

IF3110
HTTP, DNS AND WEB SERVER
MINGGU KE-2



Institut Teknologi
Sumatera



OUTLINE

HTTP & HTTPS

HTTP STRUCTURE

HTTP HANDLER

DNS

WEB SERVER

PROTOKOL

“A protocol is a standard set of rules that allow electronic devices to communicate with each other. These rules include what type of data may be transmitted, what commands are used to send and receive data, and how data transfers are confirmed. You can think of a protocol as a spoken language”.

Contoh Protokol:

- SMTP
- FTP
- IMAP
- POP3

HTTP (*HYPertext TRansfer PROTOCOL*)



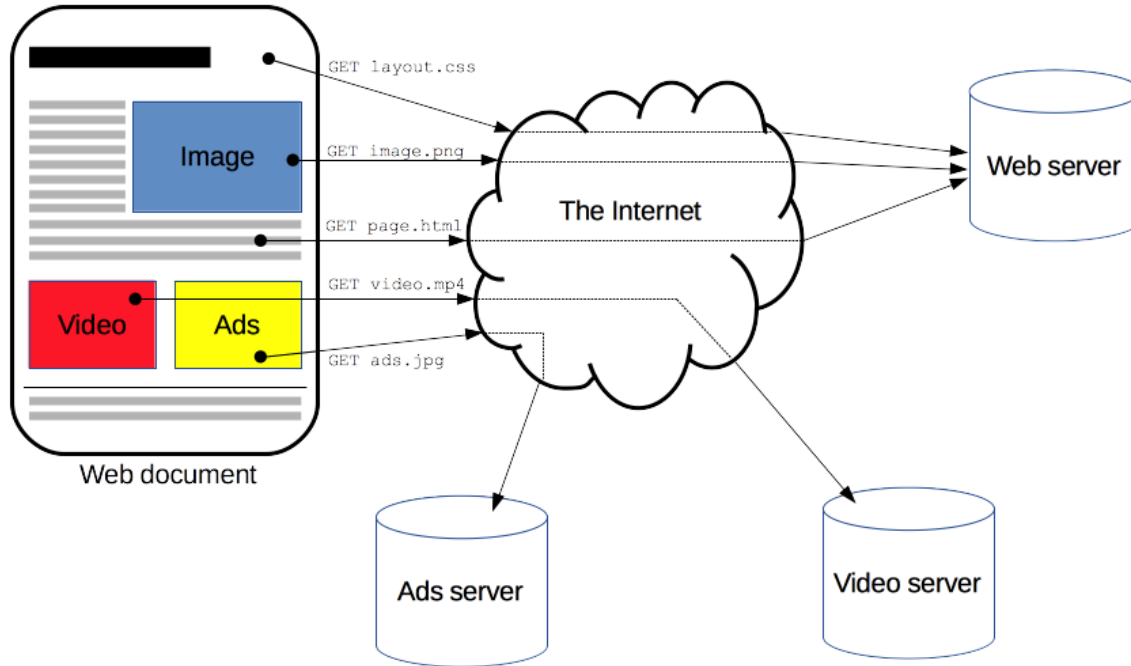
http://

HTTP merupakan standar protokol yang digunakan dalam web

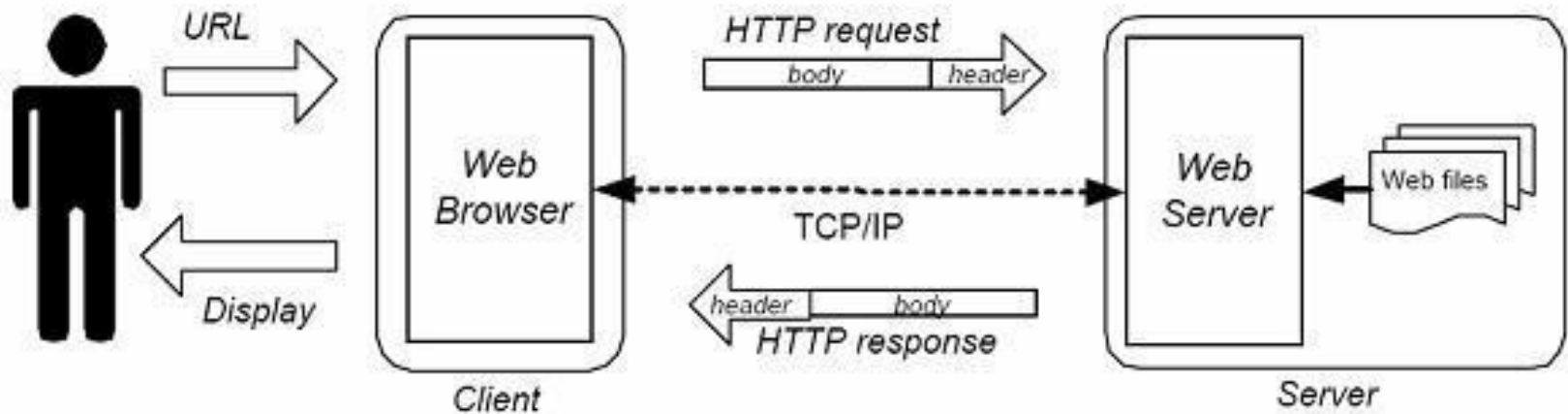
Secara Khusus, HTTP berfungsi menentukan bagaimana sebuah data dapat ditransmisikan maupun diformat menjadi bentuk yang dapat merespon browser untuk menampilkan data-data tersebut.

