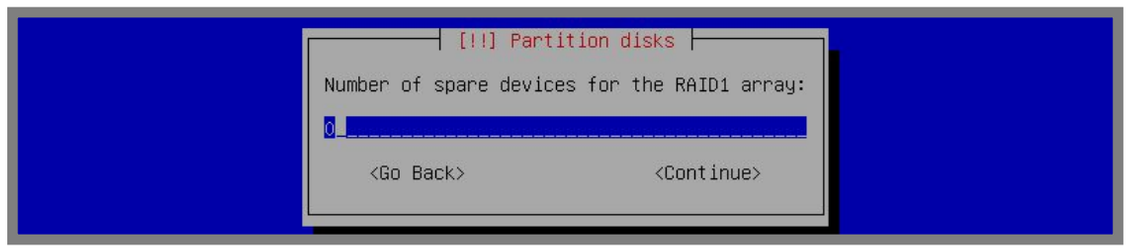
Gambar 19.39 Partisi untuk swap

Selanjutnya pilih *Create MD device* lagi dan pilih *RAID1* untuk persiapan partisi root



Gambar 19.40 Partisi raid level 1 untuk root

Gambar 19.40 Masukkann jumlah harddisk yang digunakan

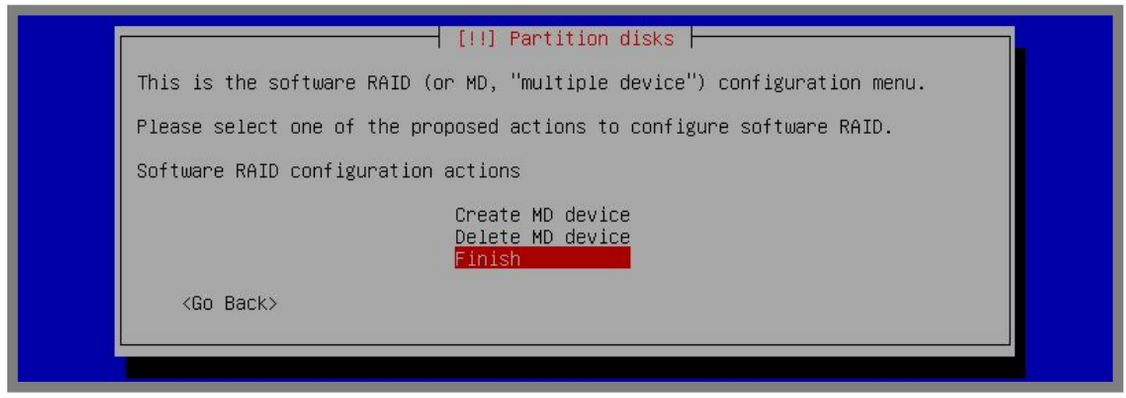


Gambar 19.41 Masukkan 0

Centang pada partisi yang akan digunakan untuk root

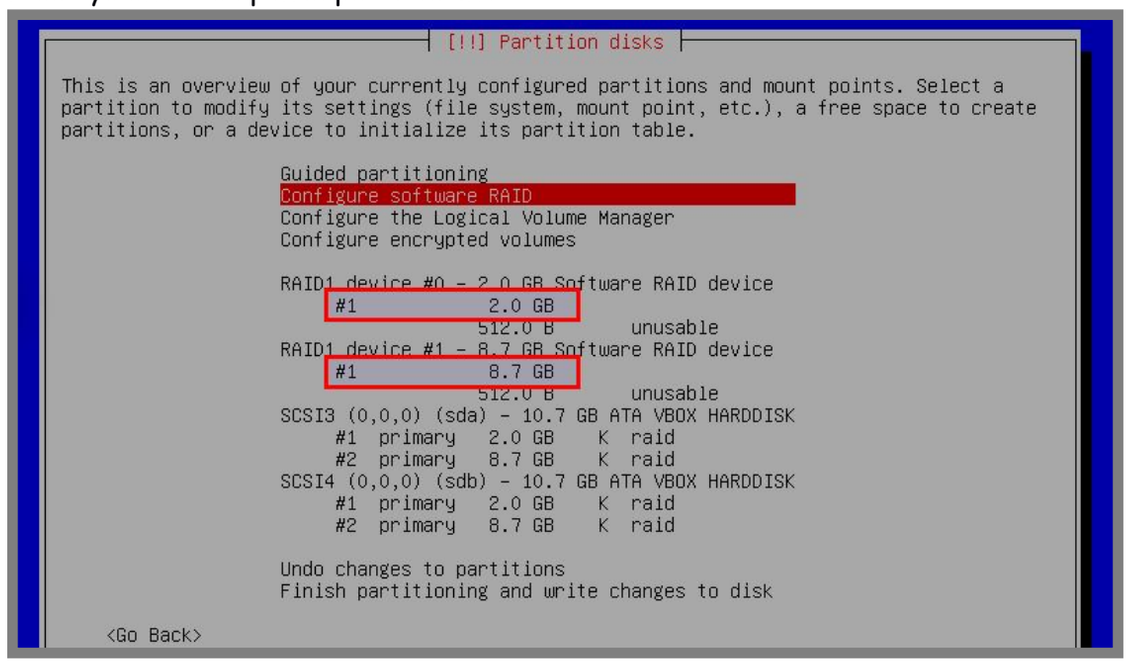
Gambar 19.42 Memilih partisi untuk root

Pilih *finish*



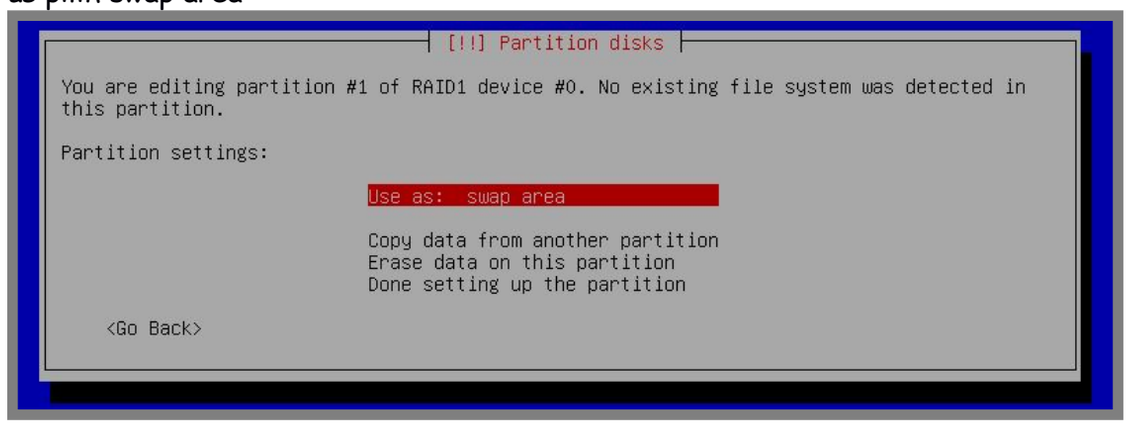
Gambar 19.43 Selesai membuat md device

Hasilnya akan nampak seperti berikut



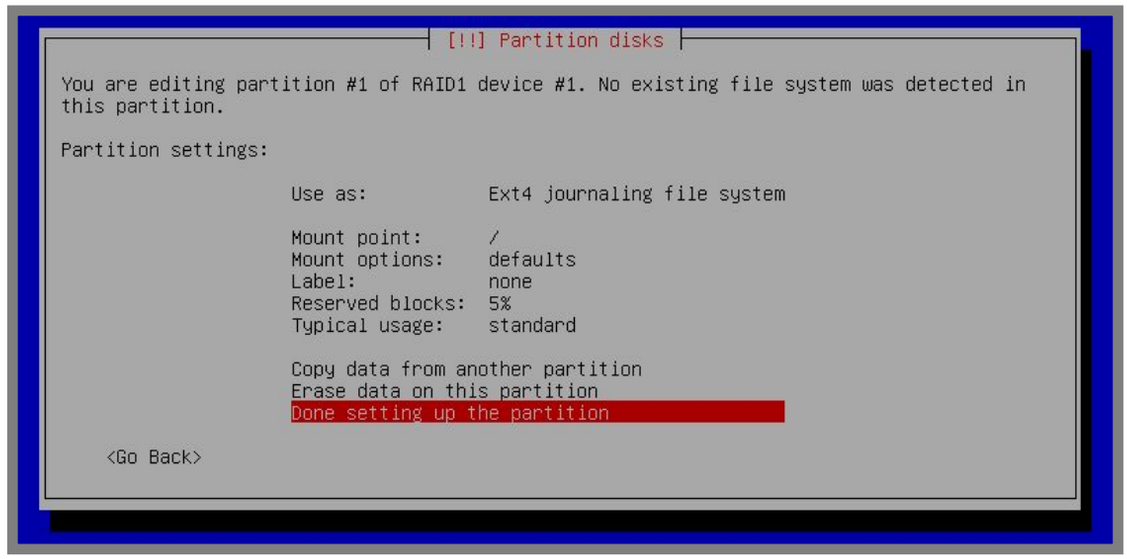
Gambar 19.44 Hasil konfigurasi software raid

Perhatikan pada partisi 2GB dan 8,7GB. Nantinya kita akan menggunakan partisi 2GB untuk swap dan partisi 8,7GB untuk root. Pilih pada partisi 2GB dan pada *use as* pilih swap area



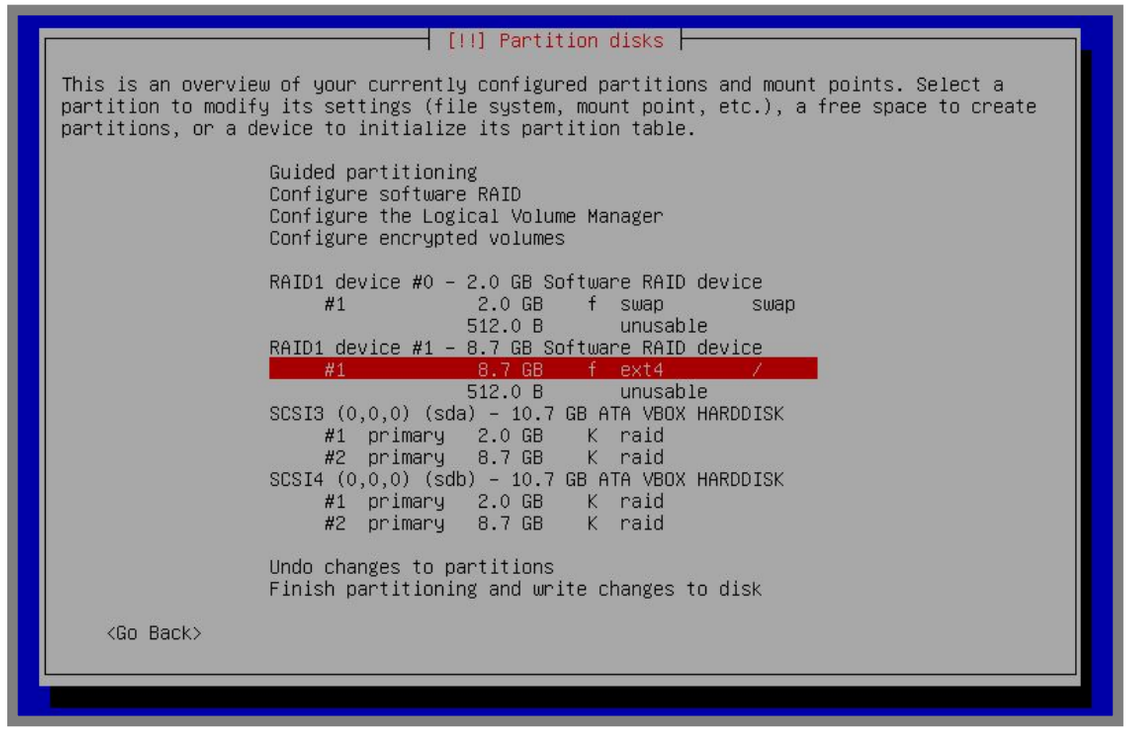
Gambar 19.45 Membuat partisi swap

Pilih *Done setting up the partition* kemudian pada partisi 8,7GB konfigurasikan untuk menjadi partisi root seperti berikut



Gambar 19.46 Konfigurasi partisi root

Sehingga hasilnya akan nampak seperti berikut



Gambar 19.47 Hasil konfigurasi partisi raid 1

Sampai saat ini kita sudah selesai melakukan konfigurasi partisi dengan raid 1. Perhatikan bahwa kita memiliki kapasitas 2GB untuk partisi swap dan 8,7GB untuk partisi root. Sehingga dapat disimpulkan bahwa total kapasitas yang kita miliki adalah 10GB.

