Telkom University

LAPISAN APLIKASI HTTP DAN FTP

Budhi Irawan, S.Si, M.T

LAPISAN APLIKASI

- Adalah lapisan yang berhubungan dengan ***aplikasi*** atau ***program komputer*** yang digunakan oleh ***user***.
- ***Aplikasi*** atau ***Program komputer*** yang dimaksud adalah aplikasi atau program yang melakukan ***akses jaringan***, **bukan program standalone**, yaitu program yang berhubungan dengan OSI.
- Contoh : ***aplikasi email, ftp client/server, telnet***

LAPISAN APLIKASI DAN LAPISAN TRANSPOR

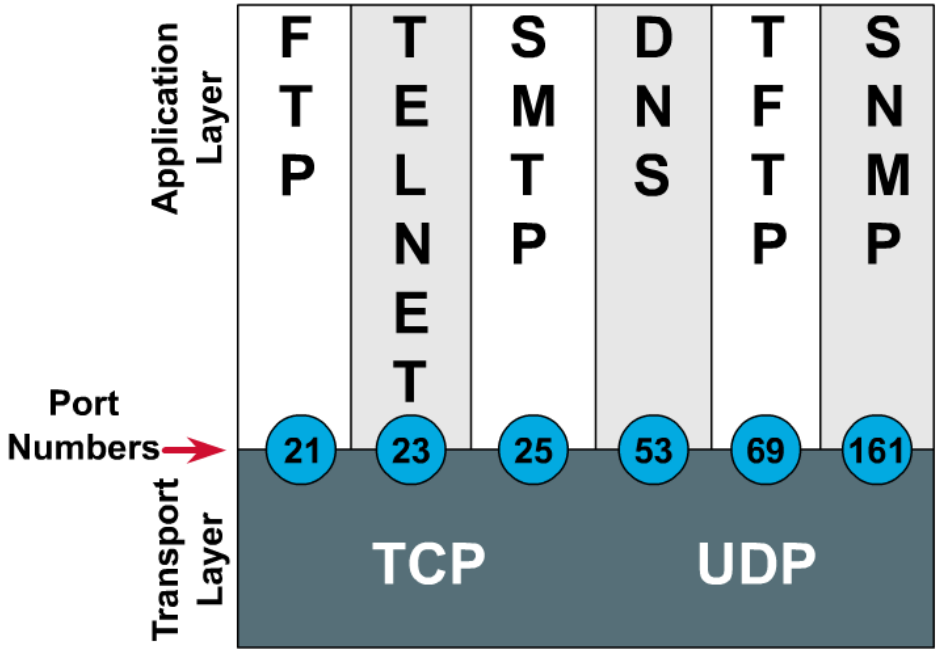
- ***Transfer data*** tergantung pada **Protokol Layer Transport** yang digunakan, bisa menggunakan ***UDP*** dan ***TCP***.
- Aplikasi yang menggunakan ***Protokol TCP*** dalam melakukan transmisi data:
  - *Hyper Text Transfer Protocol (HTTP)*
  - *File Transfer Protocol (FTP)*
  - *Terminal Emulator (Telnet)*
  - *Simple Mail Transfer Protocol (SMTP)*

LAPISAN APLIKASI DAN LAPISAN TRANSPOR

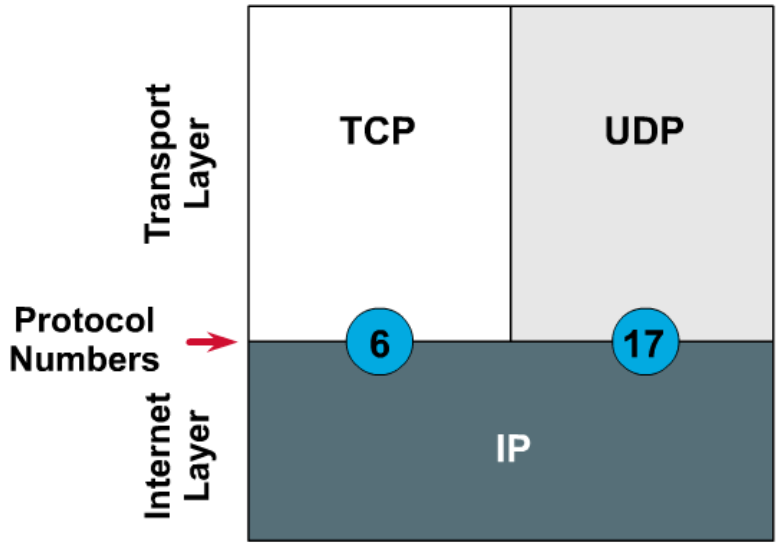
- Aplikasi yang menggunakan ***Protokol UDP*** :
  - *Dynamic Host Configuration Protocol (DHCP)*
  - *Simple Network Management Protocol (SNMP)*
  - *Trivial File Transfer Protocol (TFTP)*
- DNS menggunakan keduanya UDP dan TCP

LAPISAN APLIKASI DAN LAPISAN TRANSPOR

Port Numbers



The Protocol Field



HYPER TEXT TRANSFER PROTOCOL

- **Hypertext Transfer Protocol (HTTP)** adalah sebuah ***protokol jaringan lapisan aplikasi*** yang digunakan untuk ***sistem informasi terdistribusi***, ***kolaboratif***, dan ***menggunakan hipermedia***.

HYPER TEXT TRANSFER PROTOCOL

- Penggunaan **HTTP** diantaranya untuk ***pengambilan sumber daya yang saling terhubung dengan tautan***, yang disebut dengan **dokumen hiperteks**, yang kemudian membentuk ***World Wide Web*** pada tahun 1990 oleh fisikawan Inggris, Tim Berners-Lee.

