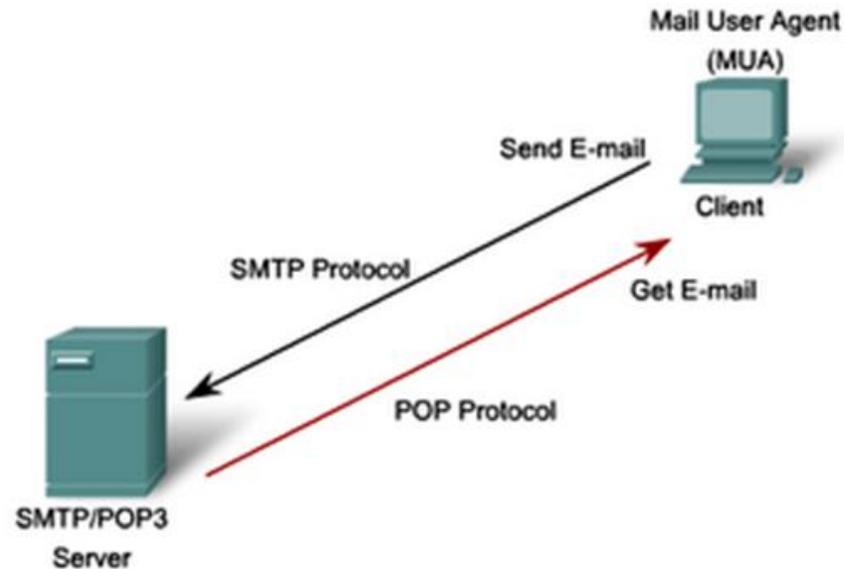


# LAPISAN APLIKASI SMTP, POP, DAN IMAP

Budhi Irawan, S.Si, M.T

# SKEMA SEND – RECEIVE E-MAIL

---



**Client** Kirim e-mail ke sebuah **server** menggunakan **SMTP** dan menerima e-mail menggunakan **POP3**

# APLIKASI E-MAIL CLIENT

---

- Ketika user menulis pesan e-mail, biasanya menggunakan aplikasi yang disebut *Mail User Agent (MUA)*, atau **e-mail client**.
- **MUA** memungkinkan untuk **mengirim pesan** dan **menyimpan pesan** yang **diterima** ke dalam *mailbox*, keduanya merupakan proses yang berbeda.

# APLIKASI E-MAIL CLIENT

---

- Agar dapat *menerima pesan e-mail* dari server, *e-mail client* menggunakan protokol *POP*.
- *Mengirim e-mail* baik dari **client** atau **server** menggunakan *format* dan *perintah string* yang didefinisikan oleh protokol SMTP.
- Biasanya *e-mail client* menyediakan *fungsi* dari kedua protokol dalam **satu aplikasi**.

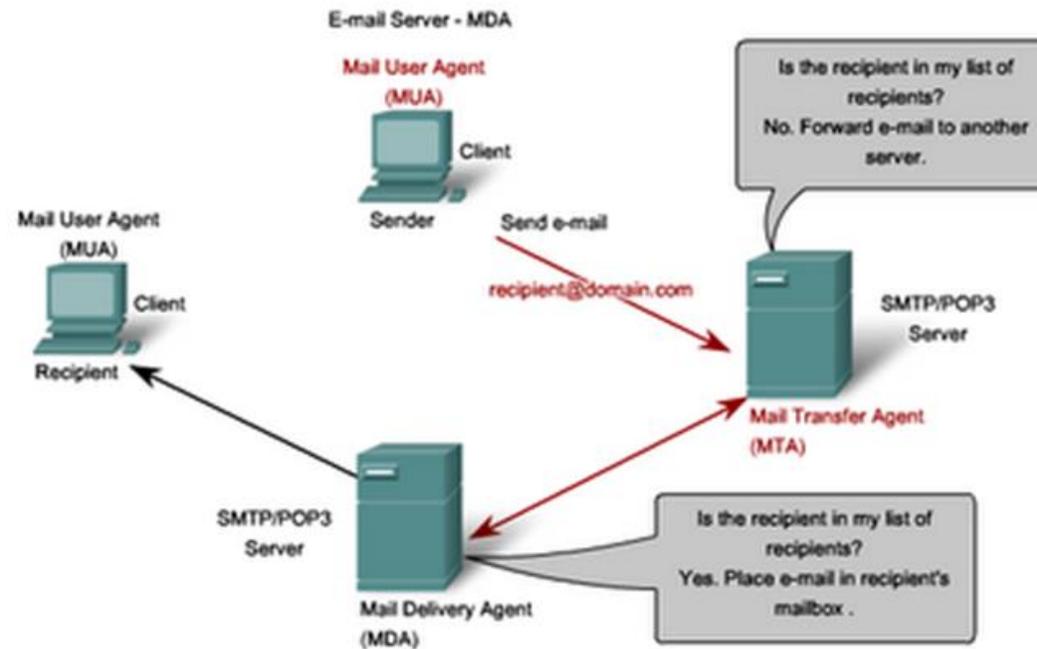
# PROSES E-MAIL SERVICE

---

*E-mail server* mengoperasikan dua proses terpisah yaitu :

