

11. Web (HTTP) server

WWW(W3)

World Wide Web (disingkat sebagai WWW atau W3, umumnya dikenal sebagai web) adalah sistem dokumen hypertext yang diakses melalui Internet. Dengan web browser, kita dapat melihat halaman web yang mungkin berisi teks, gambar, video, dan multimedia lainnya dan menavigasi antara mereka melalui hyperlink

World Wide Web dimulai sebagai proyek CERN bernama Enquire, diprakarsai oleh Tim Berners-Lee pada tahun 1989 dan Robert Cailliau pada tahun 1990. Berners-Lee dan Cailliau diberi penghargaan oleh Association for Computing Machinery pada tahun 1995, Atas kontribusi mereka kepada pengembangan World Wide Web. Berdasarkan konsep hypertext, proyek ini bertujuan untuk memfasilitasi berbagi informasi di antara para peneliti. Situs web pertama di on-line kan pada tahun 1991. Pada tanggal 30 April 1993, CERN mengumumkan bahwa World Wide Web akan bebas untuk siapa pun. Salinan asli halaman pertama web, yang diciptakan oleh Berners-Lee, masih dipublikasikan di situs World Wide Web Consortium sebagai dokumen sejarah. Sebelum pengembangan Web, CERN telah menjadi pelopor dalam pengenalan teknologi internet, dimulai pada awal tahun 1980. Sejarah singkat periode ini dapat ditemukan di CERN.ch.

Web site

Web site atau web (situs web) adalah kumpulan file-file world wide web (www) yang berisi file permulaan yang disebut halaman rumah (home page). Sebuah institusi atau perusahaan memberitahu Anda bagaimana untuk sampai ke situs web mereka dengan memberikan alamat halaman rumah mereka (URL). Dari halaman rumah, Anda bisa mendapatkan halaman lain di situs web mereka.

Contoh:

Situs web untuk STT Terpadu Nurul Fikri memiliki alamat halaman rumah <http://www.nurulfikri.ac.id>. Alamat halaman rumah STT Terpadu Nurul Fikri akan menghantarkan pengunjung web STT NF ke puluhan atau mungkin ratusan halaman lainnya

Cara kerja web

Web bekerja dalam konsep client – server. Web client meminta halaman (dokumen web) kepada web server melalui jaringan komputer. Web server merespon dengan mencarikan dokumen (halaman) pada lokasi penyimpanan dokumen-dokumen web yang ada pada web server. Selanjutnya

web server mengirimkannya kepada web client.

Contoh aplikasi web client: Mozilla Firefox, IE, Opera, Safari, Chrome dan lain lain.

Contoh aplikasi web server: Apache HTTP server, IIS, Nginx, lighttpd, tomcat dan lain lain.

Sebenarnya yang terjadi antara web client dan web server adalah proses transfer file file dokumen web. Agar mudah dipahami atau dimengerti oleh web client maka dokumen web harus memiliki format yang standar. Format dokumen web secara de facto adalah menggunakan format bahasa HTML (HyperText Mark-up Language). File format HTML ini biasanya berekstensi .html atau .htm

Contoh format HTML:

```
<HTML>
<HEAD>
<TITLE>Hello world</TITLE>
</HEAD>
<BODY>
<DIV> Hello World </DIV>
<DIV><H3>Selamat datang di web saya</H3></DIV>
</BODY>
</HTML>
```

Bahasa pemrograman web

Dalam perkembangannya, web tidak lagi hanya sekedar menampilkan informasi yang bersifat statis, melainkan juga dinamis dengan unsur unsur logika melibatkan basis data (web applications)

Untuk itu dibutuhkan mekanisme untuk menghasilkan file berformat HTML saat proses transfer file (on the fly) dari web server ke web client. Mekanisme itu terjadi di sisi web server yang ditangani oleh sebuah interpreter bahasa pemrograman.

Bahasa pemrograman yang menginterpretasi kode program menjadi file atau stream data berformat HTML yang ditransfer ke web client oleh web server disebut bahasa pemrograman web (server side language)

Contoh server side language:

- PHP, ekstensi file .php
- ColdFusion, ekstensi file .cfm
- ASP atau ASP.NET, ekstensi file .asp atau .aspx
- CGI, ekstensi file .cgi, .pl, .py