Internet Engineering Task Force bekerja sama dengan *World Wide Web Consortium* untuk pengembangan HTTP

PRINSIP KERJA HTTP

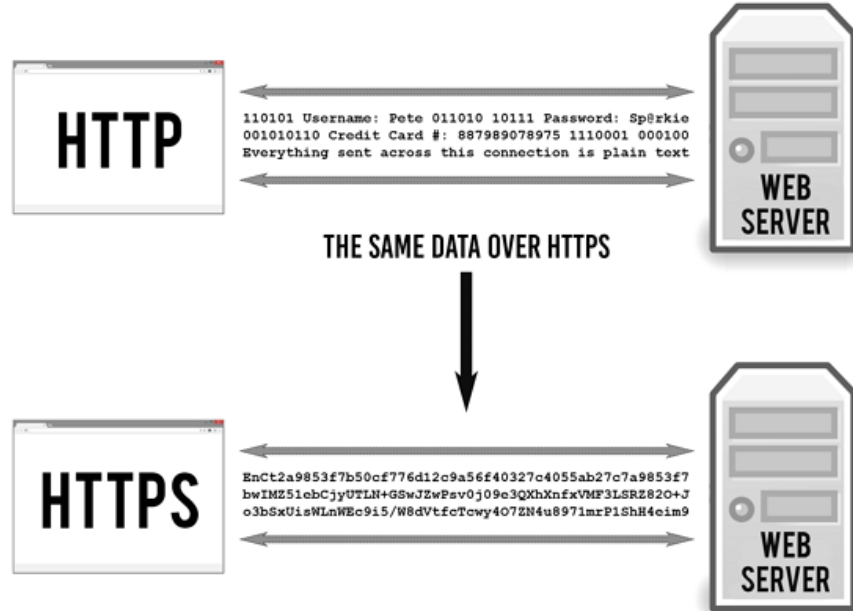


PRINSIP KERJA HTTP



HTTP VS HTTPS

HTTPS bekerja dengan menggunakan sublayer berupa SSL atau TLS (Transport Layer Security),



HTTPS

https://

Tingkat Keamanan Transmisi Data

Autentifikasi server

Enkripsi

integritas data



https://

Port yang Dipakai

- HTTP menggunakan port 80
- HTTPS menggunakan port 443

Butuh Secure Socket Layers (SSL)

HTTP menggunakan sistem client-server untuk dapat mentransmisikan data.

Untuk menggunakan HTTPS, situs harus memiliki sertifikat SSL, yaitu teknologi untuk mengenkripsi data yang ditransmisikan antara server dan klien.



PROTOKOL DAN STRUKTUR HTTP

Potokol HTTP menggunakan paradigma ***request & response*** (permintaan dan respon)

Struktur *request & response* terdiri dari:

- *Header* (informasi satu atau beberapa baris)
- Baris Kosong
- *Message Body* (isi utama)

Termasuk Stateless Protocol: transaksi yang hanya terdiri dari **sebuah *request*** dari client dan **sebuah *response*** dari server

PROTOKOL DAN STRUKTUR HTTP

Requests

POST / HTTP/1.1

Host: localhost:8000

User-Agent: Mozilla/5.0 (Macintosh;...)... Firefox/51.0

Accept: text/html,application/xhtml+xml,...,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Content-Type: multipart/form-data; boundary=-12656974

Content-Length: 345

-12656974

(more data)