VERSI HTTP

Hingga kini, ada dua versi mayor dari protokol HTTP, yaitu :

- **HTTP/1.0** yang menggunakan ***koneksi terpisah untuk setiap dokumen***,
- **HTTP/1.1** yang dapat menggunakan ***koneksi yang sama untuk melakukan transaksi***.

Dengan demikian, **HTTP/1.1** bisa lebih cepat karena memang tidak perlu membuang waktu untuk pembuatan koneksi berulang-ulang.

STANDAR HTTP

- Pengembangan standar **HTTP** telah dilaksanakan oleh Konsorsium **World Wide Web** (World Wide Web Consortium/W3C) dan juga **Internet Engineering Task Force (IETF),** yang berujung pada publikasi beberapa dokumen **Request for Comments (RFC)**, dan yang paling banyak dirujuk adalah **RFC 2616** (yang dipublikasikan pada bulan Juni 1999), yang mendefinisikan **HTTP/1.1.**

PROTOKOL HTTP

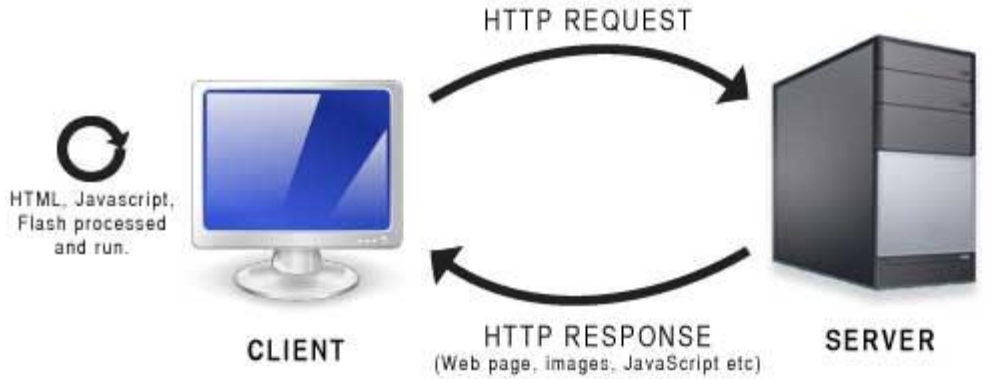
- Protokol *HTTP* adalah sebuah ***protokol meminta*** dan atau ***menjawab antara klien dan server***.
- Sebuah klien *HTTP* (seperti *web browser* atau robot dan lain sebagainya), biasanya **memulai permintaan dengan membuat hubungan ke port tertentu di sebuah server *Webhosting* tertentu** (biasanya *port* 80 atau 8080).

PROTOKOL HTTP

- Klien yang mengirimkan permintaan **HTTP** juga dikenal dengan ***user agent***.
- Server yang meresponsnya, yang menyimpan sumber daya seperti ***berkas HTML*** dan gambar, dikenal juga sebagai ***origin server***.

HTTP REQUEST & RESPONSE

- **Protokol aplikasi web**
- Menggunakan model client-server



  - **Client** : browser yang melakukan request dan menerima objek web
  - **Server** : web server yang mengirim objek sebagai respon terhadap request client
- http1.0 : RFC 1945
- http1.1 : RFC 2068

LAYANAN TCP DAN STATELESS

http: *Layanan Transport TCP*

- **Client** menginisiasi koneksi TCP (Membuat Socket) ke server **port 80**
- **Server** menerima koneksi TCP dari **client**
- Saling bertukar pesan **http** (pesan protokol lapis aplikasi) antar **browser** dengan **webserver**
- **Koneksi TCP ditutup**

http : *Tanpa State (Stateless)*

- **Server** tidak mengingat permintaan sebelumnya **client**

Contoh http:
User : www.telkomuniversity.ac.id

1. Client http membuka hubungan TCP ke www.telkomuniversity.ac.id port 80

2. Server http www.telkomuniversity.ac.id menerima hubungan dan memberitahu client

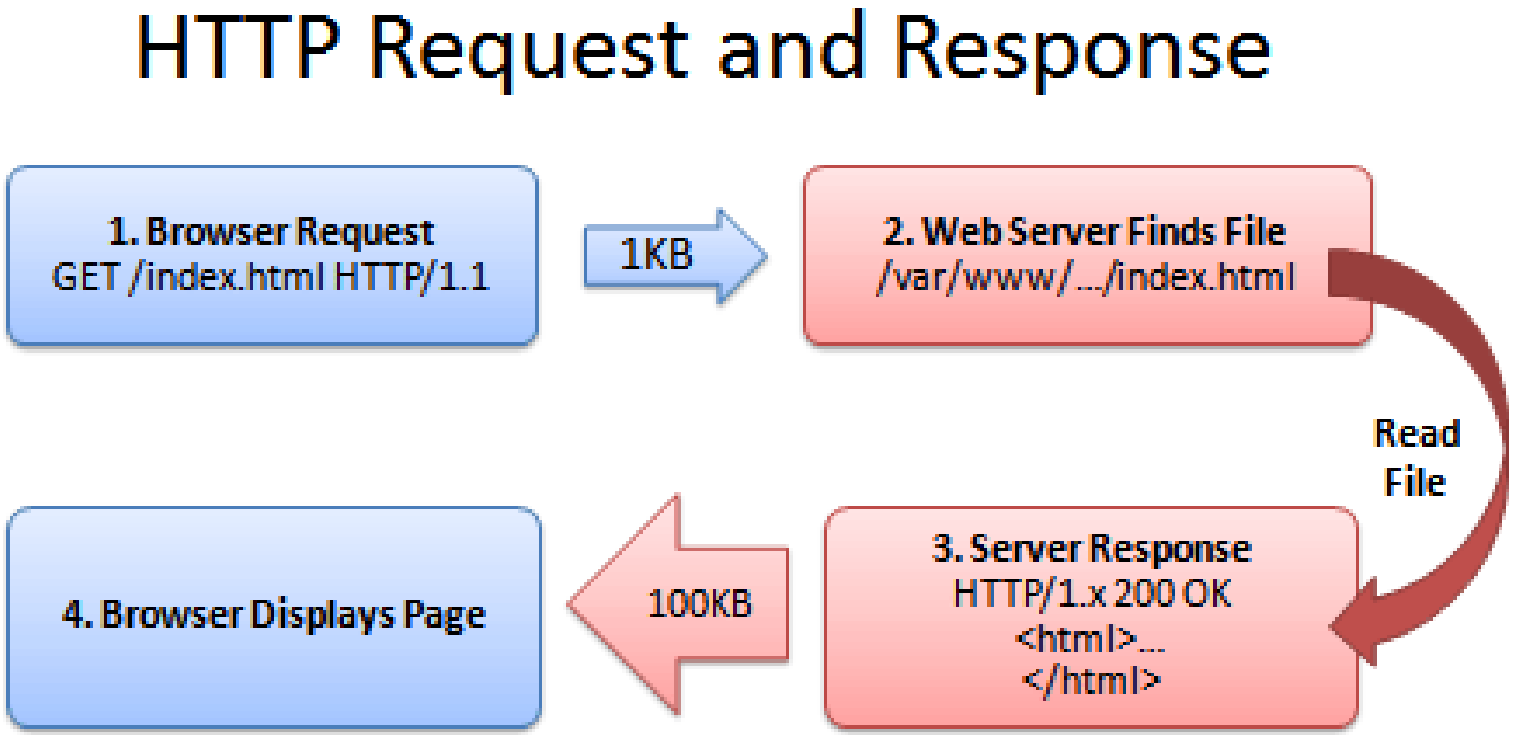
3. Client http mengirim pesan request (url) ke soket koneksi TCP