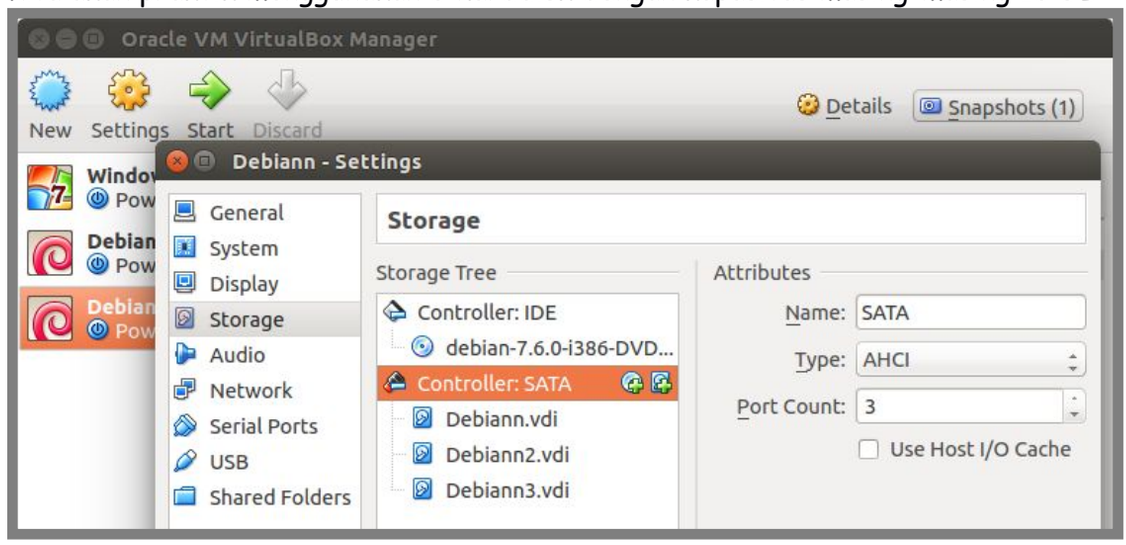
Selanjutnya pilih *Finish partitioning and write changes to disk* dan lanjutkan proses installasi debian seperti biasa.

RAID Level 5

RAID pada level ini membutuhkan minimal 3 harddisk. Pada level ini jumlah maksimum harddisk yang rusak adalah satu, artinya data tidak akan hilang atau rusak jika ada satu harddisk yang rusak.

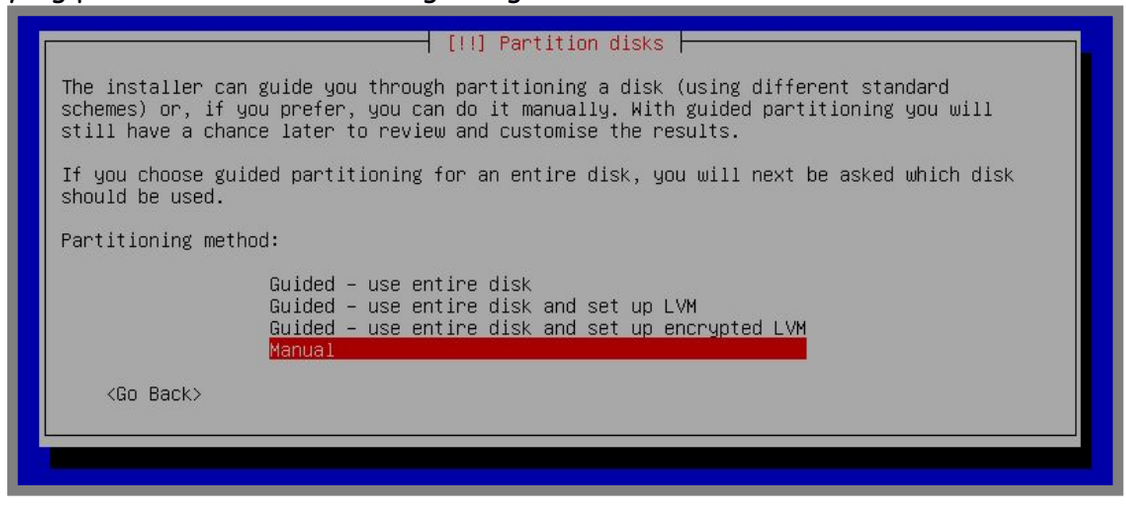
Kapasitas yang didapat pada level ini dapat kita hitung dengan rumus *(1-1/n) * m.* Misal kita memiliki 3 harddisk dengan kapasitas masing-masing 10GB maka kapasitas yang kita dapat adalah (1-1/3) * 30 = 2/3 * 30 = 20GB.

Kita akan praktik menggunakan 3 harddisk dengan kapasitas masing-masing 10 GB



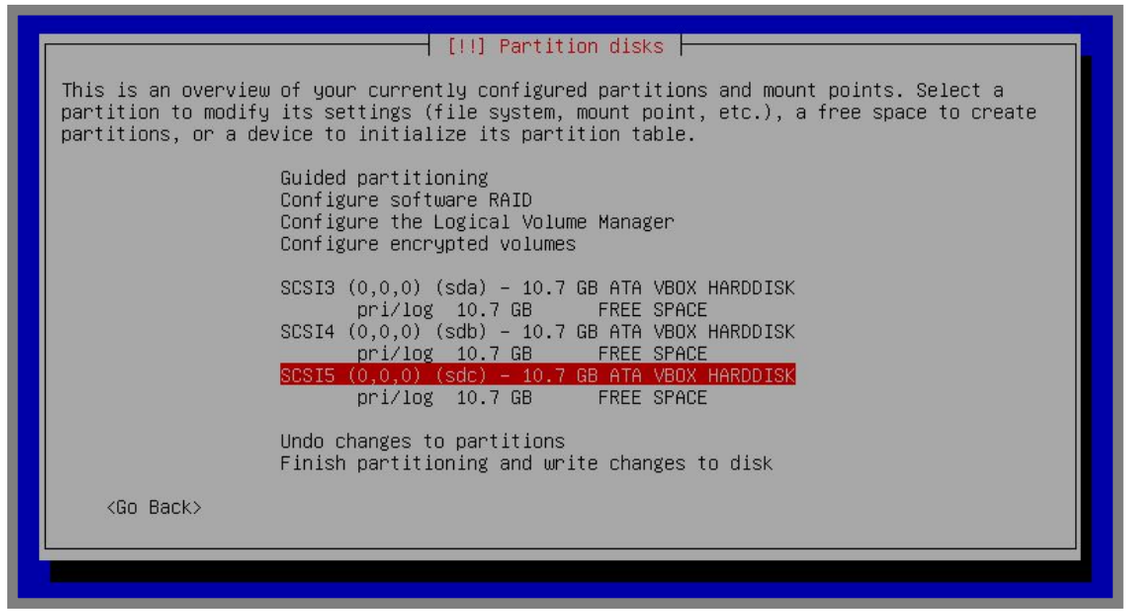
Gambar 19.48 Virtual harddisk untuk raid 5

Lakukan booting menggunakan installer debian, berikut langkah-langkah partisi yang perlu dilakukan untuk mengkonfigurasi raid 5



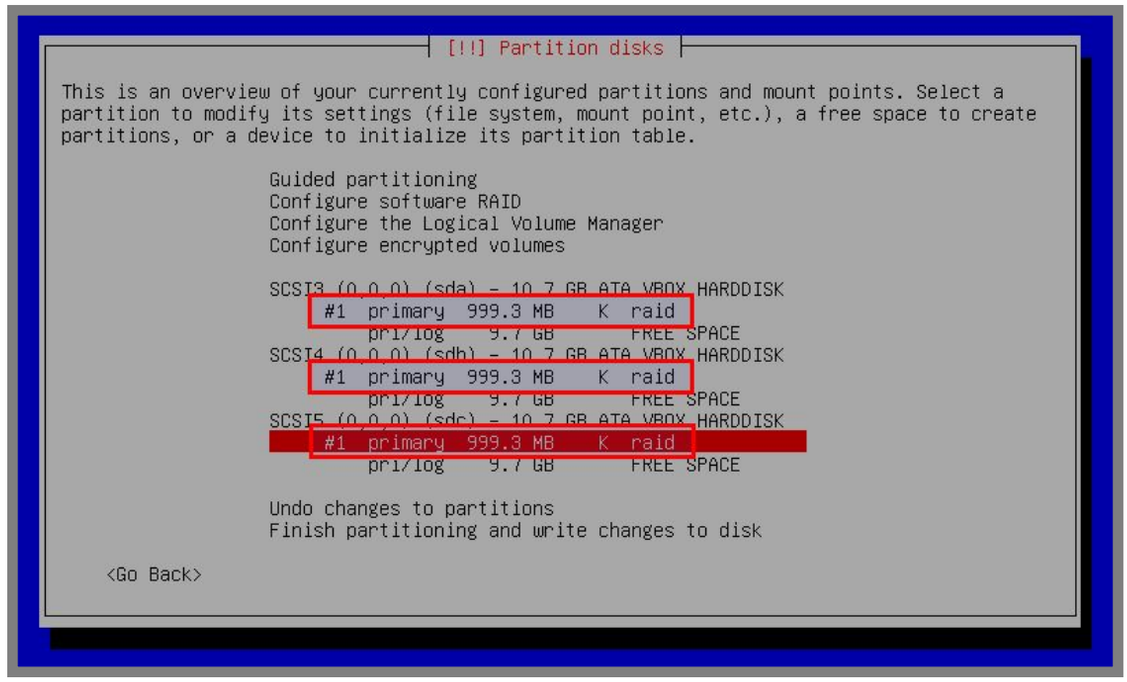
Gambar 19.49 Metode partisi manual

Hal pertama yang harus dilakukan adalah membuat tabel partisi baru pada ketiga harddisk tersebut



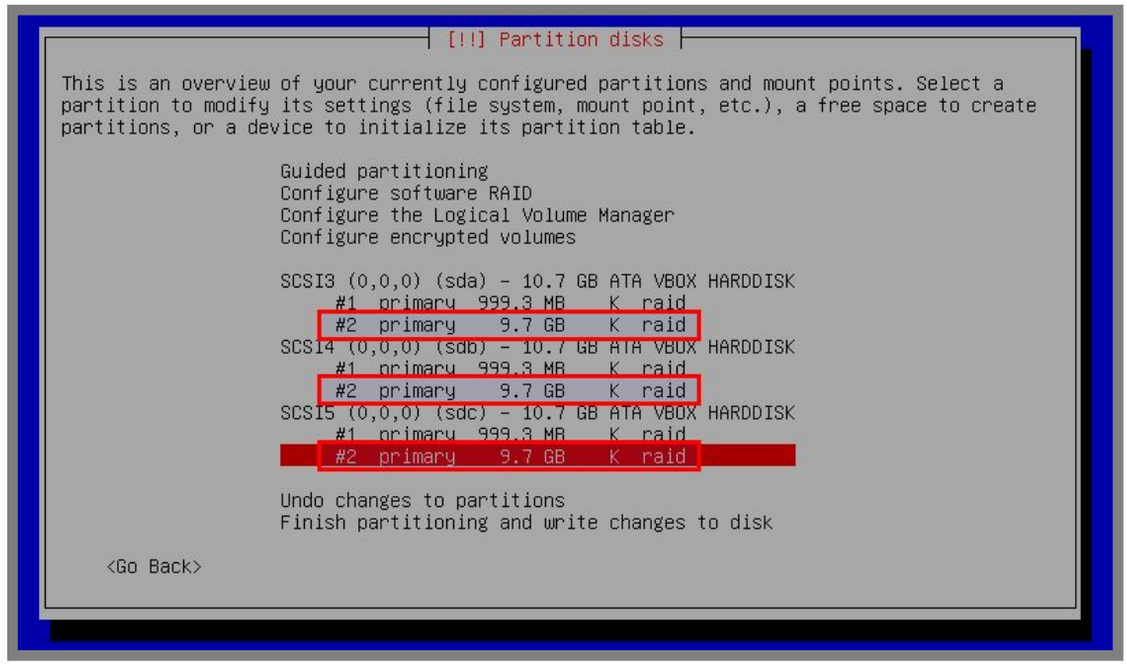
Gambar 19.50 Membuat tabel partisi baru

Kita nanti menginginkan partisi swap dengan kapasitas 2GB, untuk itu pertama-tama buat partisi 1GB pada ketiga harddisk tersebut dengan tipe raid seperti berikut



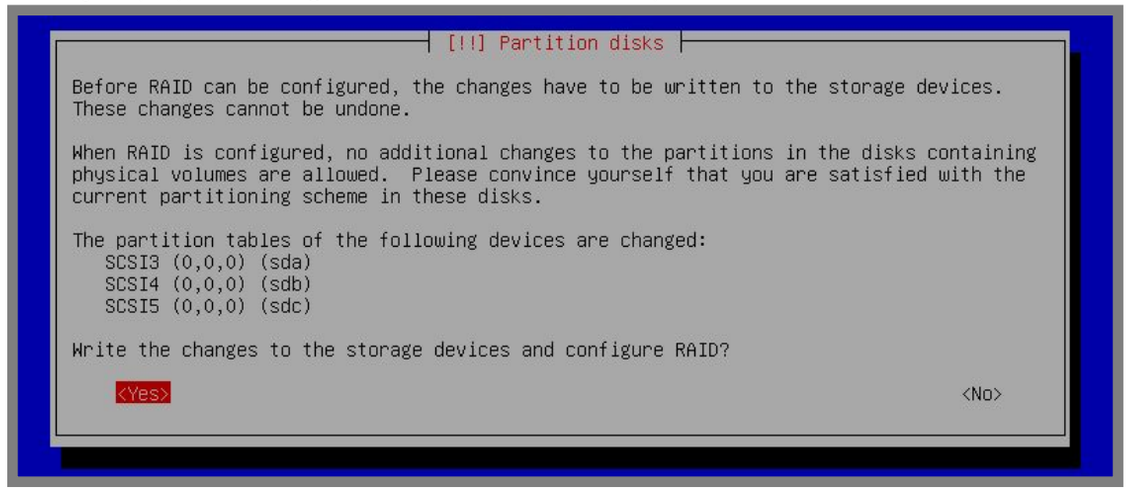
Gambar 19.51 Menyiapkan partisi untuk swap

Lakukan hal yang sama pada free space di ketiga harddisk teresebut untuk membuat partisi root