Responses

HTTP/1.1 403 Forbidden

Server: Apache

Content-Type: text/html; charset=iso-8859-1

Date: Wed, 10 Aug 2016 09:23:25 GMT

Keep-Alive: timeout=5, max=1000

Connection: Keep-Alive

Age: 3464

Date: Wed, 10 Aug 2016 09:46:25 GMT

X-Cache-Info: caching

Content-Length: 220

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">

(more data)

start-
line

HTTP headers

empty
line

body

STRUKTUR HTTP REQUEST

```
METHOD /path-to-resource      HTTP/version-number  
Header-Name-1: value  
Header-Name-2: value  
  
[Optional Request Body]
```

FORMAT

```
GET /q?s=GOOG HTTP/1.1  
Host: finance.google.com  
  
User-Agent: Mozilla/24.75 [en] (WinNT; U)
```

CONTOH

```
HEAD HTTP://www.itera.ac.id/kuliah/ HTTP/1.1  
Host: www.itera.ac.id  
User-Agent: Mozilla/24.75 [en] (WinNT; U)
```

CONTOH

STRUKTUR HTTP REQUEST

Request Line/ Start line

- *Request Method*: GET, POST, HEAD
- URL yang diakses
- Versi HTTP: 1.0 atau 1.1

Pasangan variabel *header* dan nilainya

- *Host*: Target URL yang ditentukan
- *Content-length*: Panjang konten
- *User-Agent*: Tipe OS dan browser pengirim
- *Cookie*, dll

Request Body (*Optional*)

Tempat data untuk dikirimkan ke server

STRUKTUR HTTP RESPONSE

```
HTTP/version-number      status-code      message
Header-Name-1: value
Header-Name-2: value

[response body]
```

FORMAT

```
HTTP/1.0 200 OK
Date: Sat, 03 Feb 2001 22:48:35 GMT
Connection: close
Content-type = text/HTML
Set-Cookie: B=9ql5kgct7p2m3&b=2;expires=Thu,15 Apr 2010 20:00:00
GMT; path=/; domain=.yahoo.com
<HTML>
<HEAD><TITLE>Yahoo! Finance - YHOO</TITLE></HEAD>
<BODY> _ </BODY>
</HTML>
```

CONTOH

STRUKTUR HTTP RESPONSE

Status Line

- Versi HTTP: 1.0 atau 1.1
- *Status Code* dan deskripsinya

Pasangan variabel *header* dan nilainya

- *Content-type*
- *Content-length*
- *Set-Cookie*
- *Date*, dll

Response Body (*Optional*)

Step 1: Initial user request for "http://www.cs.rutgers.edu/~shklar/"

- Once receive the document; browser parses the doc to define additional resources to be retrieved



```
GET /~shklar/ HTTP/1.1
Host: www.cs.rutgers.edu
```

Browser

Request

Server

Response

```
HTTP/1.1 200 OK
Content-Type: text/html

<html>
<head> ... </head>
<body>
...

...
</body>
</html>
```

Step 2: Secondary browser request for "http://www.cs.rutgers.edu/~shklar/images/photo.gif"



```
GET /~shklar/images/photo.gif HTTP/1.1
Host: www.cs.rutgers.edu
```

Browser

Request

Server

Response

```
HTTP/1.1 200 OK
Content-Type: image/gif
```



Figure

Sequence of browser requests for loading a sample page

REQUEST METHOD :

GET VS POST

GET

- can be cached
- remain in the browser history
- can be bookmarked
- Data is visible to everyone in the URL
- have length restrictions
- should be used only to retrieve data
- Only ASCII characters allowed

POST

- never be cached
- do not remain in the browser history
- cannot be bookmarked
- Data is not displayed in the URL
- have no restrictions on data length
- No restrictions. Binary data is also allowed

REQUEST METHOD: HEAD

Head

- Mirip Get
- Server hanya mengembalikan *header*
- Digunakan untuk mendukung **cache** berdasarkan *content modification* information (*last modified*)

STATUS CODE

Memberitahu *browser* atau *proxy* apakah *response* sesuai dengan yang diharapkan

- 1xx : Informasi
- 2xx : Sukses
- 3xx : Redirection
- 4xx : Client Request Error
- 5xx : Server Error

HTTP *HEADER*

General Header

- Date: Sun, 11 Feb 2001 22:38:31 GMT
 - Jam dan tanggal *message* dibuat
- Connection: Close
 - *Client* dan *Server* ingin koneksi tetap diperahankan atau tidak