- Mail Delivery Agent (MDA)
- Mail Transfer Agent (MTA)

# SKEMA PROSES MDA DAN MTA



**MDA** mengatur *proses pengiriman email* antara **Server** dan **Client**

# PROSES MDA DAN MTA

---

1. MTA adalah *proses yang digunakan untuk mem-forward e-mail* dan *MTA* menerima pesan dari *MUA* atau dari *MTA* lainnya yang berasal dari *e-mail server yang lain*.
2. Berdasarkan pesan **header**, hal itu menentukan bagaimana pesan harus diteruskan untuk mencapai **tujuan**.

# PROSES MDA DAN MTA

---

- a. Jika email ditujukan kepada pengguna yang berada pada server lokal, *email* tersebut akan diteruskan ke MDA.
- b. Jika email ditujukan kepada pengguna tidak berada pada server lokal, *MTA* merutekan *e-mail* ke *MTA* pada *server yang tepat*.

# PROSES MDA DAN MTA

---

3. *MDA* menerima e-mail dari sebuah *MTA* dan melakukan *pengiriman aktual*.
4. *MDA* menerima semua *inbound mail* dari *MTA* dan menempatkannya ke *user mailbox* yang sesuai.
5. *MDA* juga dapat menyelesaikan masalah pengiriman final, seperti *virus scanning*, *spam filtering*, dan *return-receive handling*.

# PROTOKOL POP DAN POP3

---

- POP dan POP3 (Post Office Protocol, version 3) adalah *protokol pengiriman email inbound* dengan tipikal *client/server protocol*.
- POP dan POP3 mengirimkan e-mail dari *server ke client (MUA)*.
- *MDA* mendengarkan ketika ada *client terhubung ke server* dan setelah koneksi terbentuk, *server dapat mengirimkan e-mail ke client*.

# SERVER & PROTOKOL POP3

---

- *Server POP3* menyimpan sementara email tiap user di dalam *mailboxnya* masing – masing *sebelum di-download* oleh user bersangkutan menggunakan *email client* seperti Outlook ataupun Eudora dan lain - lain.
- Dalam proses pengambilan tersebut *email client* terhubung ke **mail server** menggunakan protokol POP3 yang berjalan pada **TCP port 110**

# Perintah POP3

---

- Dalam berhubungan dengan server, client POP3 menggunakan beberapa perintah sebagai berikut :
  - **Stat** : meminta informasi jumlah pesan yang tersedia
  - **List** : menentukan ukuran setiap pesan yang akan diambil
  - **Retr** : mengambil pesan yang terdapat di server
  - **Quit** : mengakhiri session POP3

# PROTOKOL SMTP

---

- SMTP (*Simple Mail Transfer Protocol*) merupakan salah satu protokol yang umum digunakan untuk *pengiriman surat elektronik* di Internet.
- Protokol ini dipergunakan untuk *mengirimkan data* dari *komputer pengirim surat elektronik (MUA)* ke *server surat elektronik penerima*.

# PROTOKOL SMTP

---

- *SMTP* pertamakali didefinisikan oleh RFC 821 (1982, sering dikenal sebagai *STD10*) dan terakhir kali diperbaharui oleh RFC 5321 (2008) dimana terdapat *Extended SMTP (ESMTP)* tambahan.
- *SMTP* lebih spesifik untuk transportasi email keluar dan menggunakan *port TCP*.

# ANALOGI SMTP

---

- *SMTP* bisa dianalogikan sebagai **kantor pos**. Ketika *mengirim sebuah e-mail*, komputer akan mengarahkan e-mail tersebut ke sebuah *SMTP server*, untuk diteruskan ke *mail-server tujuan*.
- *Mail-server tujuan* ini bisa dianalogikan sebagai **kotak pos** di pagar depan rumah, atau *kotak PO BOX* di kantor pos.

# ANALOGI SMTP

---

- *Email-email* yang terkirim akan menempati di tempat tersebut hingga si pemiliknya mengambilnya.
- Urusan *pengambilan e-mail* tersebut tergantung kapan di penerima memeriksa *account e-mailnya*.