4. Server http www.telkomuniversity.ac.id menerima request dan mengirimkan web objek yang diminta

5. Client http menerima pesan berisi file html, mendisplay html, parsing html dan menemukan 10 objek jpeg referensi

6. Langkah 1 – 5 diulang untuk setiap 10 objek jpeg

SKEMA HTTP REQUEST DAN RESPONSE



KONEKSI NON-PERSISTENT DAN PERSISTENT

Non-persistent

- HTTP 1.0
- Server menelusuri (parse) request, me-respon dan menutup hubungan TCP
- Diperlukan 2 RTT untuk setiap objek
- Setiap transfer objek mengalami slow start

RTT (Round Trip Time) : Waktu yang diperlukan untuk mengirimkan sinyal ditambah waktu yang dibutuhkan untuk mendapatkan respon bahwa sinyal sudah diterima

Persistent

- HTTP 1.1
- Di koneksi TCP yang sama : server menelusuri request, me-respon, dan menelusuri request baru
- Client mengirim request untuk semua objek segera setelah menerima html- dasar
- Jumlah RTT yang diperlukan lebih sedikit dan berkurang akibat slow-start

HTTP 2.0

- *HTTP 2.0* adalah versi berikutnya dari HTTP dan didasarkan pada *Google SPDY,* yang dirancang untuk mempercepat *loading* halaman web dan proses browsing.
- *HTTP 2.0* adalah standar baru dan akan mengambil alih protokol *HTTP 1.1* yang saat ini digunakan oleh sebagian besar situs di internet.

PERBEDAAN HTTP 2.0 DENGAN 1.1

- ***HTTP 2.0*** adalah protokol yang lebih modern yang bisa meningkatkan kecepatan ***browsing web*** dengan menggunakan cara-cara baru transportasi data antara browser dan server di internet.
- ***HTTP 2.0*** kompatibel dengan ***HTTP 1.1*** dan menggunakan sebagian besar teknologi yang sama, tetapi lebih efisien dan memungkinkan server untuk merespon dengan lebih banyak konten daripada yang diminta, menghilangkan kebutuhan komputer pengguna terus mengirim permintaan informasi lebih lanjut sampai situs benar-benar ditampilkan

IMPLEMENTASI HTTP 2.0

- Standar ***HTTP 2.0*** secara resmi telah disetujui oleh *Internet Engineering Task Force* dan akan segera dipublikasikan.
- Google telah mengatakan bahwa protokol SPDY saat ini akan ditarik guna mendukung ***HTTP 2.0*** di Chrome pada awal tahun 2016.

URI DAN URL

- Sumber daya yang hendak diakses dengan menggunakan *HTTP* diidentifikasi dengan menggunakan *Uniform Resource Identifier (URI)*, atau lebih khusus melalui *Uniform Resource Locator (URL)*, menggunakan skema URI **http:** atau **https:**.

SESI HTTP

- Sebuah ***sesi HTTP*** adalah ***urutan transaksi permintaan*** dan ***respons jaringan dengan menggunakan protokol HTTP***.
- Sebuah **klien *HTTP*** akan memulai sebuah **permintaan**, **Klien** tersebut akan membuka sebuah koneksi ***Transmission Control Protocol (TCP)*** ke ***sebuah port tertentu*** yang terdapat dalam sebuah host (umumnya port 80 atau 8080).