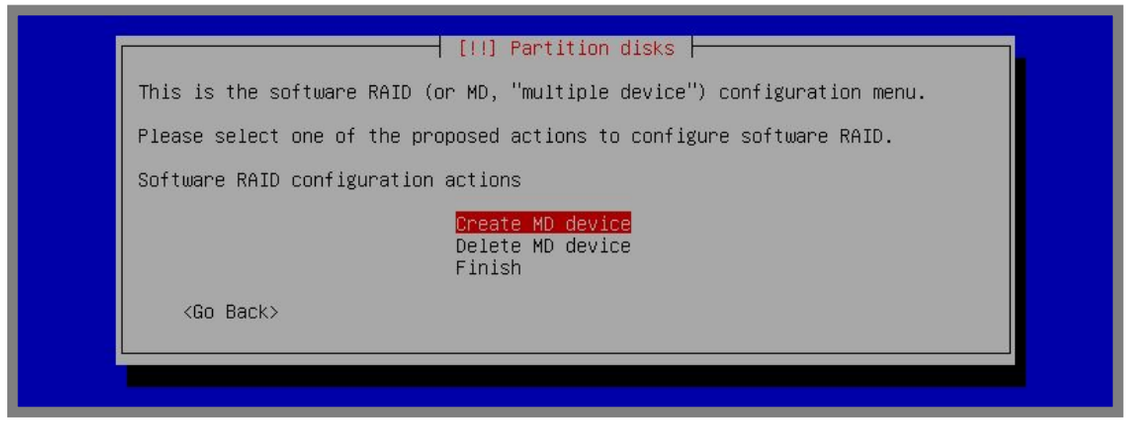
Gambar 19.52 Menyiapkan partisi untuk root

Selanjutnya pilih *Configure software RAID* dan pilih *Yes*



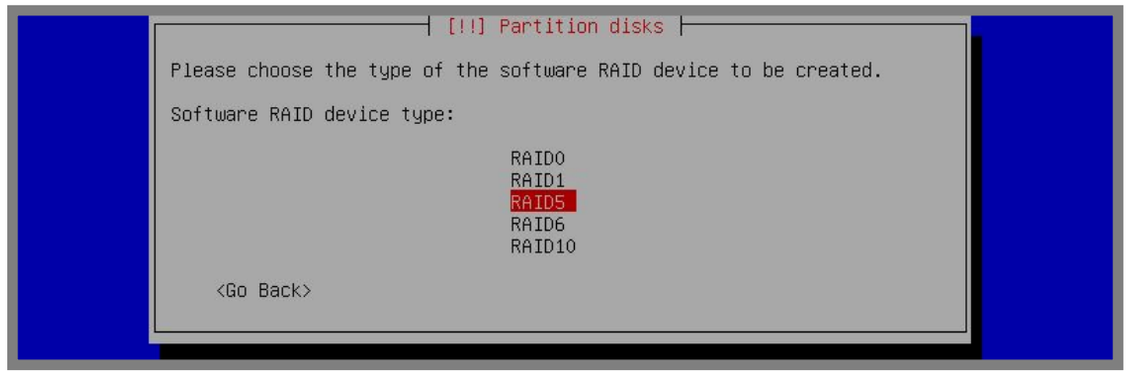
Gambar 19.53 Verfikasi konfigurasi software raid

Pilih *Create MD device*



Gambar 19.54 Membuat md device

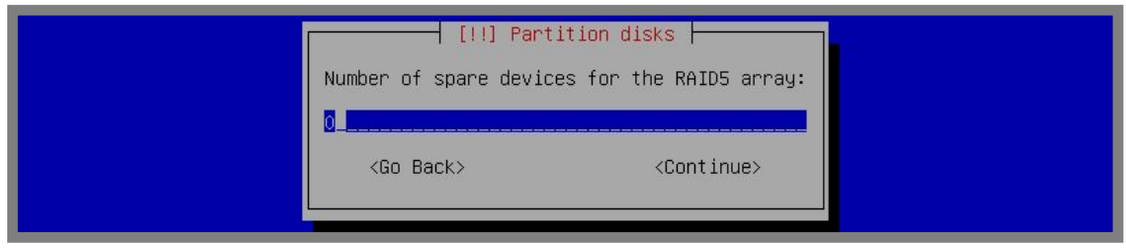
Pilih *RAID5*



Gambar 19.55 Pilih raid level 5

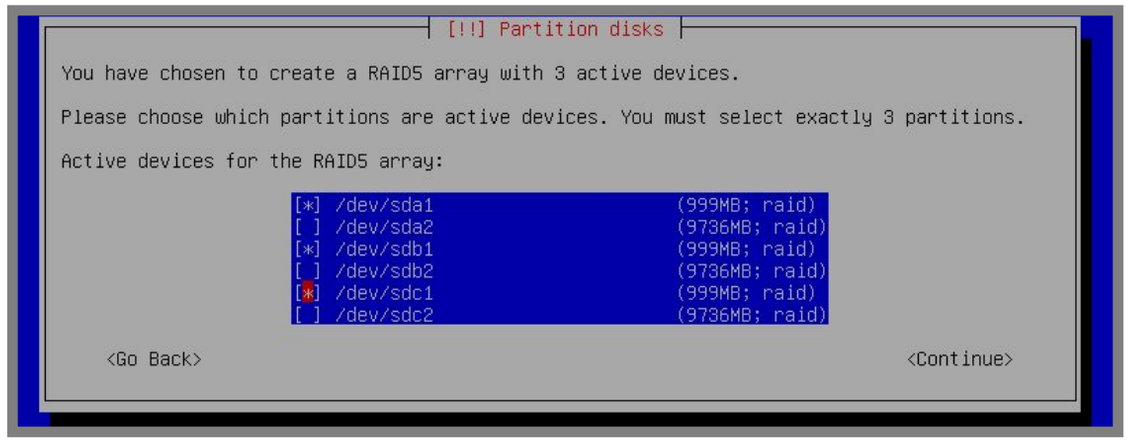
Masukkan jumlah harddisk yang digunakan

Gambar 19.56 Jumlah harddisk yang digunakan



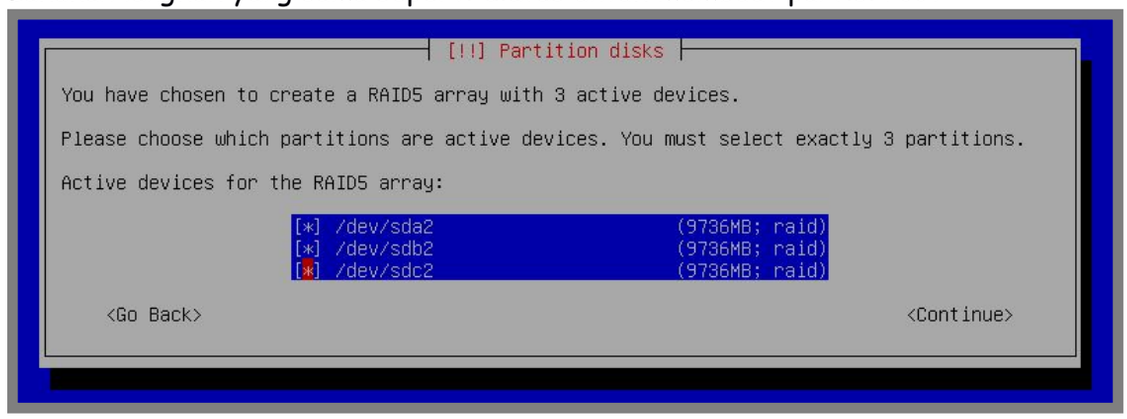
Gambar 19.57 Konfigurasi dengan 0

Centang pada partisi yang akan digunakan untuk swap



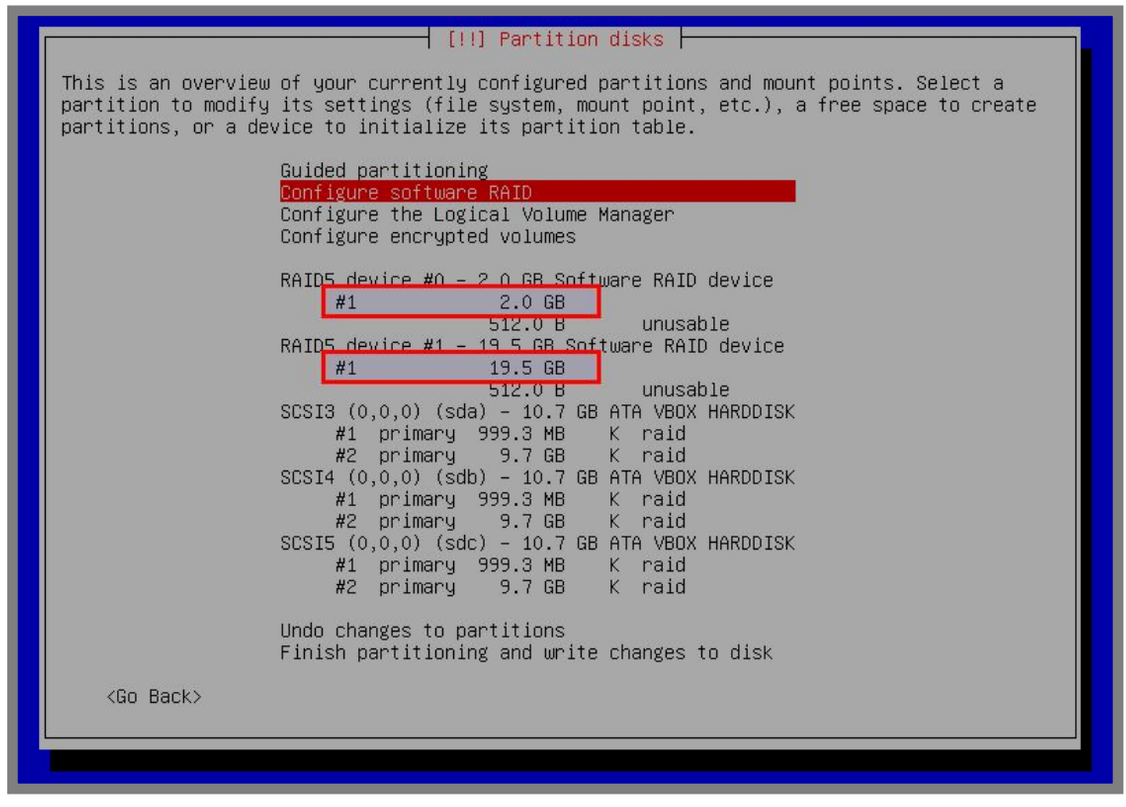
Gambar 19.58 Memilih partisi untuk swap

Lakukan langkah yang sama seperti diatas untuk membuat partisi root



Gambar 19.59 Memilih partisi untuk root

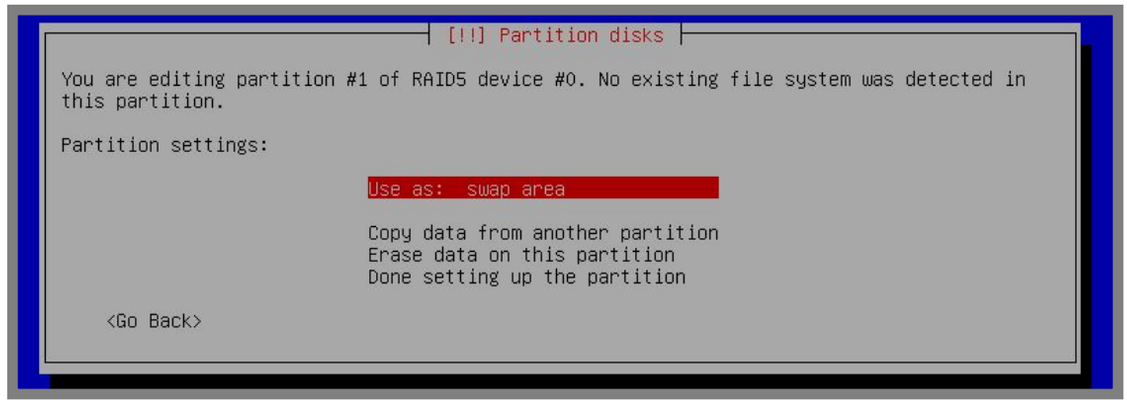
Selanjutnya klik *finish* dan berikut hasilnya



Gambar 19.60 Hasil konfigurasi software raid

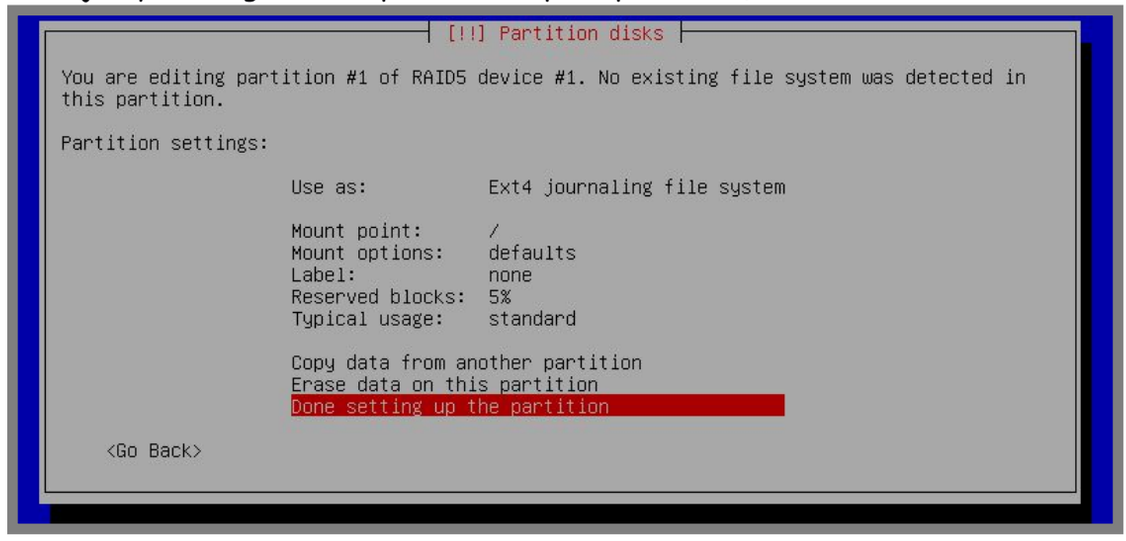
Perhatikan bahwa saat ini kita memiliki dua partisi dengan kapasitas 2GB dan 19,5GB. Nantinya kita akan menggunakan partisi 2GB untuk swap dan partisi 19,5GB untuk root.