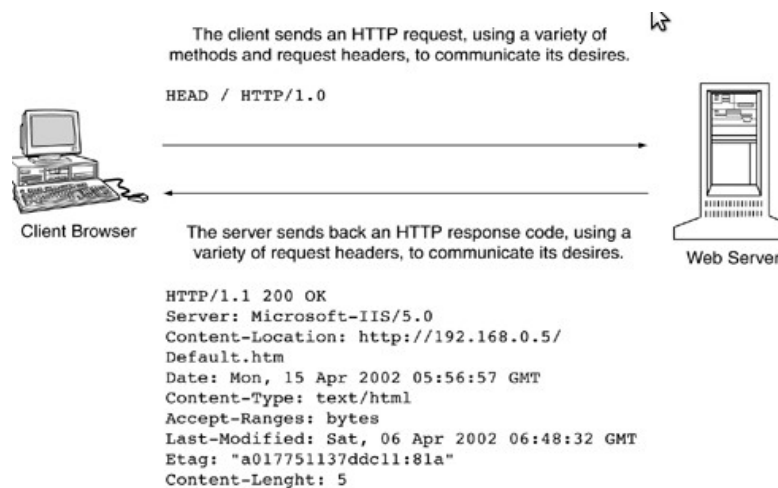
Protokol web

Untuk berkomunikasi dalam proses transfer dokumen web, antara web client dan web server, dibutuhkan serangkaian prosedur dan aturan dalam berkomunikasi yang disepakati dan dimengerti oleh keduanya.. Prosedur dan aturan berkomunikasi tersebut dinamakan protokol. Maka prosedur

dan aturan berkomunikasi antara web client dan web server kita sebut dengan istilah Protokol Web

Protokol web terdiri dari dua protokol yaitu:

- HTTP (HyperText Transfer Protocol), ini adalah protokol default komunikasi web, protokol ini secara lahiriah membawa kerentanan yang dikarenakan proses transfer data berupa clear text data. Nomor port service 80.
- HTTPS (HyperText Transfer Protocol Secure), ini disebut juga dengan istilah HTTP over SSL. Protokol ini relatif lebih aman dikarenakan telah menerapkan proses enkripsi data yang akan ditransfer. Nomor port service 443.



Gambar 1: Ilustrasi protokol web

Lab 11.1. Instalasi Apache HTTP Server

- Periksalah terlebih dahulu pada sistem komputer linux Anda apakah telah terinstall aplikasi apache HTTP server , dengan menggunakan perintah berikut ini:
`# sudo dpkg-query -f apache2`

atau jika sistem Anda menggunakan linux distro lain seperti RedHat dan CentOS, gunakan perintah berikut:

```
# rpm -q httpd
```

- Jika belum terinstal, lakukan instalasi apache HTTP server dengan perintah sebagai berikut:
`# sudo apt-get install apache2`
atau jika sistem Anda menggunakan linux distro lain seperti RedHat dan CentOS, gunakan perintah berikut:
`# yum install httpd`
- Lakukan kembali pemeriksaan apakah saat ini apache HTTP server telah terpasang pada

sistem komputer Anda seperti langkah sebelumnya.

Lab 11.2. Start/Stop Apache HTTP Server

- Lakukan pemeriksaan apakah apache HTTP server telah berjalan / aktif atau belum , dengan perintah berikut ini:
`sudo /etc/init.d/apache2 status`
atau jika sistem Anda menggunakan linux distro lain seperti RedHat dan CentOS, gunakan perintah berikut:
`/etc/init.d/httpd status`
- Jika ternyata service HTTP belum berjalan, maka lakukan perintah berikut ini untuk menjalankannya:
`sudo /etc/init.d/apache2 start`
atau jika sistem Anda menggunakan linux distro lain seperti RedHat dan CentOS, gunakan perintah berikut:
`/etc/init.d/httpd start`
- Periksa apakah apache HTTP server telah benar benar berjalan, dengan menggunakan web browser pada desktop Anda, dan ketikkan pada web browser alamat URL `http://localhost`
- Untuk mematikan service HTTP server dapat dilakukan dengan perintah berikut ini:
`sudo /etc/init.d/apache2 stop`
atau jika sistem Anda menggunakan linux distro lain seperti RedHat dan CentOS, gunakan perintah berikut:
`/etc/init.d/httpd stop`

Lab 11.3. Menemukan lokasi penyimpanan dokumen web

- Agar Anda dapat menempatkan file file atau dokumen web maka Anda harus mengetahui lokasi penyimpanan dokumen web pada server
- Untuk dapat mengetahui atau mendapatkan lokasi penyimpanan dokumen dokumen web, Anda dapat menggunakan perintah berikut ini:
`sudo grep DocumentRoot /etc/apache2/sites-enabled/000-default`
- Jika Anda mendapatkan output sebagai berikut :
`DocumentRoot /var/www`

Maka lokasi penyimpanan dokumen web adalah pada direktori `/var/www`

- Kemudian buktikan bahwa apache HTTP server akan mencari/mengambil dokumen web yang diminta oleh web client dengan menemukannya pada direktori `/var/www` , yaitu dengan cara Anda membuat file dengan nama `hello.html` pada direktori `/var/www` .
Buatlah file `hello.html` tersebut dengan isi file seperti berikut ini:
<HTML>
<HEAD><TITLE>Hello</TITLE></HEAD>
<BODY>

```
<H2>Hello STTT NF</H2>
</BODY>
</HTML>
```