SESI HTTP

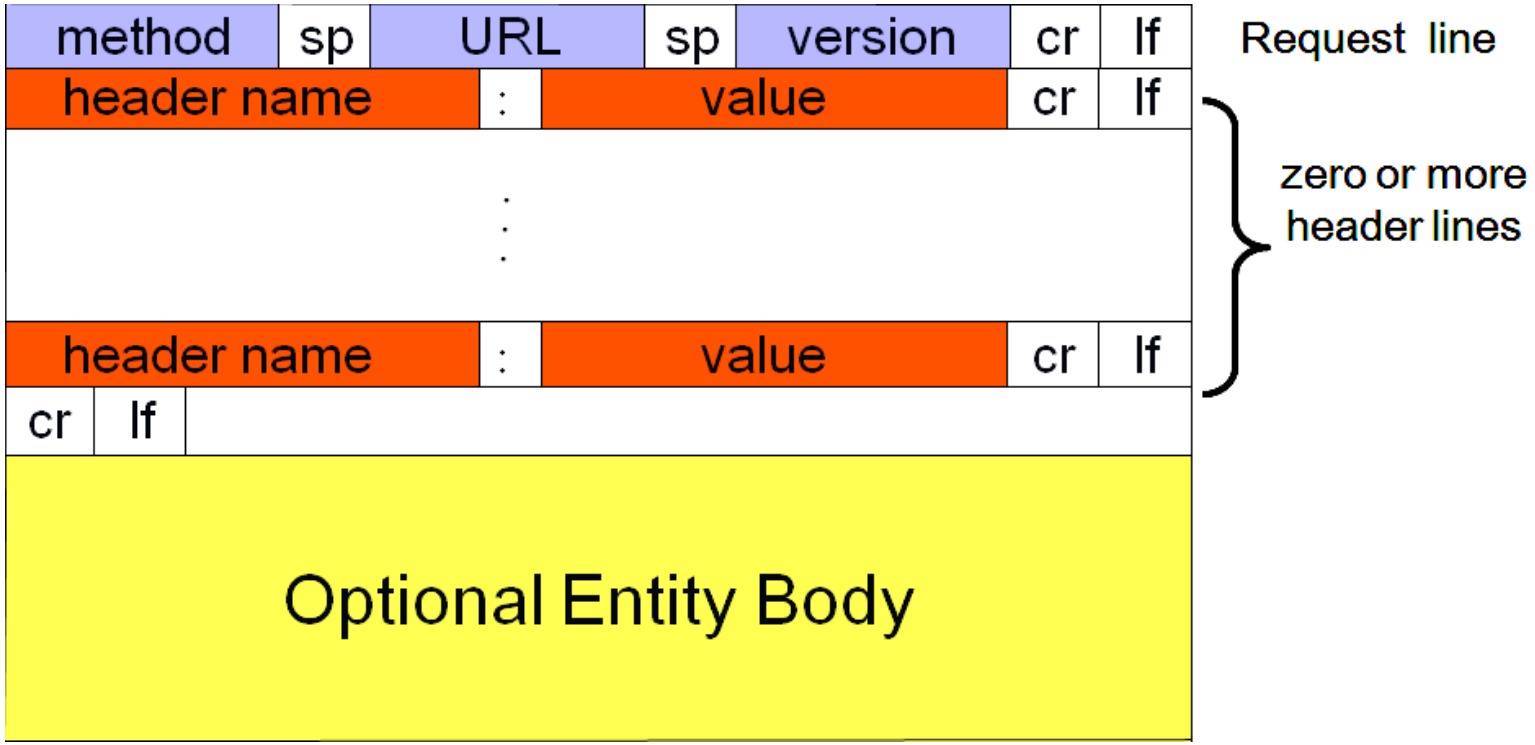
- ***Server*** yang mendengarkan pada ***port 80*** tersebut akan menunggu pesan permintaan **klien**.
- Saat menerima permintaan, **server** akan mengirimkan kembali baris status, seperti "**HTTP/1.1 200 OK**", dan pesan yang hendak diminta, pesan kesalahan atau informasi lainnya.

SESI HTTP

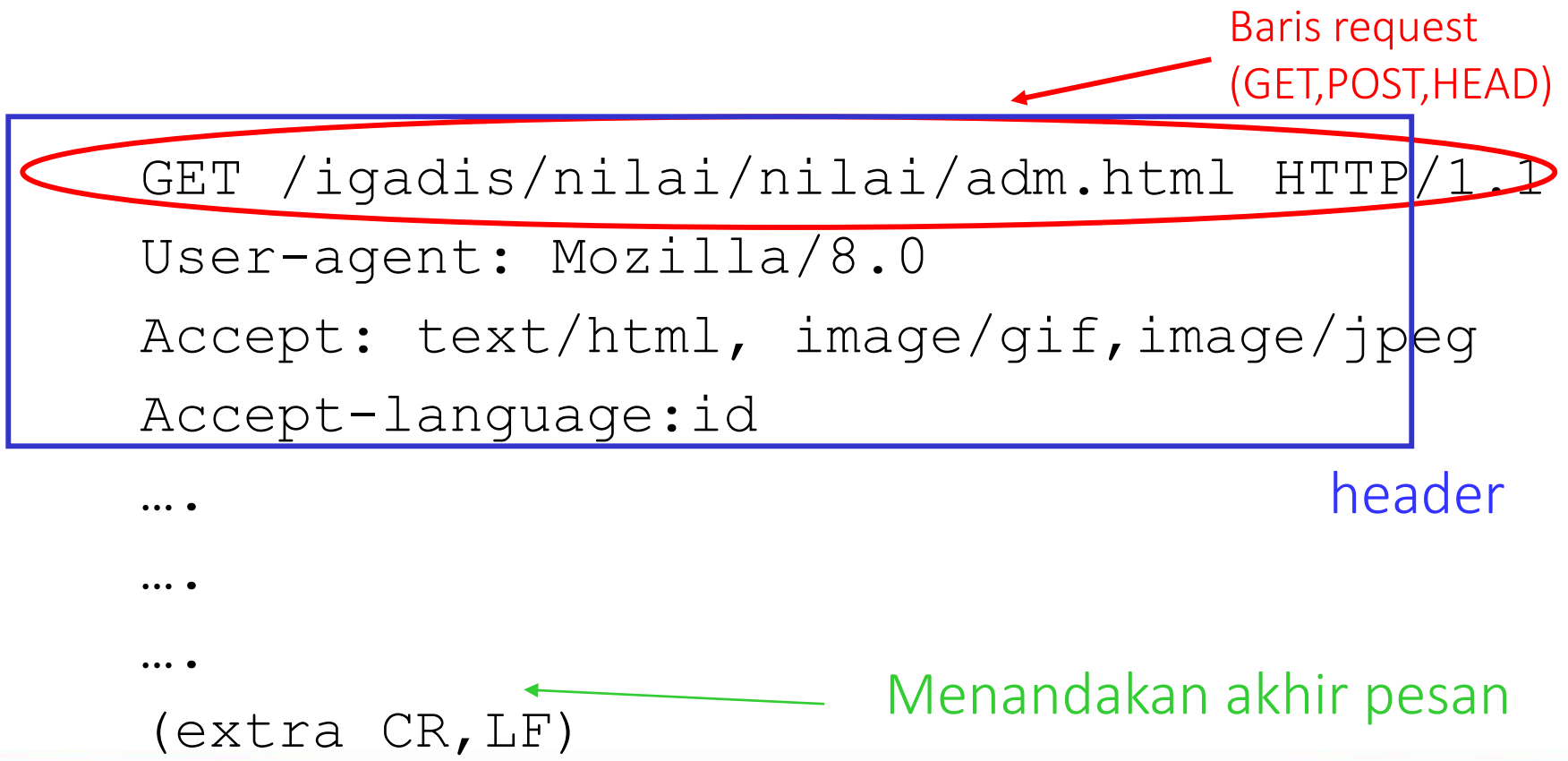
- contoh transaksi yang dilakukan oleh server dan klien. S = Server C = Client

```
C : (Inisialisasi koneksi)
C : GET /index.htm HTTP/1.1
C : Host: www.wikipedia.org
S : 200 OK
S : Mime-type: text/html
S :
S : -- data dokumen --
S : (close connection)
```