# KONEKSI KEAMANAN SMTP

---

- Koneksi SMTP diamankan oleh **SSL (Secure Sockets Layer)** yang lebih dikenal sebagai **SMTPS**.

# FUNGSI LAIN SMTP

---

- *SMTP*, di sisi lain *mengatur transfer outbound e-mail dari klien yang mengirimkan email ke server (MDA) serta transportasi e-mail antara server e-mail (MTA).*
- *SMTP* memungkinkan *e-mail* diangkut melalui *jaringan data* antara berbagai jenis perangkat lunak *server* dan *client* dan membuat *e-mail exchange* melalui Internet.

# FORMAT PESAN SMTP

---

- Format pesan protokol SMTP menggunakan *command yang baku beserta reply-nya*.
- Command atau perintah-perintah ini mendukung prosedur yang digunakan dalam *SMTP*, seperti *sesi inisiasi, transaksi email, forwarding mail, verifikasi nama-nama mailbox, mengembangkan milis* serta *pertukaran antara pembukaan dan penutupan*.

# PORT SMTP

---

- *SMTP* bekerja berdasarkan pengiriman *end-to-end*, dimana *SMTP client* (pengirim) akan menghubungi *SMTP server* (penerima) untuk segera mengirimkan *email*.

# Port SMTP

---

- *SMTP server* melayani pengguna melalui **port 25 (TCP)**, dimana setiap pesan yang dikirimkan melalui *SMTP* harus memiliki :
  - Header atau amplop, yang dijabarkan pada **RFC 822**.
  - Konten atau isi, yang berisi tentang isi dari surat yang akan dikirimkan

# COMMAND PESAN SMTP

---

- **HELO** - mengidentifikasi proses client SMTP ke proses server SMTP
- **EHLO** - versi yang lebih baru dari HELO, yang mencakup perluasan layanan
- **MAIL FROM** - Mengidentifikasi pengirim
- **RCPT TO** - Mengidentifikasi penerima
- **DATA** - Mengidentifikasi tubuh pesan

# Contoh Transaksi e-mail

---

S = Server C = Klien

C : (inisialisasi hubungan)

S : 220 hotmail ESMTP server ready

C : mail from: user1@hotmail.com

S : 250 mail ok

C : rcpt to: user2@hotmail.com

S : 250 rcpt ok

C : data

S : 354 end data with CRLF.CRLF

C : — email data —

C : CRLF.CRLF

S : 250 data ok

C : QUIT

S : 221 bye

# IMAP

---

- IMAP (Internet Message Access Protocol) adalah *protokol standar* untuk *mengakses / mengambil e-mail dari server*.
- *IMAP* memungkinkan pengguna *memilih pesan e-mail* yang akan *diambil*, membuat *folder di server*, mencari pesan *e-mail* tertentu, bahkan menghapus pesan *e-mail* yang ada.

# IMAP

---

- Dengan *IMAP*, **mailbox** dapat dibaca dan dikelola secara **simultan** (bersamaan) oleh sejumlah **email client berbeda**.
- *IMAP* seringkali digunakan oleh sebagian besar pengguna Internet untuk mendownload *email dari web mail server*.

# IMAP

---

- Pada dasarnya Protokol *IMAP* ini dirancang agar user dapat mengakses *e-mail* pada *mailbox* serta dapat berinteraksi dengan *server*.
- *PORT* yang digunakan untuk protocol ini dalam bentuk *TCP/IP* yaitu pada **PORT 143**.

# IMAP

---

- Protokol ini menggunakan koneksi yang *terus menerus ke server* dan ketika *e-mail* masuk maka akan terlihat langsung di *e-mail* komputer *Client* (dengan posisi online).

# POP3 VERSUS IMAP

Pembanding	POP3	IMAP
Email dapat ditinggalkan di Server	Setting "leave message" harus diaktifkan	Otomatis
Email beserta attachment-nya langsung di download ke local PC	Otomatis	Setting "Download all header and attachment" harus diaktifkan
Folder yang dibuat di Outlook otomatis muncul di Webmail dan sebaliknya	Tidak	Ya
Membaca email baru	Cepat	Ada delay sekitar 1-5 detik
Rule/Filter yang dibuat di Outlook otomatis berlaku di Webmail	Tidak	Hanya rule dari Webmail yang berlaku untuk Outlook
Email Backup	Backup email harus dilakukan manual kecuali setting "leave message on server" diaktifkan	Backup email otomatis dilakukan di server
Delete email	Email yang dihapus langsung masuk ke "Deleted Folder"	Email yang dihapus akan ditandai berupa garis coret pada header-nya. Untuk menghapus secara permanent maka lakukan "Purge Deleted Messages"