- Selanjutnya coba Anda akses file tersebut melalui web browser dengan mengetikkan alamat URL <http://localhost/hello.html> . Apakah benar bahwa file hello.html ditemukan oleh apache HTTP server dan ditampilkan oleh web browser Anda ?
- Anda dapat juga membuktikannya dari sisi apache HTTP server, dengan melihat log. Untuk melihat log apache HTTP server gunakan perintah berikut ini:
tail /var/log/apache/access.log
- Perhatikan baris dengan keterangan mengandung pesan “GET /hello.html HTTP/1.1” , Jika

Lab 11.4. Memasang modul PHP pada Apache HTTP Server

- Periksa apakah apache HTTP server yang telah terpasang pada komputer Anda telah memiliki dukungan modul php atau belum, yaitu dengan membuat file info.php, yang diletakkan pada direktori /var/www .
- Isi dari file info.php adalah sebagai berikut:
<?php phpinfo();?>
- Kemudian coba Anda akses file tersebut dengan menggunakan web browser, dan mengetikkan alamat URL <http://localhost/info.php> .
- Jika tidak tampak halaman informasi tentang modul PHP hanya halaman blank, maka modul PHP belum terpasang.
- Untuk memasang modul PHP pada apache HTTP server, Anda harus menginstal perangkat lunak modul php dengan perintah sebagai berikut:
sudo apt-get install libapache2-mod-php5
atau jika sistem Anda menggunakan linux distro lain seperti RedHat dan CentOS, gunakan perintah berikut:
yum install php
- Setelah modul php terpasang, pastikan bahwa modul php telah diaktifkan dengan melihat file konfigurasi /etc/apache2/mods-enabled/php5.load , pada file tersebut akan tercantum konfigurasi yang akan memuat modul php pada apache HTTP server, buktikan dengan perintah berikut ini:
grep php5_module /etc/apache2/mods-enabled/php5.load
- Jika belum tampak file /etc/apache2/mods-enabled/php5.load, coba jalankan perintah berikut ini:
a2enmod php5
/etc/init.d/apache2 restart
- Selanjutnya panggil atau akses kembali file info.php dengan menggunakan web browser, dan mengetikkan alamat URL <http://localhost/info.php> , dan perhatikan kini apa yang tampil pada web browser..!

Lab 11.5. Memasang modul SSL pada Apache HTTP Server

- Modul ssl diperlukan untuk memungkinkan penggunaan channel HTTP yang aman dengan metode enkripsi
- Cobalah periksa apakah saat ini modul ssl telah terpasang, dengan mencoba mengetikkan URL <https://localhost> pada web browser. Jika pesan yang tampil adalah “Unable to connect” maka itu menandakan apache HTTP server belum berjalan dalam model secure alias belum terpasang dukungan SSL (https).
- Atau Anda bisa juga memeriksanya dengan menjalankan perintah berikut ini pada server:
`# sudo netstat -tanp | grep ":443"`
- Jika tidak ada output sama sekali menandakan tidak ada service https yang berjalan pada server Anda. Yang bisa disimpulkan juga bahwa modul SSL belum terpasang atau belum diaktifkan.
- Atau Anda juga bisa memastikan dengan melihat apakah terdapat file konfigurasi modul ssl pada direktori /etc/apache2/mods-enabled/ .
- Untuk memasang modul ssl , terlebih dahulu Anda harus menginstal perangkat lunak modul SSL dengan perintah berikut:
`# sudo apt-get install libapache2-mod-gnutls`
- Selanjutnya aktifkan modul ssl dan mengaktifkan konfigurasi ssl dengan perintah:
`# a2enmod ssl`
`# a2ensite default-ssl`
`# /etc/init.d/apache2 restart`
- Dan akhirnya coba sekali lagi mengakses web server Anda dengan mengetikkan URL <https://localhost> pada web browser
- Atau Anda bisa juga memeriksanya dengan menjalankan perintah berikut ini pada server:
`# sudo netstat -tanp | grep ":443"`
- Apakah apache HTTP server mode HTTPS sudah berjalan ?

Lab 11.6. Mengubah port apache HTTP server

- Jika suatu saat Anda ingin mengubah port service HTTP menjadi port tertentu yang bukan lagi port 80, maka Anda dapat melakukannya dengan mengubah konfigurasi apache HTTP server
- Untuk mengubah port service apache HTTP server, ubah direktif “Listen” yang terdapat dalam file konfigurasi /etc/apache2/ports.conf . Misal: ubah port menjadi 81 , sehingga menjadi seperti berikut:
Listen 81
- Juga ubah file /etc/apache2/sites-enabled/000-default.conf , ubah port 80 menjadi 81 pada bagian <VirtualHost *:80>.
- Kemudian restart service apache HTTP server.
- Selanjutnya coba akses web server Anda dengan mengetikkan URL <http://localhost> pada web browser , dan perhatikan apa yang terjadi ?