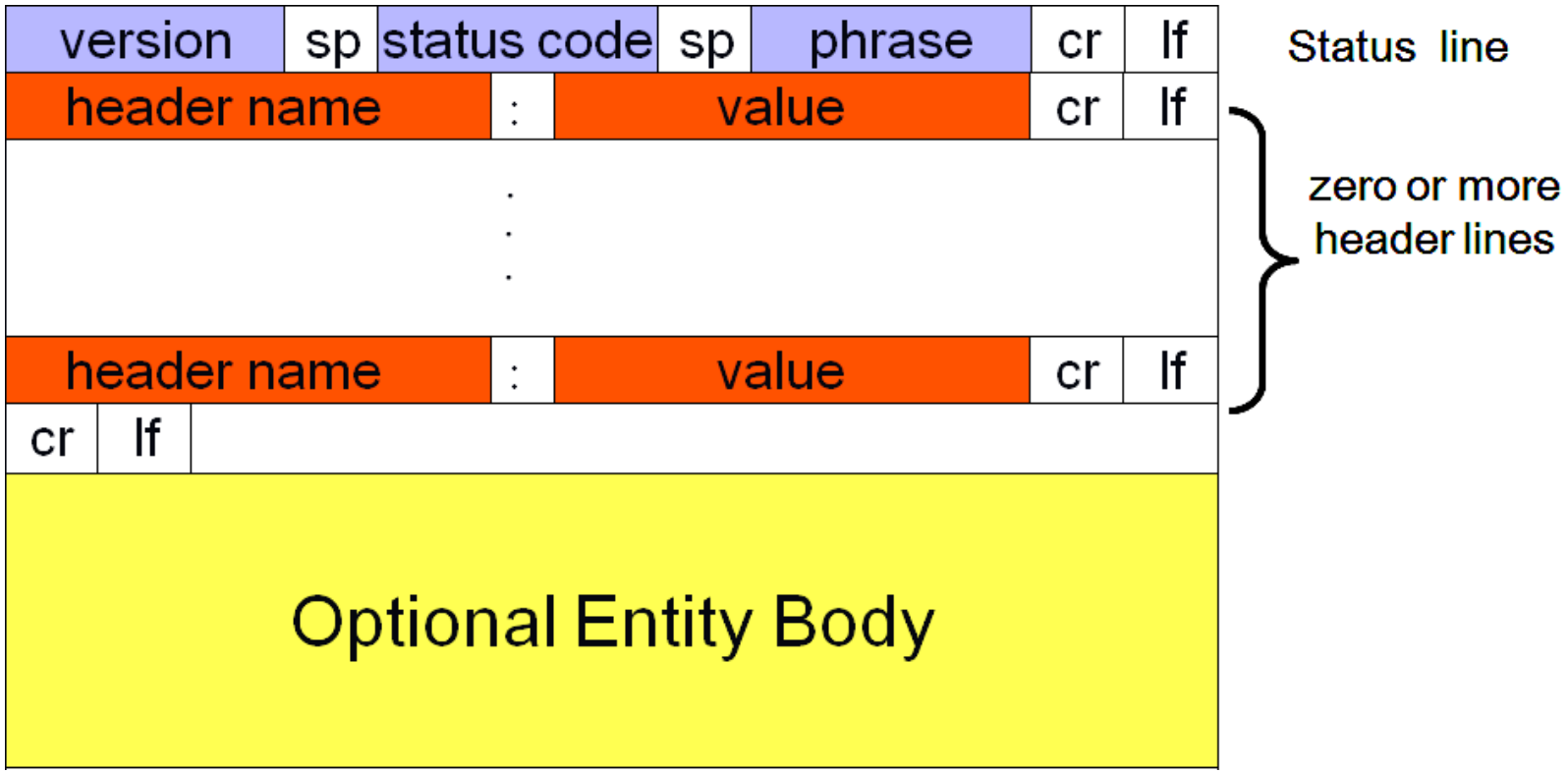
FORMAT UMUM : PESAN REQUEST HTTP



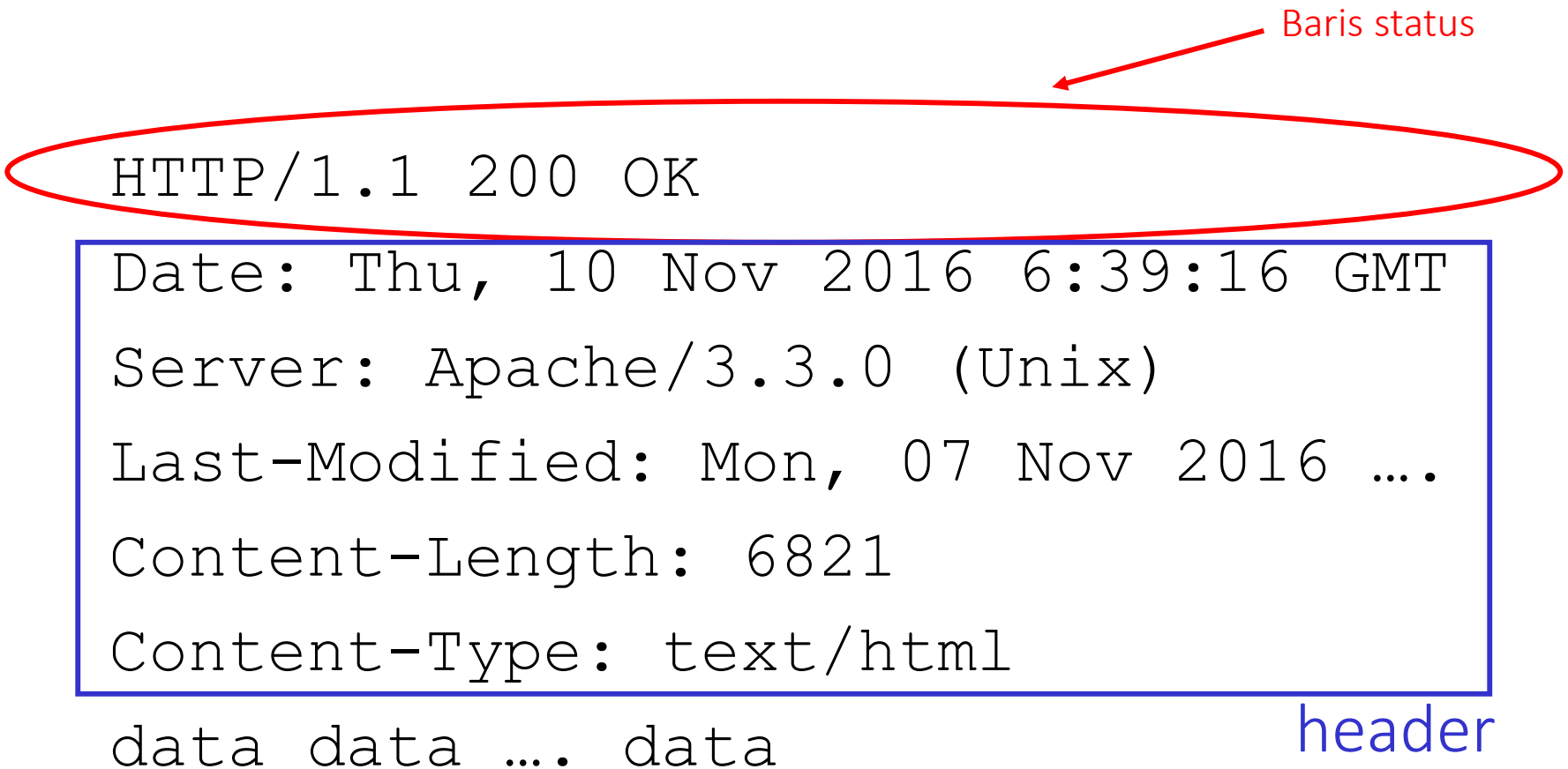
FORMAT PESAN HTTP: REQUEST



FORMAT UMUM : PESAN RESPONS HTTP



FORMAT PESAN HTTP: RESPONS



CODE STATUS RESPONSE HTTP

- **200 OK** → permintaan sukses, objek yang diminta sesudah pesan ini
- **301 Moved Permanently** → objek diminta sudah dipindah, lokasi baru sesudah pesan ini
- **400 Bad Request** → permintaan pesan tidak dimengerti server
- **404 Not Found** → dokumen yang diminta tidak ada diserver
- **505 HTTP Version Not Supported**

CODE STATUS RESPONSE HTTP

- **400** Bad Request The request contains bad syntax or cannot be fulfilled.
- **401** Unauthorized Similar to *403 Forbidden*, but specifically for use when authentication is possible but has failed or not yet been provided. See Basic access authentication and Digest access authentication.
- **402** Payment Required The original intention was that this code might be used as part of some form of digital cash or micropayment scheme, but that has not happened, and this code has never been used.
- **403** Forbidden The request was a legal request, but the server is refusing to respond to it. Unlike a *401 Unauthorized* response, authenticating will make no difference.
- **404** Not Found The requested resource could not be found.

CODE STATUS RESPONSE HTTP

- **405** Method Not Allowed A request was made of a resource using a request method not supported by that resource; for example, using GET on a form which requires data to be presented via POST, or using PUT on a read-only resource.
- **406** Not Acceptable
- **407** Proxy Authentication Required
- **408** Request Timeout Client failed to continue the request — except during playing Adobe Flash videos where it just means the user closed the video window or moved on to another video.

CODE STATUS RESPONSE HTTP

- 409 Conflict
- 410 Gone Indicates that the resource requested is no longer available and will not be available again. This should be used when a resource has been intentionally removed; however, in practice, a *404 Not Found* is often issued instead.
- 411 Length Required
- 412 Precondition Failed
- 413 Request Entity Too Large
- 414 Request-URI Too Long
- 415 Unsupported Media Type
- 416 Requested Range Not Satisfiable The client has asked for a portion of the file, but the server cannot supply that portion (for example, if the client asked for a part of the file that lies beyond the end of the file).