Pilih pada partisi 2GB dan pada bagian *Use as* pilih *swap area*



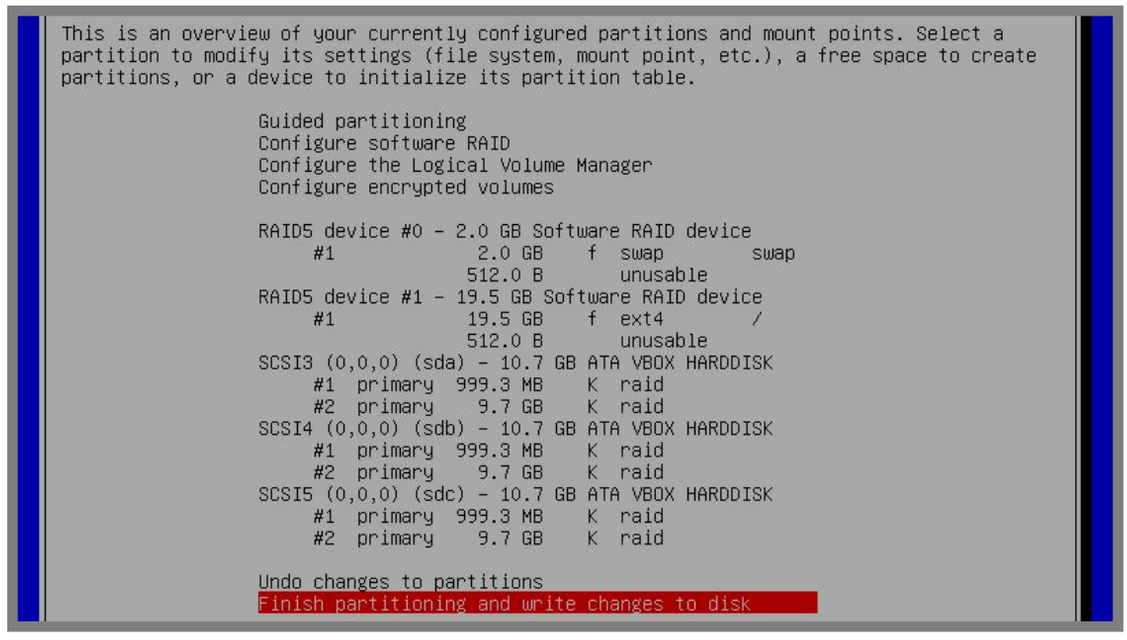
Gambar 19.61 Konfigurasi partisi swap

Selanjutnya konfigurasikan partisi root pada partisi 19,5GB



Gambar 19.62 Konfigurasi partisi root

Hasil ahirnya akan nampak seperti berikut



Gambar 19.63 Hasil ahir konfigurasi raid 5

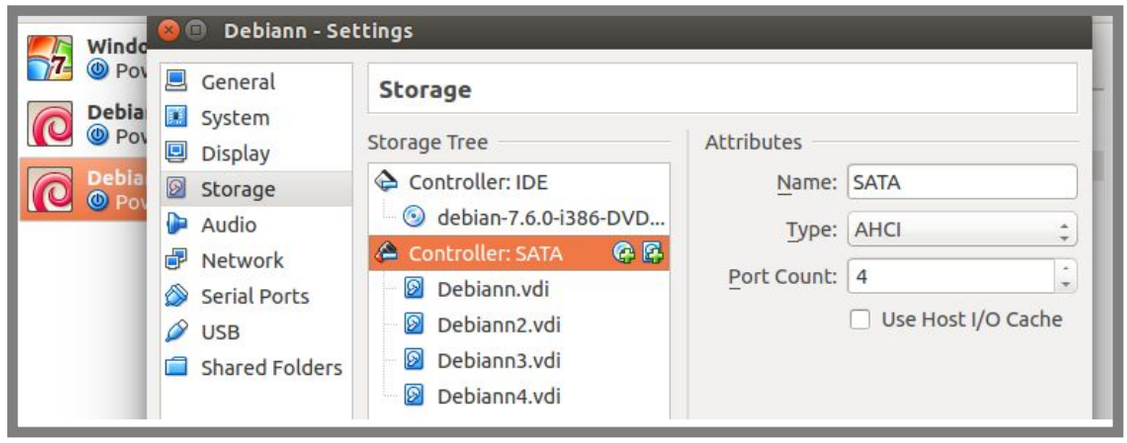
Sampai saat ini kita sudah selesai melakukan konfigurasi raid level 5, selanjutnya kita bisa melanjutkan proses installasi debian seperti biasa

RAID Level 6

RAID pada level ini memiliki teknologi yang hampir sama dengan raid level 5, hanya saja jumlah harddisk maksimum yang rusak pada level ini adalah 2. Sehingga data akan rusak dan hilang jika harddisk yang rusak melebihi 2 buah.

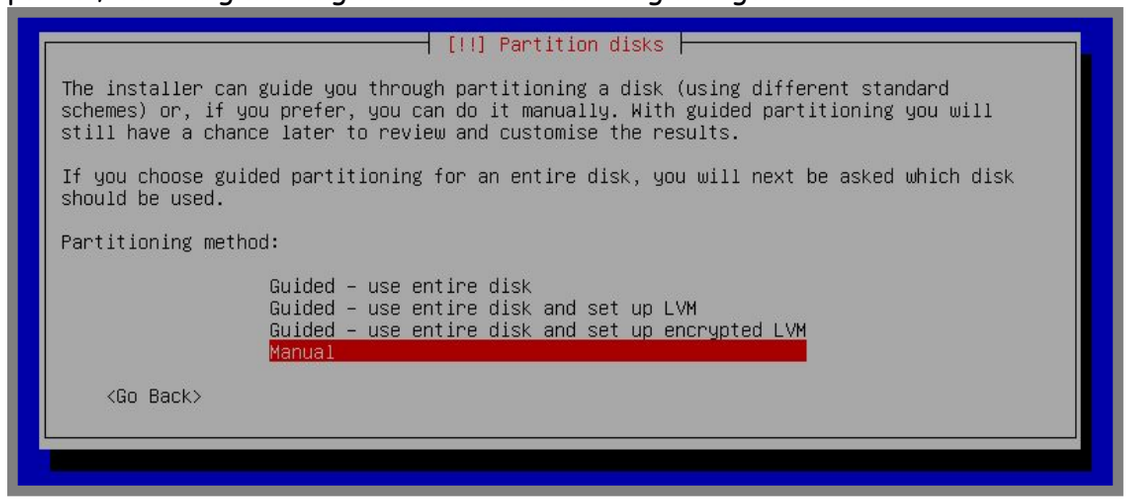
Jumlah harddisk minimum yang harus digunakan oleh raid pada level ini adalah 4. Dengan rumus kapasitas adalah *(1 - 2/n) * m.* Misal kita memiliki 4 harddisk dengan kapasitas masing-masing 10GB maka kapasitas yang diperoleh dengan raid 6 adalah (1 - 2/4) * 40 = 2/4 * 40 = 20GB

Kita akan praktik konfigurasi raid 6 menggunakan 4 harddisk dengan kapasitas masing-masing 10GB



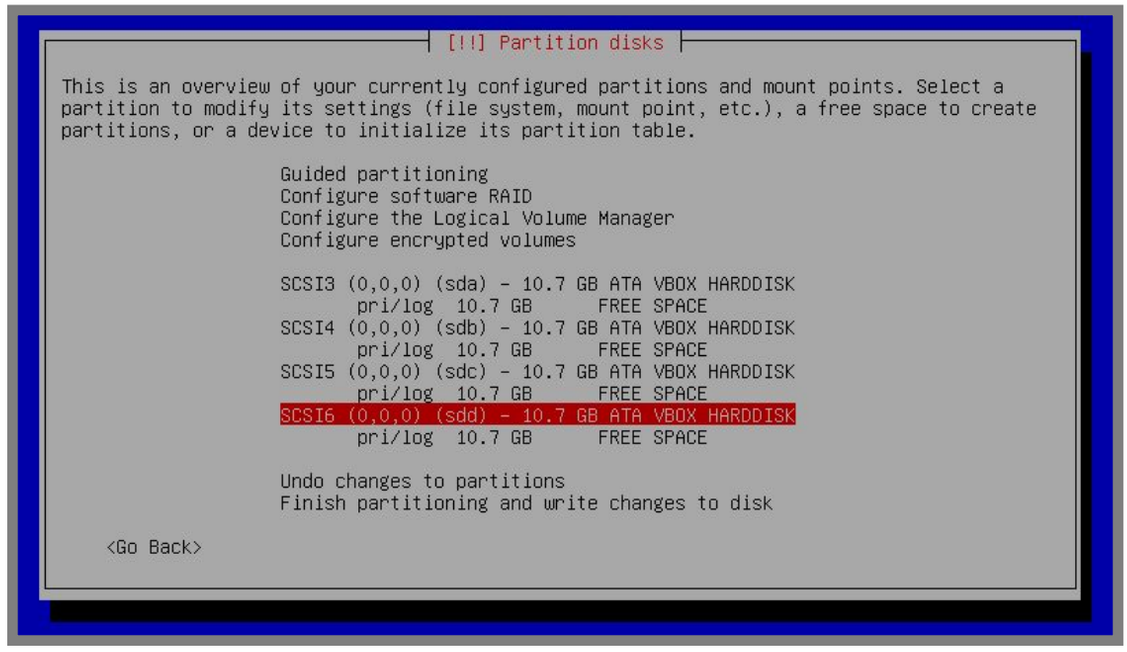
Gambar 19.64 Virtual harddisk untuk raid 6

Lakukan booting dari installaer debian seperti biasa, sedangkan pada proses partisi, ikuti langkah-langkah berikut untuk mengkonfigurasi raid 6