Request Header

- User-Agent: Mozilla/4.75 [en] (WinNT; U)
 - Identitas *Browser*
 - Host: www.neurozen.com
 - untuk mendukung *virtual host*
 - Referer: http://www.cs.rutgers.edu/index.html
 - URL dari halaman sebelumnya (yang *me-refer*)
-

HTTP *HEADER*

Response Header

- Location(URL): `http://www.mywebsite.com/Page.html`
 - Halaman yang harus dituju (*redirect*)
- Server: `Apache/1.2.5`
 - Identitas *web server* yang digunakan

Entity Header

- Content-Type: `mime-type/mime-subtype`
 - Tipe dari *message body*
 - Content-Length: `xxx`
 - Panjang *message body*
 - Last-Modified: `Sun, 11 Feb 2001 22:28:31 GMT`
 - Tanggal modifikasi *content*
-

VIRTUAL HOSTING

Melakukan *hosting* terhadap beberapa nama domain pada server

Umumnya apabila seseorang membuat situs, akan dihosting di ***shared web hosting***, terkecuali perusahaan besar yang memiliki ***dedicated server*** sendiri

```
GET http://finance.google.com/q?=GOOG HTTP/1.1  
Host: finance.google.com
```

```
GET /q?s=GOOG HTTP/1.1  
Host: finance.google.com
```

AUTHENTICATION

Otentikasi adalah proses menentukan apakah seseorang yang mengakses adalah diri seseorang yang dia klaim.

*karena bisa saja seseorang menggunakan identitas orang lain

```
HTTP/1.1 401 Authenticate  
Date: Mon, 05 Feb 2001 03:41:23 GMT  
Server: Apache/1.2.5  
WWW-Authenticate: Basic realm="Chapter309"
```

```
GET /book/chapter3/index.html HTTP/1.1  
Date: Mon, 05 Feb 2001 03:41:24 GMT  
Host: www.neurozen.com  
Authorization: Basic eNCoDed-uSErId: password
```

SESSION MANAGEMENT

Pengaturan Sesi adalah pengaturan terhadap informasi semi-permanen antara dua pihak (biasanya *client-server*)

```
GET /kelas/pengembangan-web/HTTP/1.1
Host: www.itera.ac.id
Authorization: ...
```

```
HTTP/1.1 200 OK
Set-Cookie: Client=Rich; path=/kelas
...
```

```
GET /kelas/pengembangan-web HTTP/1.1
Host: www.itera.ac.id
Cookie: Client=Rich
```


CACHING CONTROL

Caching control mengatur penyimpanan sementara pada suatu tempat dari berkas-berkas seperti halaman HTML, gambar, dan dokumen/obyek web lainnya dengan tujuan mengurangi penggunaan *bandwith*, *lag* dan beban pada server

```
GET /~shklar/ HTTP/1.1  
Host: www.cs.rutgers.edu  
If-Modified-Since: Fri, 11 Feb 2001 22:28:00 GMT
```

PERSISTENT CONNECTION

Persistent Connection atau biasa disebut HTTP keep-alive bertujuan menjaga koneksi agar tetap hidup sehingga dapat mengirimkan beberapa (*multiple*) HTTP-request atau HTTP-response secara konkuren