FILE TRANSFER PROTOCOL

- **FTP** (*File Transfer Protocol*) adalah sebuah protokol Internet yang berjalan di dalam lapisan aplikasi yang merupakan ***standar untuk pengiriman berkas (file) komputer*** antar mesin-mesin dalam sebuah jaringan.

FILE TRANSFER PROTOCOL

- *FTP* merupakan salah satu protokol Internet yang paling awal dikembangkan, dan masih digunakan hingga saat ini untuk melakukan **pengunduhan** (*download*) dan **penggugahan (upload)** berkas-berkas komputer antara **klien FTP** dan **server FTP**.

FTP SERVER & FTP CLIENT

- Sebuah **Klien FTP** merupakan aplikasi yang dapat mengeluarkan **perintah-perintah FTP** ke sebuah **server FTP**, sementara **server FTP** adalah ***sebuah Windows Service atau daemon*** yang berjalan di atas sebuah komputer yang merespons perintah-perintah dari sebuah *klien FTP*.

FTP SERVER & FTP CLIENT

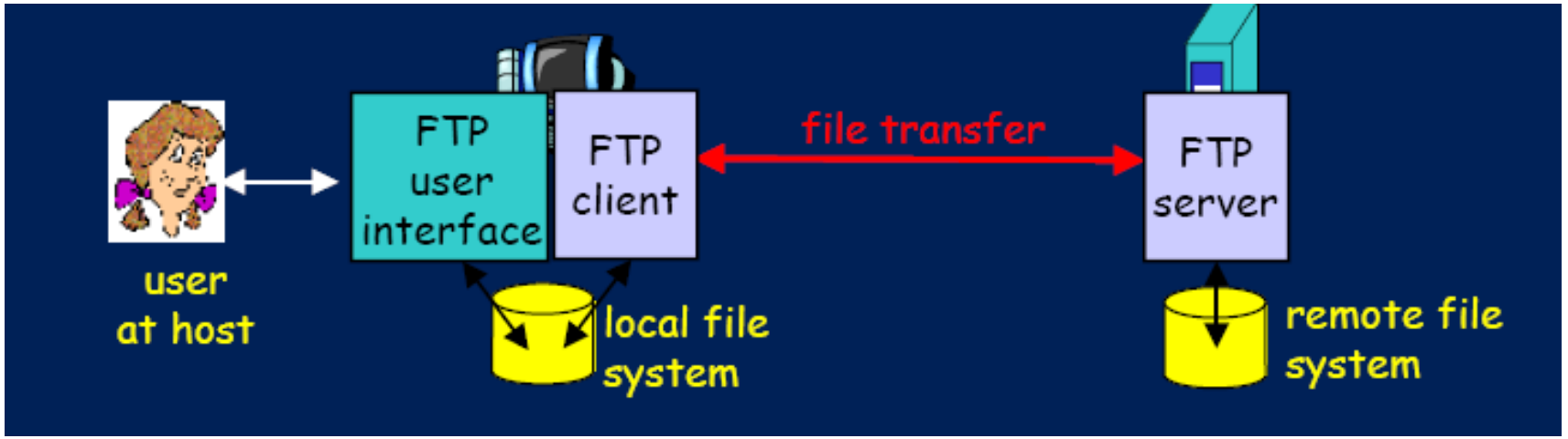
Perintah-perintah FTP dapat digunakan untuk :

- Mengubah direktori,
- Mengubah modus pengiriman antara biner dan ASCII,
- Menggugah berkas komputer ke server FTP,
- Mengunduh berkas dari server FTP.

AKSES FTP

- Sebuah server FTP diakses dengan menggunakan ***Universal Resource Identifier*** **(URI)** dengan menggunakan format ftp://namaserver.
- ***Klien FTP*** dapat menghubungi ***server FTP*** dengan membuka **URI** tersebut.

FTP : FILE TRANSFER PROTOCOL



- Transfer file dari/ke remote host
- Menggunakan model client-server
  - Client : sisi yang menginisiasi transfer
  - Server : remote host
- RPC 959
- ftp server port : 21

PORT TCP

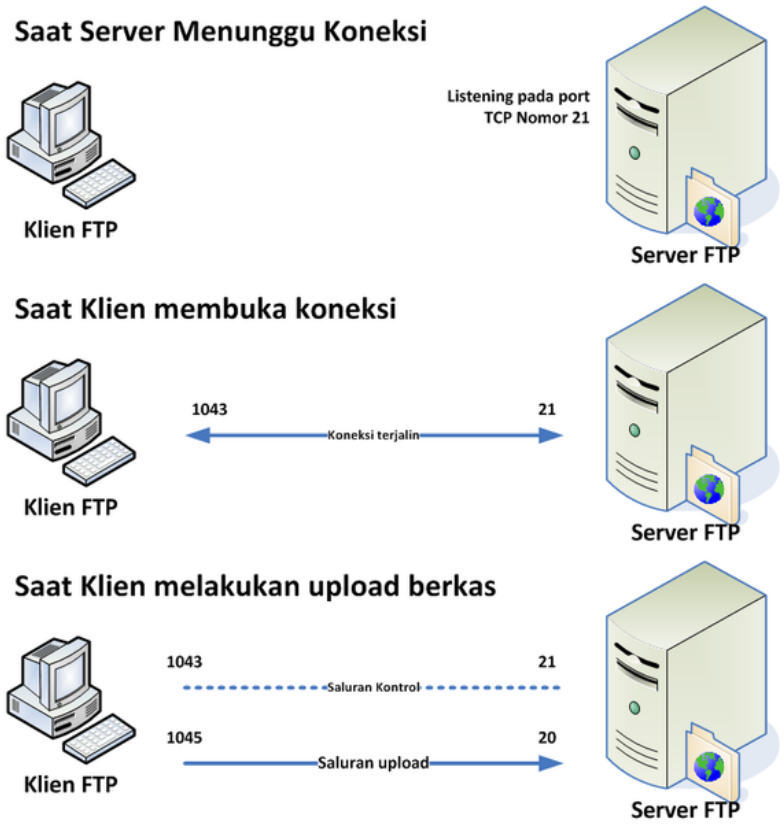
- *FTP* menggunakan protokol *Transmission Control Protocol* (TCP) untuk ***komunikasi data antara klien dan server***, sehingga di antara kedua komponen tersebut akan dibuatlah ***sebuah sesi komunikasi*** sebelum pengiriman data dimulai.