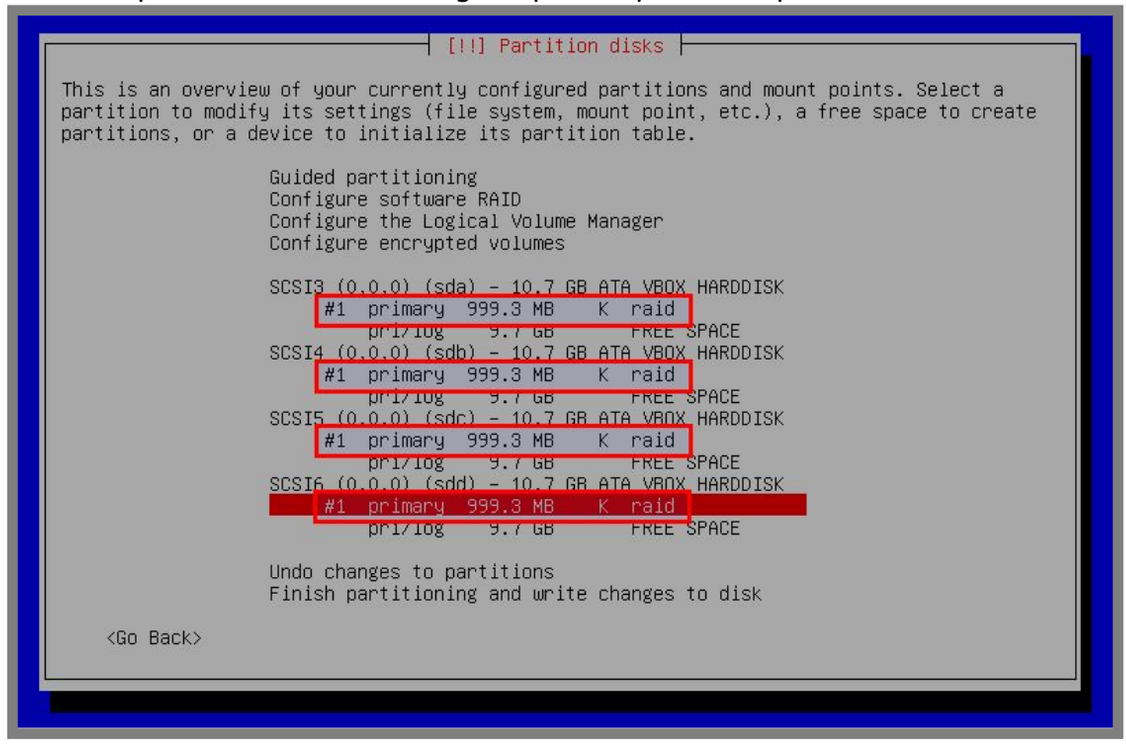
Gambar 19.65 Metode partisi manual

Buat tabel partisi baru pada keempat harddisk yang ada



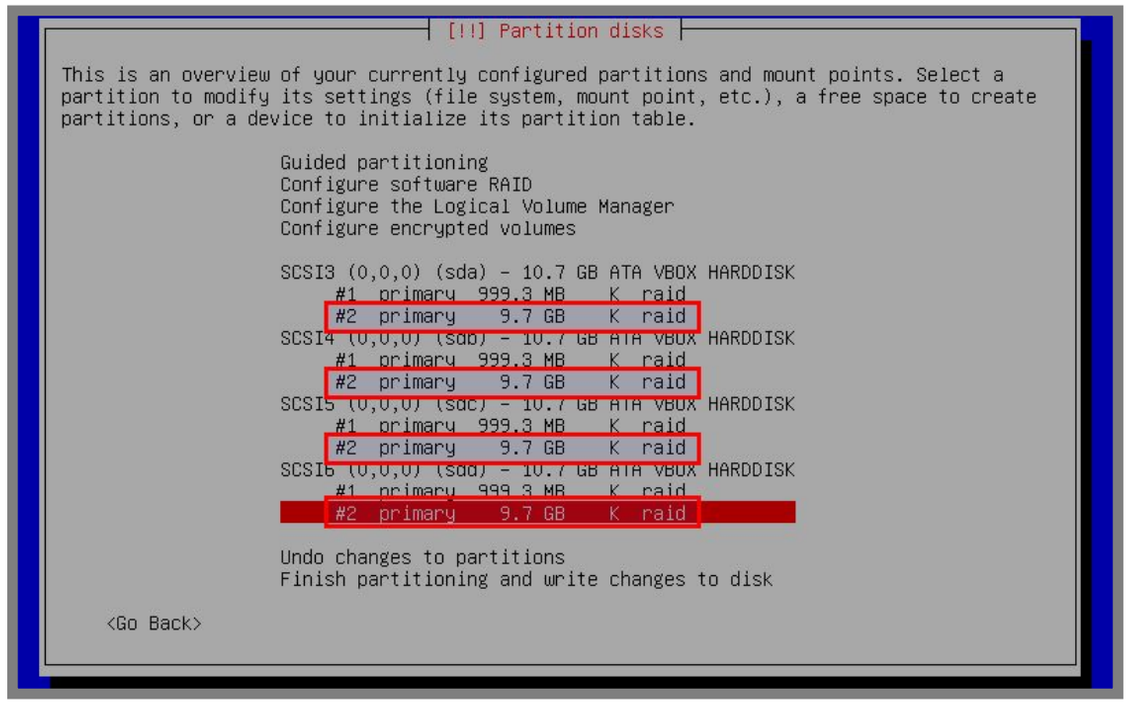
Gambar 19.66 Membuat tabel partisi baru

Kita nanti akan membuat partisi swap dengan ukuran 2GB, karena itu kita harus membuat partisi sebesar 1GB dengan tipe raid pada keempat harddisk tersebut



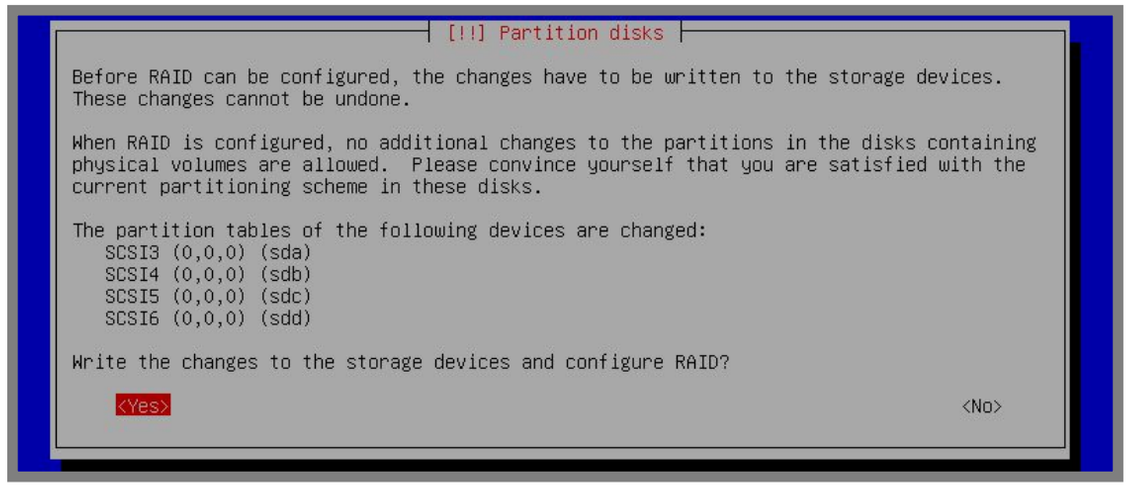
Gambar 19.67 Menyiapkan partisi untuk swap