- Coba sekali lagi akses web server Anda dengan mengetikkan URL <http://localhost> pada web browser, dan perhatikan apa yang terjadi ?

Lab 11.7. Mengubah DocumentRoot

- Anda juga mungkin saja ingin mengubah lokasi penyimpanan dokumen web (DocumentRoot)
- Misal Anda ingin mengubah lokasi penyimpanan dokumen web nantinya pada direktori /opt/site .
- Untuk memungkinkan hal tersebut, maka langkah yang harus dilakukan pertama adalah, membuat direktori /opt/site.

```
# mkdir /opt/site  
# chmod 755 /opt/site
```
- Kemudian ubah konfigurasi direktif DocumentRoot yang terdapat pada file /etc/apache2/sites-enabled/000-default.conf dan /etc/apache2/apache2.conf, temukan semua entri yang mengandung keterangan direktori /var/www dan ganti dengan /opt/site .
- Selanjutnya restart apache HTTP server
- Dan buatlah sebuah file index.html yang diletakkan atau disimpan kedalam direktori /opt/site. Dan isilah file index.html dengan isi sebagai berikut:

```
<HTML>  
<HEAD><TITLE>New DocumentRoot</TITLE></HEAD>  
<BODY>  
<H2>DocumentRoot : /opt/site </H2>  
</BODY>  
</HTML>
```
- Selanjutnya coba akses web server Anda dengan mengetikkan URL <http://localhost> pada web browser, dan perhatikan sekarang tampilan halaman web akan menunjukkan pesan sebagaimana yang tercantum dalam file /opt/site/index.html

Lab 11.8. Pengaturan VirtualHost

- Virtualhost adalah suatu fitur yang disediakan oleh apache HTTP server untuk memungkinkan web server apache Anda menjadi seperti web hosting. Yakni menjadi penyedia ruang atau space penyimpanan dokumen web yang masing masing berdiri sendiri sebagai sebuah web yang disebut virtualhost. Jadi Anda memungkinkan untuk membuat web site atau aplikasi web yang berbeda beda dan diletakkan pada web server yang sama.
- Masing masing virtualhost akan memiliki lokasi penyimpanan dokumen web masing masing.
- Misalnya, pada web server apache Anda, akan dibuat 2 buah virtualhost, masing masing dengan nama host(web) www.sttnf.ok dan www.nf.ok .
- Lokasi penyimpanan dokumen web untuk web www.sttnf.ok adalah pada direktori /var/sttnf/html, sedangkan lokasi penyimpanan dokumen web untuk web www.nf.ok adalah pada direktori /var/nf/html

- Maka langkah langkah konfigurasi virtualhost yang Anda harus lakukan adalah sebagai berikut:
 - Buat direktori /var/sttnf/html dan /var/nf/html , dan ubah permission nya agar user apache dapat mengakses direktori tersebut (contoh: `chmod 755 /var/sttnf/html`)
 - Buat file (kosong) baru dengan nama “vhost.conf” dan diletakkan pada direktori /etc/apache2/sites-enabled . Dengan perintah berikut:
touch /etc/apache2/sites-enabled/vhost.conf
 - Selanjutnya tambahkan baris baris berikut ini kedalam file vhost tersebut:

```
<VirtualHost *:80>
ServerName www.sttnf.ok
DocumentRoot /var/sttnf/html

<Directory /var/sttnf/html/>
    Options All
    AllowOverride All
    Require all granted
</Directory>
</VirtualHost>

<VirtualHost *:80>
ServerName www.nf.ok
DocumentRoot /var/nf/html

<Directory /var/nf/html/>
    Options All
    AllowOverride All
    Require all granted
</Directory>
</VirtualHost>
```

- Selanjutnya edit file /etc/apache2/apache2.conf, tambahkan baris berikut ini dibaris paling akhir , namun diatas direktif “Include sites-enabled/”:

```
NameVirtualHost *:80
```

- Dan buatlah file index.html pada masing masing DocumentRoot. Agar berbeda, isilah dengan pesan/keterangan yang berbeda.
- Kemudian restart service apache HTTP server
- Dan tambahkan entri berikut ini pada file /etc/hosts :
127.0.0.1 www.sttnf.ok
127.0.0.1 www.nf.ok
- Kemudian uji dengan mengakses masing masing virtualhost menggunakan web browser, dengan mengetikkan alamat URL <http://www.sttnf.ok> dan <http://www.nf.ok> . Buktikan hasilnya !.