PORT TCP

- Sebelum membuat koneksi, **port TCP nomor 21** di sisi server akan ***"mendengarkan"*** percobaan koneksi dari sebuah **klien FTP** dan kemudian akan digunakan sebagai port pengatur (*control port*) untuk :

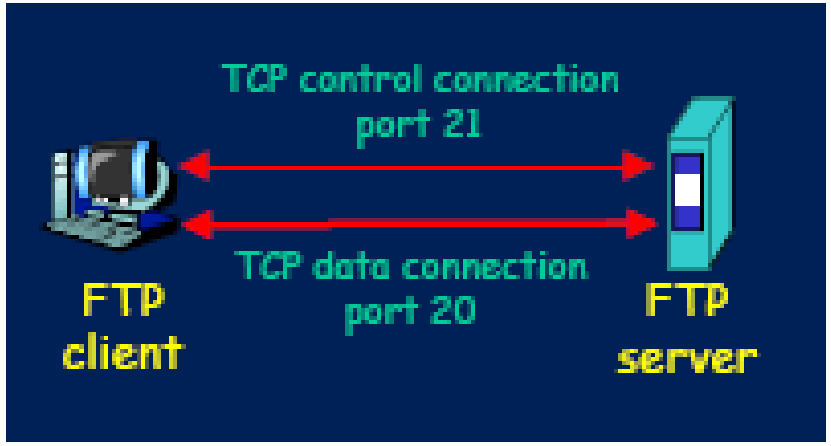
  1. Membuat sebuah koneksi antara klien dan server,
  2. Mengizinkan klien untuk mengirimkan sebuah perintah FTP kepada server
  3. Mengembalikan respons *server* ke perintah tersebut. Sekali koneksi kontrol telah dibuat, maka server akan mulai membuka port TCP nomor 20 untuk membentuk sebuah koneksi baru dengan klien untuk mengirim data aktual yang sedang dipertukarkan saat melakukan pengunduhan dan penggugahan.

KONEKSI FTP


Saat Server Menunggu Koneksi
Listening pada port
TCP Nomor 21
Klien FTP
Server FTP
Saat Klien membuka koneksi
1043
21
Koneksi terjalin
Klien FTP
Server FTP
Saat Klien melakukan upload berkas
1043
21
Saluran Kontrol
1045
20
Saluran upload
Klien FTP
Server FTP

FTP : KONEKSI DATA DAN KENDALI TERPISAH

- Klien ftp menghubungi server ftp pada port 21, menggunakan TCP sebagai protokol transport



- 2 koneksi TCP paralel dibuka:
  - Data : data file dari/ke server
  - Kendali : bertukar command dan response antar client dan server
  - Sever ftp mempertahankan ‘state’ : direktori sekarang dan autentifikasi sebelumnya

COMMAND DAN RESPONSE FTP

Contoh command:

- Dikirim dalam bentuk teks ASCII melalui kanal kendali
- **USER** *username*
- **PASS** *password*
- **LIST** {memperlihatkan daftar file di direktori saat ini}
- **RETR** *filename* {mengambil file}
- **STOR** *filename* {menyimpan file ke remote host}

Contoh response:

- Response berisi code status dan kalimat perberitahuan standar
- 331 Username OK, password required
- 125 data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file

FTP login

```
TCP:      .... ..0. SYN Flag = FALSE
TCP:      .... ...0 FIN Flag = FALSE
TCP: Window = 17264
TCP: Checksum = 518C
TCP: Urgent pointer = 00000000

------------  FTP Header  ------------
FTP: USER agv<0D><0A>
```

```
Record #52      (From Hub To Node) Captured on 05.14.07 at 10:48:33.984375000 I

   00 00 cd 25 fe e0 00 12    f0 68 40 52 08 00 45 00    ...%.... .h@R..E.
   00 32 03 34 40 00 80 06    48 50 0a 0e d0 1a 0a 0e    .2.4@... HP......
   cb 0b 04 e6 00 15 00 98    58 72 e2 63 fb 93 50 18    ........ Xr.c..P.
   43 70 51 8c 00 00 55 53    45 52 20 61 67 76 0d 0a    CpQ...US h@R..E..
   14 92 4a 11                                           ..J.
```

Ready FastEth

```
TCP:      .... ..0. SYN Flag = FALSE
TCP:      .... ...0 FIN Flag = FALSE
TCP: Window = 17232
TCP: Checksum = D88F
TCP: Urgent pointer = 00000000

------------  FTP Header  ------------
FTP: PASS ader<0D><0A>
```

```
Record #54      (From Hub To Node) Captured on 05.14.07 at 10:48:34.000072

   00 00 cd 25 fe e0 00 12    f0 68 40 52 08 00 45 00    ...%.... .h@R..E.
   00 33 03 38 40 00 80 06    48 4b 0a 0e d0 1a 0a 0e    .3.8@... HK......
   cb 0b 04 e6 00 15 00 98    58 7c e2 63 fb b3 50 18    ........ X|.c..P.
   43 50 d8 8f 00 00 50 41    53 53 20 61 67 75 73 0d    CP....PA .h@R..E.
   0a 28 c3 fc b4                                        .(...
```

Ready Fas