Lakukan hal yang sama pada free space untuk partisi root



Gambar 19.68 Menyiapkan partisi untuk root

Selanjutnya pilih *Configure softwrae RAID* dan pilih *Yes*

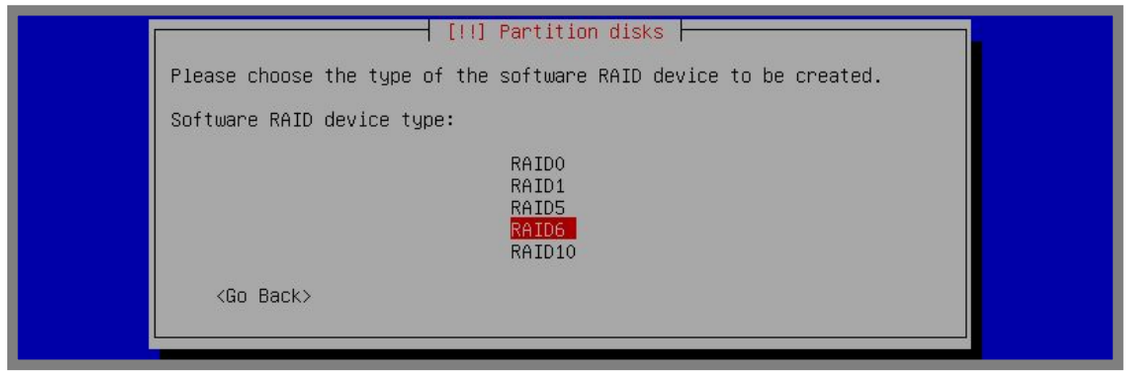


Gambar 19.69 Verifikasi konfigurasi software raid

Pilih *Create MD device*

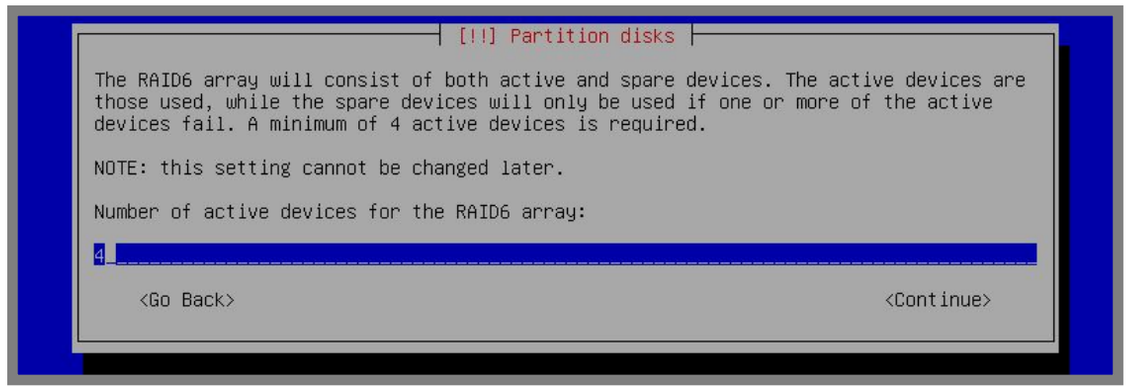
Gambar 19.70 Membuat md device

Pilih *RAID6*

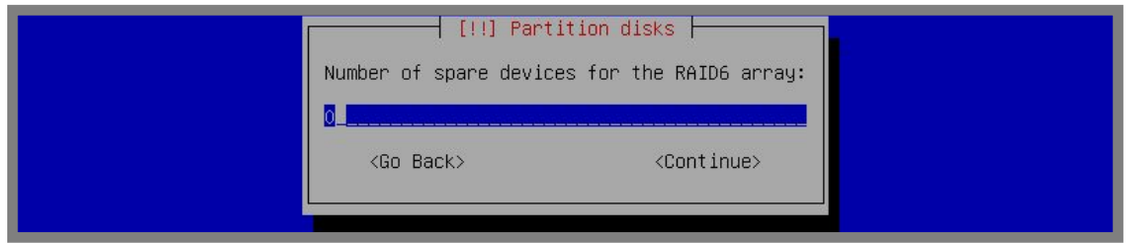


Gambar 19.71 Memilih raid level 6

Masukkan jumlah harddisk yang digunakan

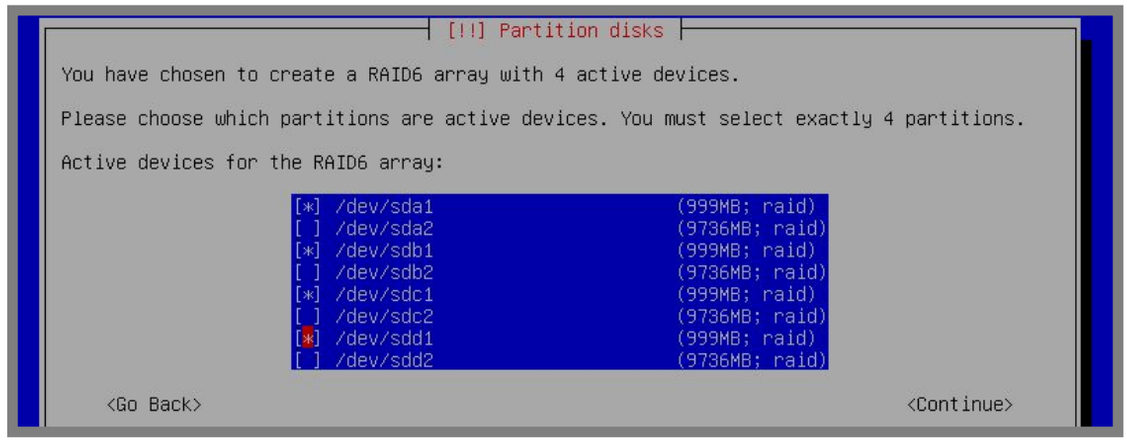


Gambar 19.72 Jumlah harddisk yang digunakan



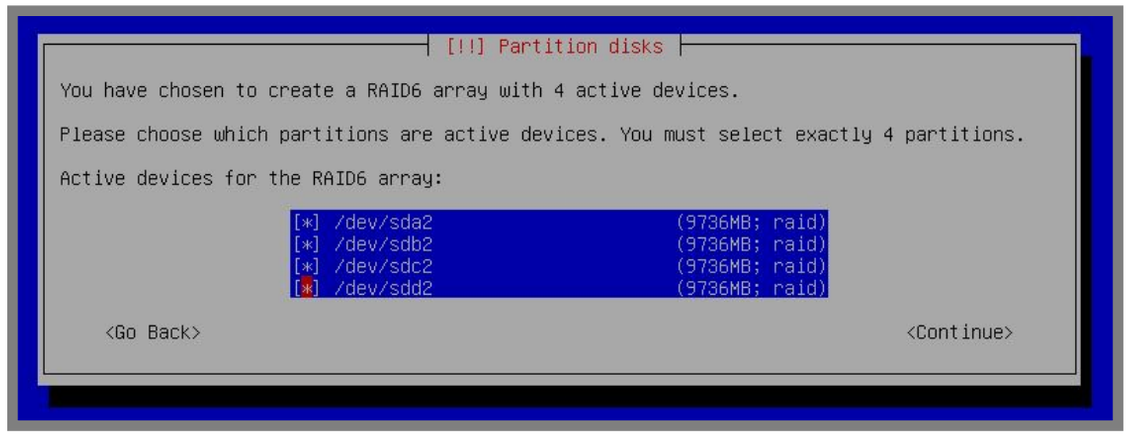
Gambar 19.73 Konfigurasi dengan 0

Centang pada partisi untuk swap



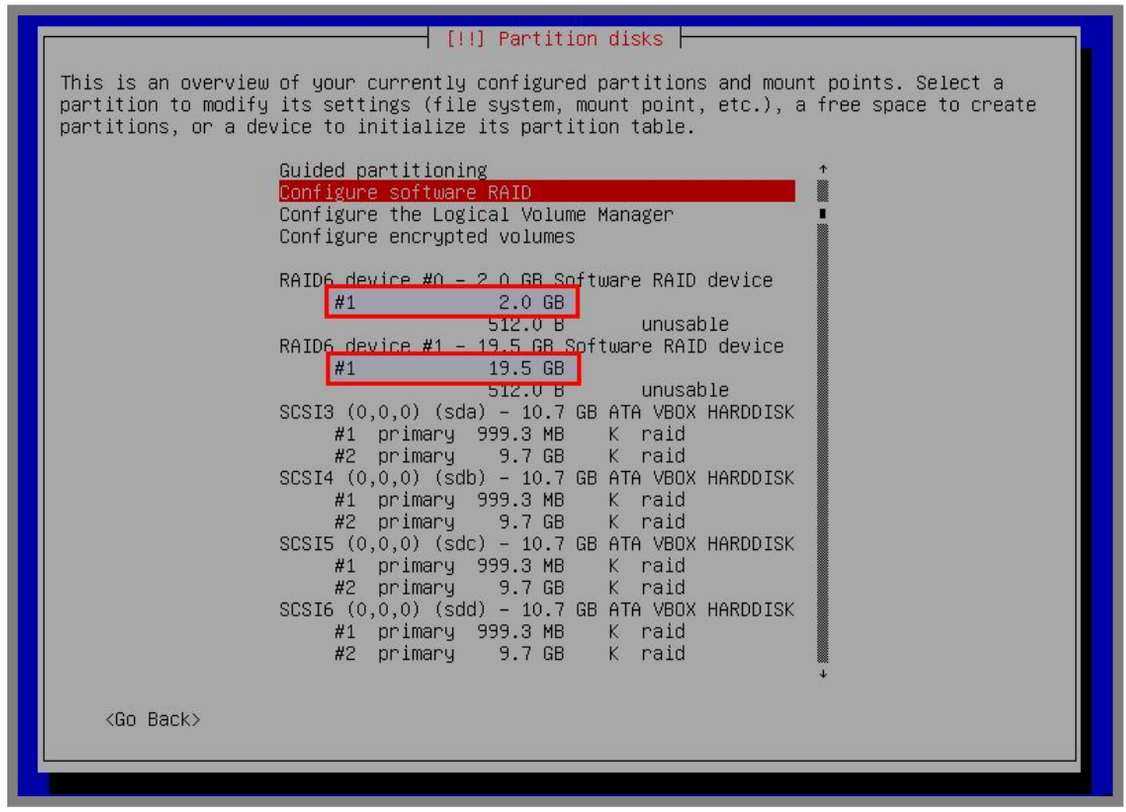
Gambar 19.74 Memilih partisi untuk swap

Lakukan langkah yang sama seperti diatas untuk membuat partisi root



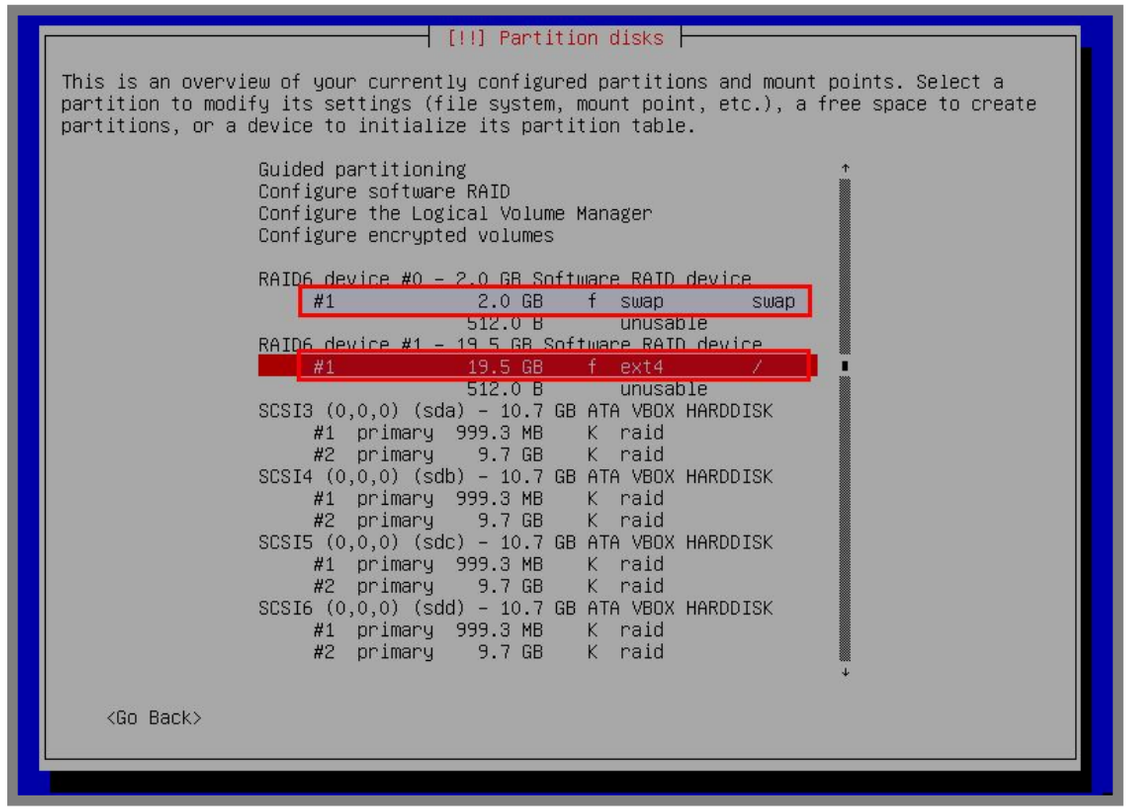
Gambar 19.75 Memilih partisi untuk root

Selanjutnya pilih *finish,* berikut hasil ahir setelah konfigurasi software raid



Gambar 19.76 Hasil ahir konfigurasi software raid

Perhatikan bahwa saat ini kita memiliki dua partisi, yaitu 2GB dan 19,5GB. Nantinya kita akan menggunakan partisi 2GB sebagai swap dan partisi 19,5GB sebagai partisi root. Lakukan konfigurasi pada kedua partisi tersebut hingga hasilnya akan nampak seperti berikut

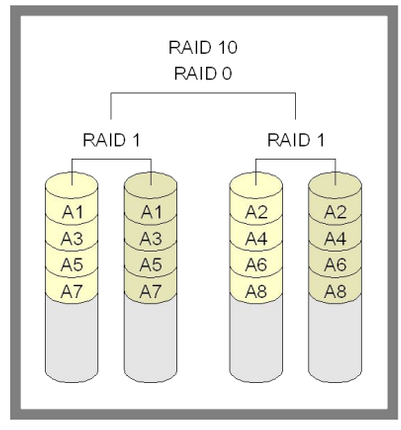


Gambar 19.77 Hasil ahir partisi dengan raid level 6

Sampai saat ini kita telah mempunyai partisi swap dengan ukuran 2GB dan partisi root dengan ukuran 19,5GB. Selanjutnya kita bisa melanjutkan proses installasi debian seperti biasa

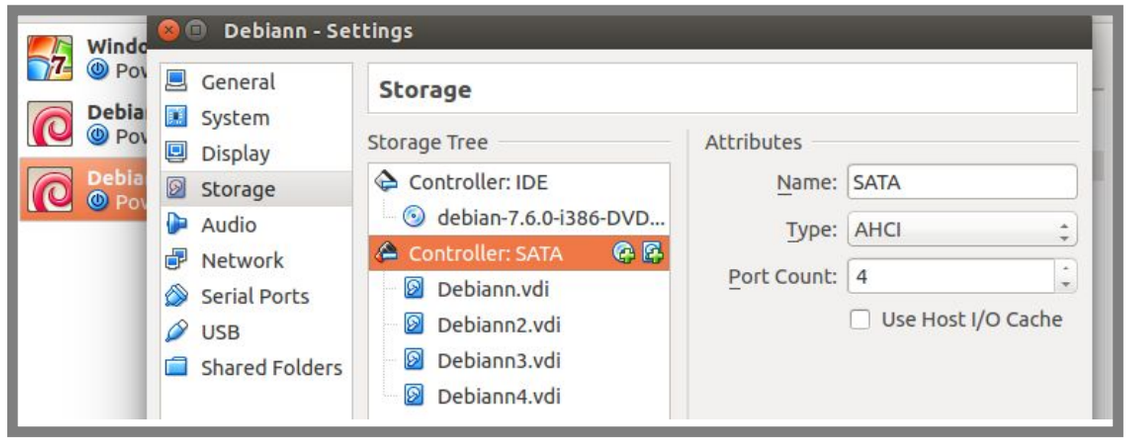
RAID Level 10

RAID level ini merupakan kombinasi dari raid level 1 dan raid level 0. Penggunaan raid level ini ditujukan untuk memperoleh performa kecepatan baca tulis pada raid 0 dengan dukungan mirroring dari raid 1. Berikut gambaran umum penerapan raid level 10



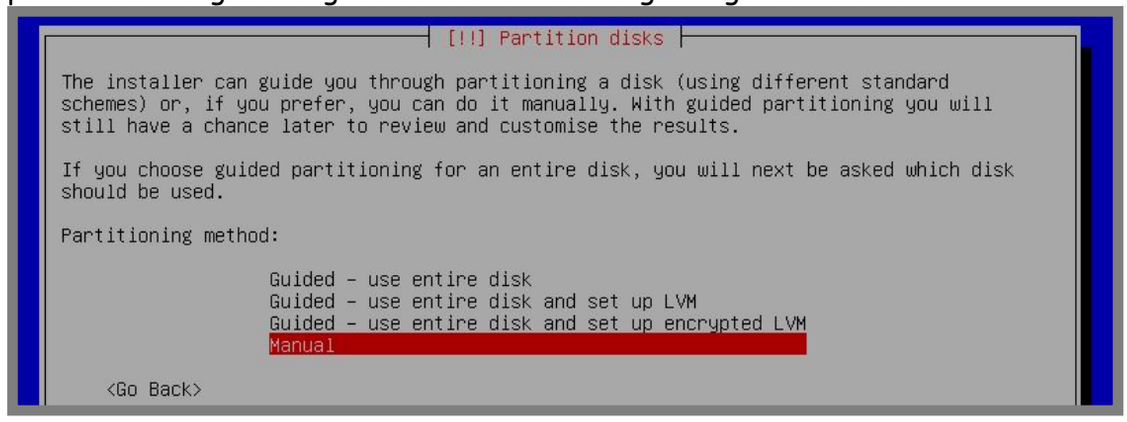
Gambar 19.78 Konsep penarapan raid 10

Dapat dilihat pada gambar diatas, bahwa minimal harddis yang dibutuhkan untuk konfigurasi raid 10 adalah 4 buah. Kita akan praktik menggunakan 4 harddisk dengan kapasitas masing-masing 10GB



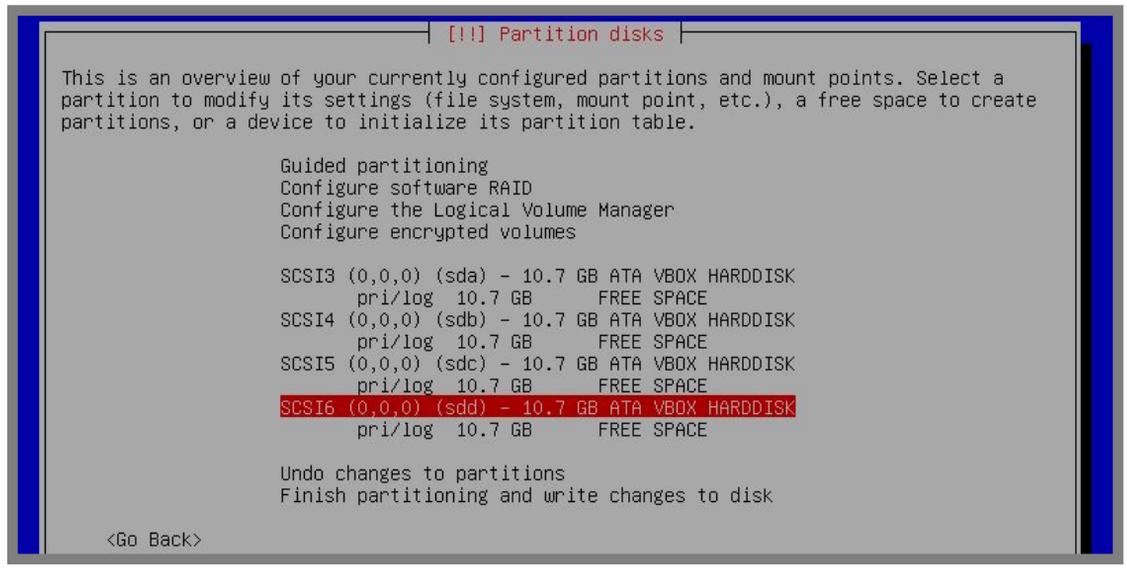
Gambar 19.79 Harddisk untuk raid 10

Lakukan booting dari installer debian seperti biasa, selanjutnya pada proses partisi ikuti langkah-langkah berikut untuk mengkonfigurasi raid 10