HTTP 1.0 menggunakan TCP terpisah untuk setiap request

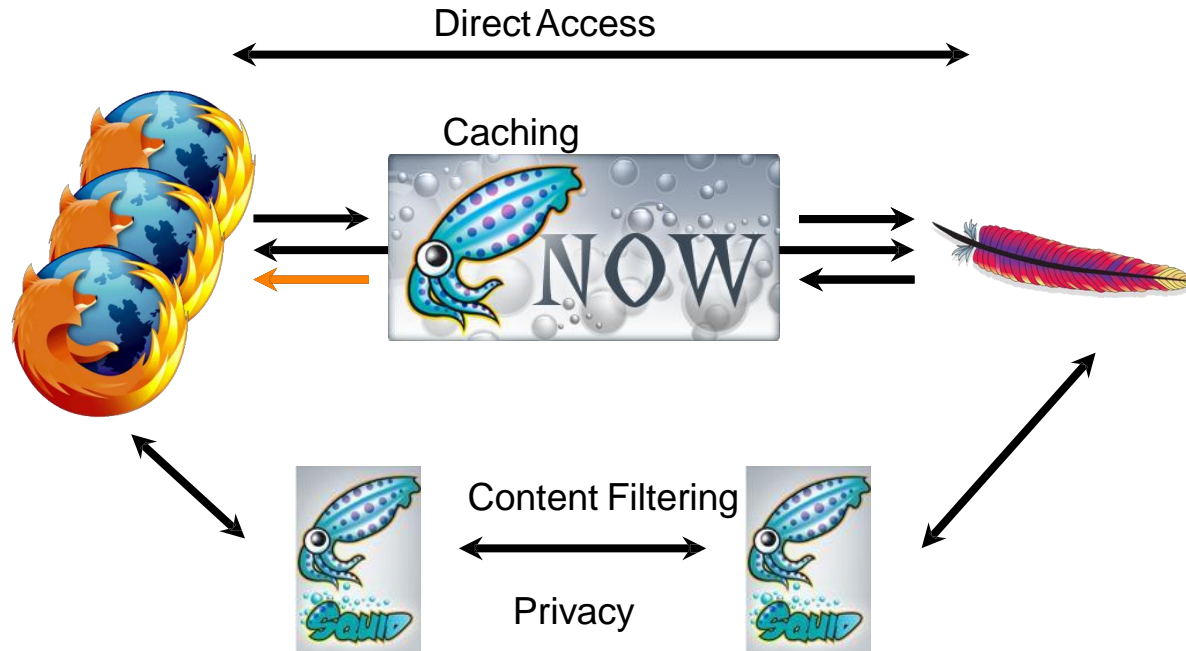
- tidak efisien
- lambat

HTTP 1.1 menggunakan persistent connection, yang bisa digunakan untuk banyak request

HTTP 1.1 VS HTTP 1.0

- Additional Methods (PUT, DELETE, TRACE, CONNECT + GET, HEAD, POST)
- Additional Headers
- Transfer Coding (chunk encoding)
- Persistent Connections (content-length matters)
- Request Pipelining

WHY USE A PROXY?



A SIMPLE WEB PROXY

- Forward client requests to the remote server and return response to the client
- Handle HTTP 1.0 (GET)
- Multi-process, non-caching web proxy
- `./proxy <port>`

Reference: RFC 1945 (HTTP 1.0) / RFC 2616 (HTTP 1.1)

HANDLING REQUESTS

What you need from a client request: host, port, and URI path

```
GET http://www.princeton.edu:80/ HTTP/1.0
```

What you send to a remote server:

```
GET / HTTP/1.0
```

```
Host: www.princeton.edu:80
```

```
Connection: close
```

Check request line and header format

HANDLING RESPONSES



DNS (DOMAIN NAME SYSTEM)

DNS adalah pendistribusian sistem database yang digunakan untuk mencari "nama" komputer pada jaringan internet yang memakai TCP/IP (penamaan komputer) dengan mengarahkan nama domain kepada alamat IP address.

Berlaku sebaliknya juga dari alamat IP address ke domain.



FUNGSI DNS

- Menerjemahkan hostname menjadi alamat IP address atau menerjemahkan alamat IP address menjadi hostname.
- Untuk mengetahui informasi secara lengkap mengenai sebuah hostname yang ada di internet baik itu alamat IP address, alamat lokasi server atau hostname, dan waktu yang digunakan.

DNS SERVER

DNS Server adalah sebuah sistem server yang menerima permintaan dari client untuk mengetahui alamat IP address yang dipakai oleh sebuah domain.

Misal domain google.com, dengan menggunakan tool online ipadress.com berhasil mengetahui IP address google yakni 172.217.21.206.

IP Address of Google is 172.217.21.206			
Hostname:	google.com	City:	Mountain View
IP Address:	172.217.21.206	Country:	United States 🇺🇸
Host of this IP:	fra16s12-in-f14.1e100.net	State:	California
Organization:	Google	Postal Code:	94043
ISP/Hosting:	Google	Timezone:	America/Los_Angeles
Updated:	10/27/2016 03:26 AM	Local Time:	10/30/2016 06:36 PM

WEB SERVER

Software yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS

Mengirimkan kembali hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML.



Web Server

FUNGSI WEB SERVER

Fungsi Web server adalah

- Untuk mentransfer berkas yang diminta pengguna melalui protokol komunikasi yang telah ditentukan
- Berkas dikirim berupa halaman web yang didalamnya bisa terdapat teks, video, gambar, file dan lainnya.

CONTOH WEB SERVER

- Apache
- Nginx
- Microsoft windows Server 2003 Internet Information Services (IIS)
- Lighttpd
- Sun Java System Web Server
- Xitami Web Server
- Zeus Web Server