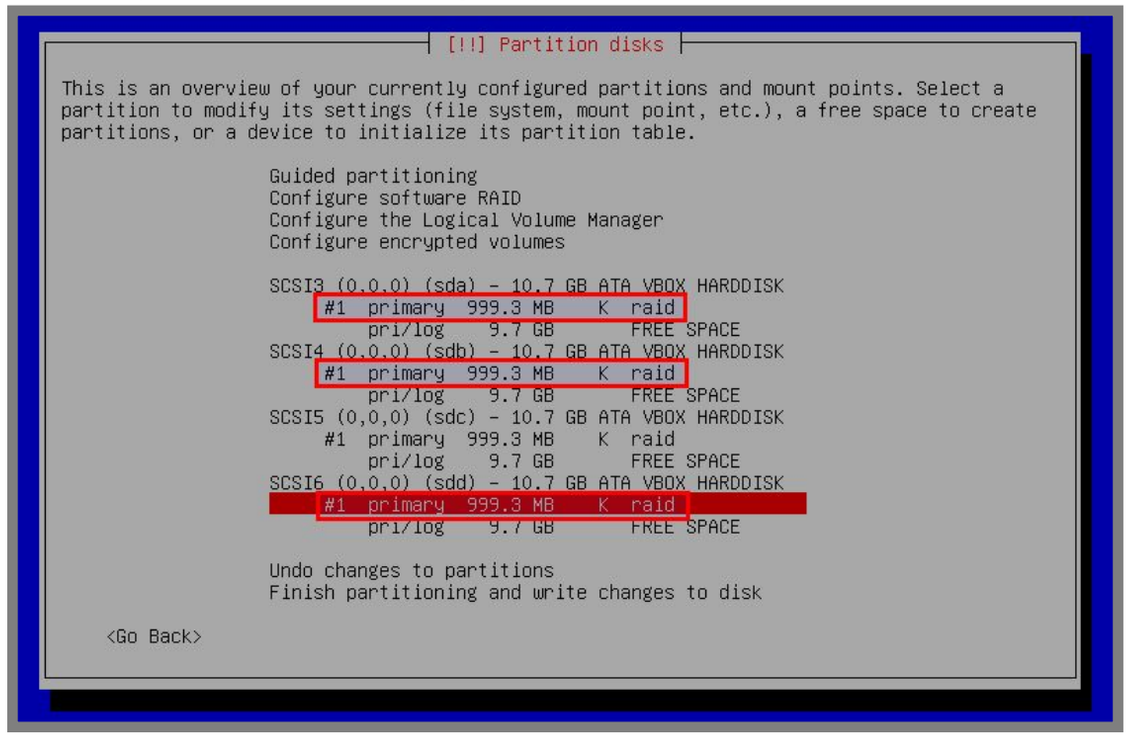
Gambar 19.80 Metode partisi manual

Buat tabel partisi baru pada keempat harddisk tersebut



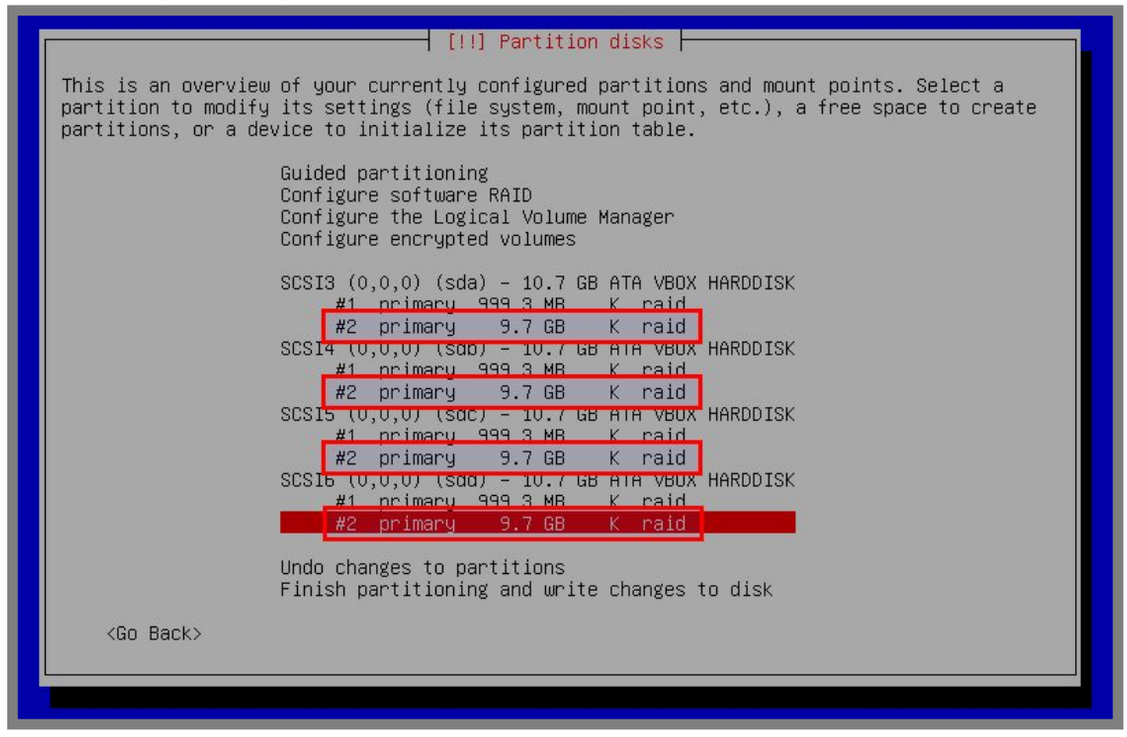
Gambar 19.81 Membuat tabel partisi baru

Diasumsikan kita ingin membuat partisi swap dengan ukuran 2GB, maka kita harus membuat partisi sebesar 1GB dengan tipe raid pada keempat harddisk tersebut



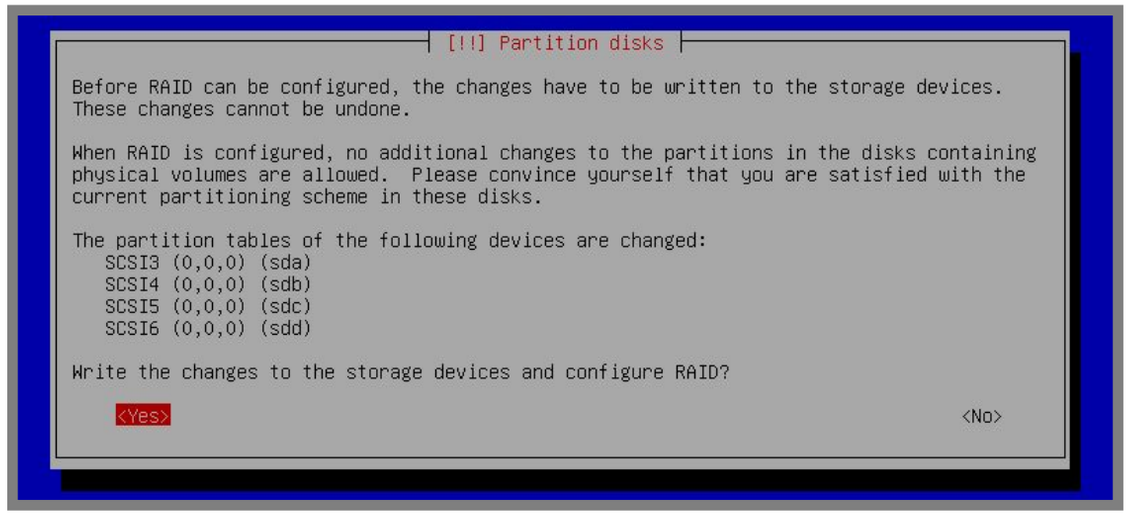
Gambar 19.82 Menyiapkan partisi untuk swap

Lakukan langkah yang sama pada free space untuk membuat partisi root



Gambar 19.83 Menyiapkan partisi untuk root

Selanjutnya pilih *Configure software RAID* kemudian pilih *Yes*

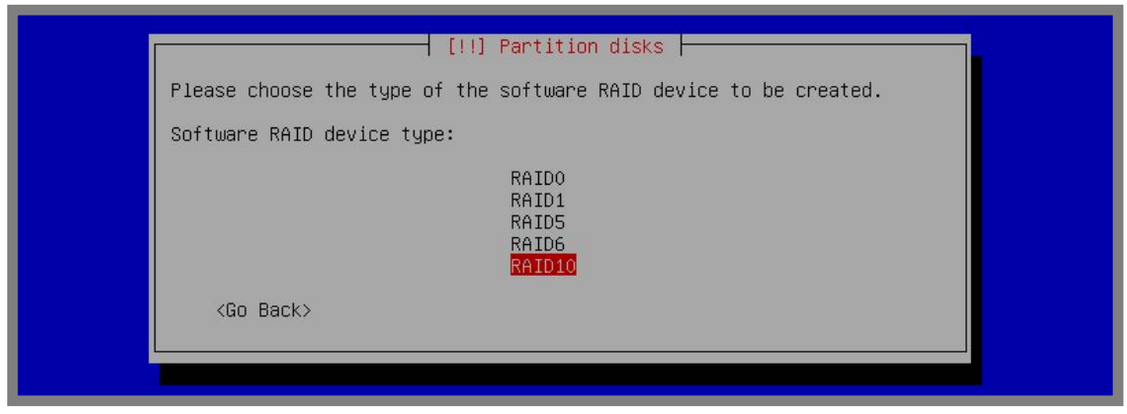


Gambar 19.84 Verifikasi konfigurasi software raid

Pilih *Create MD device*

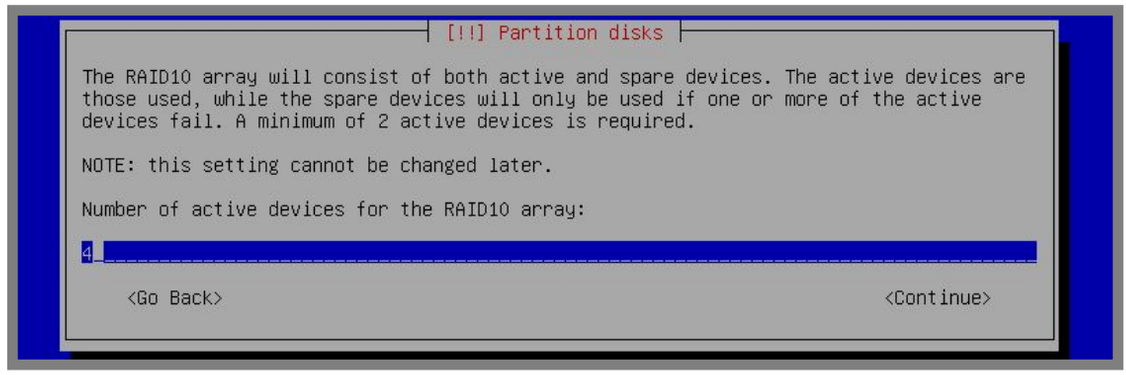
Gambar 19.85 Membuat md device

Pilih *RAID10*

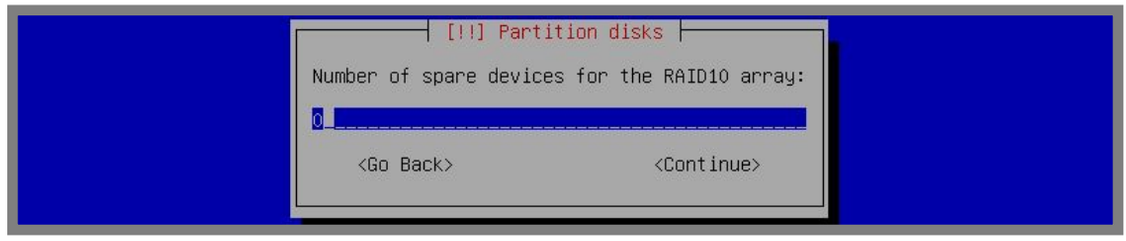


Gambar 19.86 Pilih raid level 10

Masukkan jumlah harddisk yang digunakan

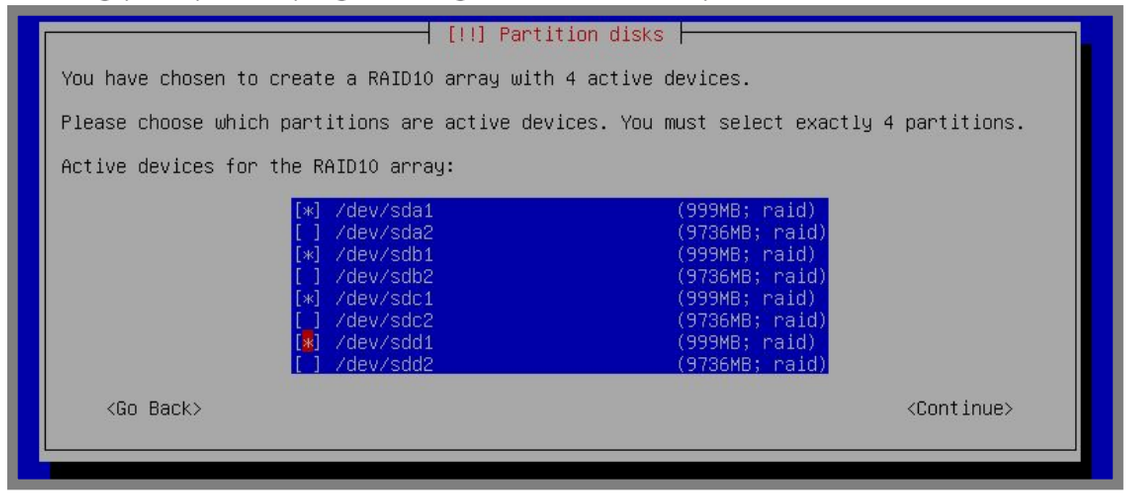


Gambar 19.87 Jumlah harddisk yang digunakan



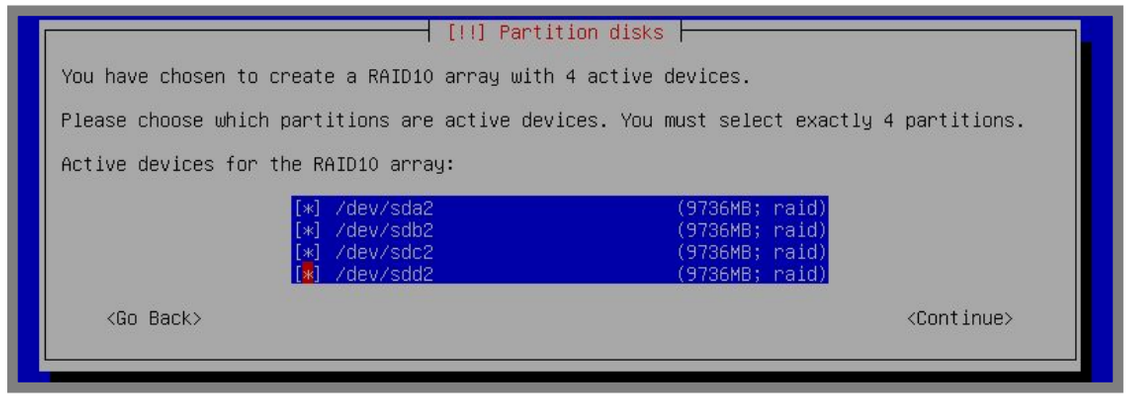
Gambar 19.88 Konfigurasi dengan 0

Centang pada partisi yang akan digunakan untuk swap



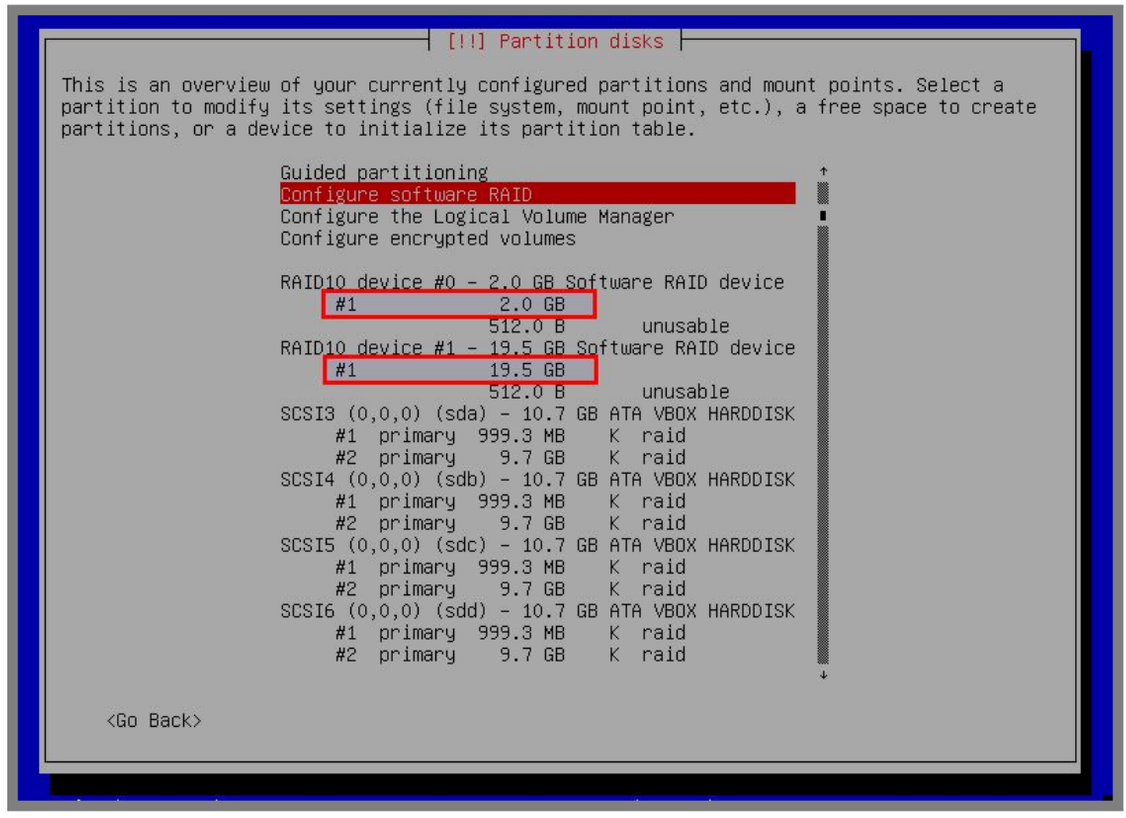
Gambar 19.89 Menyiapkan partisi untukk swap

Lakukan langkah yang sama untuk menyiapkan partisi untuk root



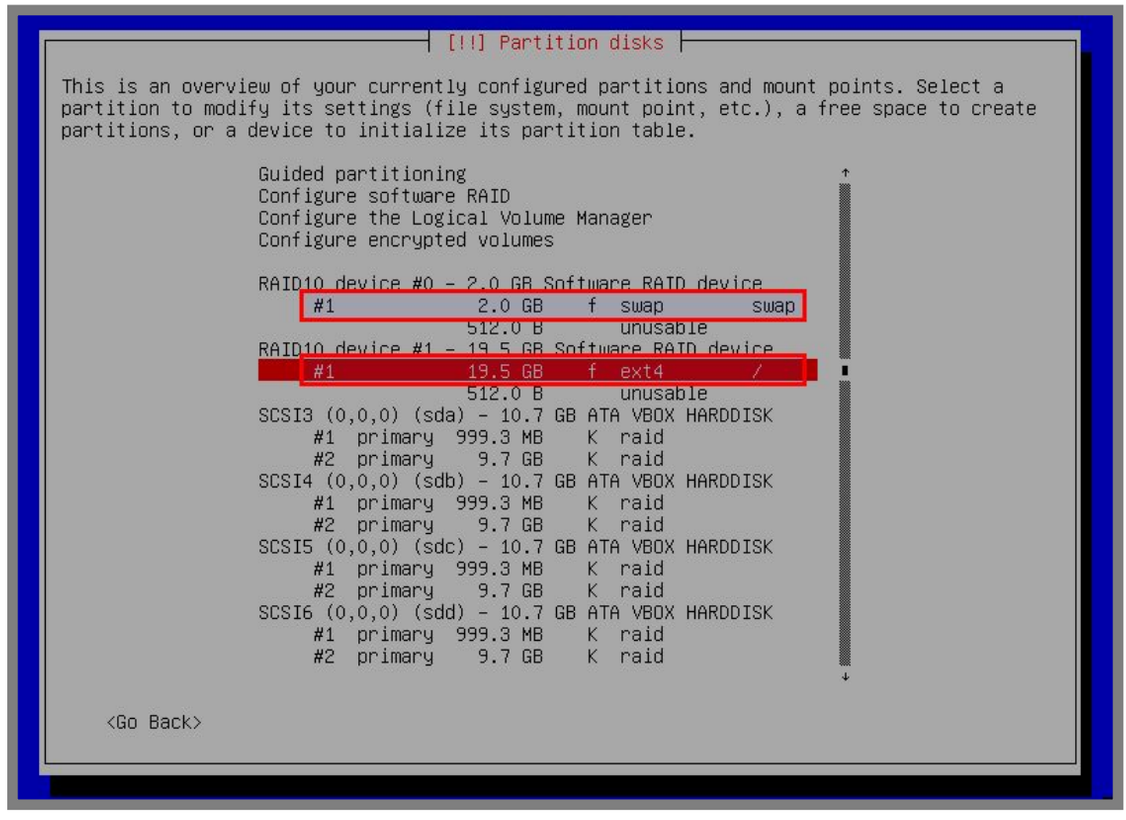
Gambar 19.90 Menyiapkan partisi root

Selanjutnya pilih *finish* dan berikut hasil ahir setelah konfigurasi software raid



Gambar 19.91 Hasil ahir konfigurasi software raid

Konfigurasikan partisi swap pada partisi 2GB dan root pada partisi 19,5 GB



Gambar 19.92 Hasil ahir konfigurasi raid 10

Perhatikan gambar diatas, terlihat bahwa saat ini kita telah mempunyai partisi swap dengan ukuran 2GB, kita juga mempunyai partisi root (/) dengan ukuran 19,5GB.

Selanjutnya kita bisa melanjutkan proses installasi debian seperti biasa

---END OF CHAPTER---

Daftar Pustaka

Beginning Unix - Paul Love, Joe Merlino, Jeremy C. Reed, Craig Zimmerman
Debian 7: System Administration Best Practices - Rich Pinkall Pollei
Managing Raid on Linux - O'reilly
Linux iptables - Gregor N. Purdy
Mikrotik Kungfu 2 - Rendra Towidjojo
60 Menit Belajar Monitoring - Syamsudin M
Self Signed Certificate - Endy Muhardin
DNS Filtering BIND9 RPZ - katalis.web.id
RAID - padliyulian@ymail.com
Membangun Server dengan Debian 7 - Agus Prasetiyo
Konfigurasi Debian Server - Al Mansyurin
Step by Step LKS Nasional - Ahmad Imanudin
Modul KK 12 - Very Setiawan, S.Kom
Modul KK 17 - Very Setiawan, S.Kom
Tutorial Administrasi Server - Farouq Alamsyah

Autobiografi Penulis

Perkenalkan nama saya **Ahmad Rosid Komarudin**, alahmdulillahh saya dilahirkan di kabupaten kebanggaan saya, Kabupaten Blitar, saya lahir tanggal 30 Desember 1997 di Kabupaten Blitar, lebih tepatnya di Desa Nglegok.. Saya adalah anak pertama dari 4 bersaudara.

Saya punya bapak dengan nama Muhammad Fauzi, beliau menikah dengan seorang wanita dengan nama Siti Khotimah. Beliau berdua lah yang telah mengajari saya tentang hidup, beliau yang mengajarkan bagaimana memecahkan misteri dunia yang tiada ahirnya ini. *Pleas pray for my father and mother,..!*

Saya adalah salah satu lulusan dari SMK termuda di Kabupaten Blitar, yaitu SMK Negeri 1 Nglegok. Di SMK ini, saya adalah siswa angkatan 5 dan lulus bersamaan dengan diterbitkannya ebook ini, yaitu tahun 2016. Saya mengambil bidang keahlian Teknik Komputer dan Jaringan di SMK ini.

Semasa di sekolah, saya aktif dalam salah satu organisasi yang dibentuk guru besar saya, Very Setiawan, S.Kom. Organisasi ini memiliki nama resmi KITS (Komunitas IT SMKN 1 Nglegok).

Saya bisa menemukan betapa luasnya dunia bersama organisasi ini, banyak sekali pelajaran-pelajaran yang saya dapatkan dari komunitas ini, betapa indahnya kekeluargaan, kebersamaan, betapa banyak orang-orang yang awalanya biasa saja bisa menjadi hero yang sangat hebat. *Thanks to Mr. Very Setiawan, Mas Heru, Mas Farouq, Mas Fandi,,, thanks to all member of KITS!!*

Banyak sekali guru yang telah mengajari saya, mendidik saya, memberi saya panutan, memberi saya semangat dan motivasi. Salah satu guru terhebat saya adalah Bapak Very Setiawan, S.Kom.

Beliau adalah salah satu guru di SMKN 1 Nglegok, beliau juga menjabat sebagai Kepala Jurusan Teknik Komputer Jaringan sekaligus Pembina Komunitas IT SMKN 1 Nglegok. Beliau adalah sosok yang telah membina saya, membina anggota KITS dari nol hingga menjadi hero.

Harapan saya, semoga semua guru-guru saya, entah guru pendidikan formal, guru ngaji, guru-guru diinternet, panutan saya, semoga beliau semua mendapat penghargaan yang setinggi-tingginya dari Allah SWT.

Cita-cita?? Mungkin hanya satu, memecahkan misteri dunia yang tak ada habisnya ini, :D. Ya,. semoga saja kita semua selalu mendapat perlindungan dan ridho dari Allah SWT.

Saya punya moto, Seperti filosofi tanah **"Bermanfaat untuk orang sebanyak mungkin, namun dikenal oleh orang sesedikit mungkin"**. Pernahkah kita mengingat jasa tanah? Sedetik saja dalam sehari? Saya rasa hanya sedikit sekali yang mengingatnya, berbeda dengan matahari yang jasanya selalu kita ingat. Padahal kita semua tentu tahu bagaiman jadinya dunia ini jika tidak ada tanah. *Be helpful for others….*

Dhawuh dari Pak Vhe

Hal yang tidak akan hilang dimakan waktu adalah ilmu yang tertuang dalam buku. Karena buku merupakan ketajaman pemikiran seseorang yang dituangkan dengan imajinasi dan konsep yang jelas dan bentuk kreatifitas yang luas.

Dalam buku "For KITS Book - Administrasi Server Jaringan dengan Debian Wheezy" ini penulis, Ahmad Rosid Komarudin menuangkanya dengan sangat detail dan konseptual yang dibubuhi contoh-contoh nyata yang bisa digunakan.

Penulisnya sendiri Ahmad Rosid Komarudin adalah seserang siswa, lulusan KITS TKJ SMKN 1 Nglegok, yang mumpuni dibidangnya, murid kebanggaanku yang selalu kusayangi. Dia merupakan tipikal seseorang yang tidak pernah menyerah dalam belajar, semoga kesuksesan dan kemanfaatan selalu mengiringi langkahnya dan setiap yang dia kerjakan.

Saya yakin seseorang pemulapun akan cepat menguasai buku ini, apalagi seorang IT Profesional sangat cocok untuk menjadikan buku ini sebagai referensi, buku ini dipersembahkan untuk Komunitas IT SMKN 1 Nglegok (KITS) yang diharapkan dapat memberikan kemanfaatan dan acuan bagi yang mau belajar lebih dalam tentang administrasi server, maka bacalah, pahami, coba, dan terapkan dan jangan lupa sampaikan kembali materi yang anda terima, supaya materi anda tidak mengendap di angan-angan anda melainkan terasah dan bermanfaat, sehingga mata rantai kemanfaatan penulispun juga tersalurkan.

Tidak ada kata-kata lagi yang bisa saya sampaikan selain luar biasa terhadap buku karya Ahmad Rosid Komarudin ini, semoga cita-citaku dan cita-citanya untuk bermanfaat bagi orang lain selalu istiqomah dan mendapatkan keridhoan Allah SWT.

Terakhir ingin saya sampaikan ucapan terima kasih yang mendalam kepada muridku Ahmad Rosid Komarudin atas kesediaan waktunya untuk menulis buku ini, setiap kerja keras,dan belajar yang telah kau rangkum dalam buku ini, bagiku kau adalah salah satu murid terhebat yang pernah kupunya, dan murid yang mampu melampauiku dan menjadikanku bersyukur atas itu, semoga diluar sana kau tak pernah berhenti untuk belajar dan mengamalkanya untuk kemasalahatan orang banyak, ingatlah ini bahwa "Berbuatlah hal yang bermanfaat maka kesuksesan itu sendiri yang menghampirimu". Untuk seluruh anak-anak KITS jangan lupakan jasa Ahmad Rosid Komarudin dengan cara pelajari selalu buku ini dan kembangkan sebaik-baiknya, saya tunggu karya-kraya yang lain.

Wish You success in your live, Be Helpful fo the others and dont forget, "Practice Make Better".
*A "**Good Book**" for a "**Good KITS**"*