



**SMK-TI
TRAINING
AND CERTIFICATION**

Modul 5

**WEB SERVER
(APACHE & PHP)**

**SMK-TI
TRAINING
AND CERTIFICATION**

Tujuan:

- ❖ Siswa dapat mengetahui Apache & PHP sebagai web server
- ❖ Siswa dapat menginstall Apache & PHP dengan format tar.gz
- ❖ Siswa dapat memulai Apache & PHP setelah instalasi
- ❖ Siswa dapat melakukan konfigurasi-konfigurasi yang diperlukan pada web server tersebut

1. PENDAHULUAN

Web Server adalah software server yang menjadi tulang belakang dari World Wide Web (WWW). Web server menunggu permintaan dari client yang menggunakan browser seperti netscape navigator, Internet Explorer, mozilla, dan program browser lainnya. Jika ada permintaan dari browser, maka web server akan memproses permintaan itu dan kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Data ini mempunyai format yang standar disebut dengan format SGML (Standard General Markup Language). Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemampuan browser itu. Contohnya ialah bila data yang dikirim berupa data gambar, browser yang hanya mampu menampilkan text (misalnya lynx) tidak akan mampu menampilkannya dan jika ada akan menampilkan alternatifnya saja.

Web Server, untuk berkomunikasi dengan clientnya (web browser) mempunyai protokol sendiri yaitu HTTP (HyperText Transfer Protocol). Dengan protokol ini, komunikasi antar web server dengan clientnya (browser) dapat saling dimengerti dan lebih mudah.

Seperti telah dijelaskan diatas, Standar format data pada World Wide Web adalah SGML. Tapi sudah menjadi hal yang umum bahwa para pengguna internet lebih banyak menggunakan format HTML (HyperText Markup Language) karena penggunaannya yang lebih sederhana dan mudah dipelajari.

Kata HyperText mempunyai arti bahwa seorang pengguna internet dengan web browsernya dapat membuka dan membaca dokumen-dokumen yang ada dalam komputernya atau bahkan komputer yang jauh tempatnya sekalipun. Hal ini memberikan cita rasa dari suatu proses yang tridimensional, artinya pengguna internet dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa bagian dari halaman-halaman dokumen (web) itu.

Proses yang dimulai dari permintaan webclient (browser), diterima web server, diproses, dan dikembalikan hasil prosesnya oleh web server ke webclient lagi dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui apa yang terjadi pada tiap-tiap proses. Secara garis besarnya web server hanya memproses semua masukan yang diperolehnya dari web clientnya.

Untuk membuat sebuah web server, maka kita akan menemukan berbagai macam persoalan, dimulai dari pemilihan software web browser yang manayang paling sesuai kebutuhan, apa spesifikasi hardware yang dibutuhkan, bagaimana kondisi interkoneksi jaringan internet yang ada, dan lain sebagainya. Belum lagi termasuk bagian pembuatan halaman-halaman webnya, mau menggunakan format apa (HTML, SGML, PHP, PHP3, CGI, dan lain-lain). Hal yang paling utama dalam proses pembuatan Web Server adalah memilih software mana yang akan digunakan sebagai web server kita. Untuk itu, perlu adanya beberapa pertimbangan sebagai berikut :

1. Komersial lawan Freeware (software gratis).
2. Kemudahan instalasi
3. Kemudahan mengonfigurasi
4. Kemudahan untuk menambah atau mengubah periferalnya.
5. Kemampuan software.
6. Besar ruang yang dibutuhkan untuk menyimpan file-file minimal yang dibutuhkan agar software berfungsi dengan baik
7. Prospek software tersebut dimasa yang akan datang
8. Performasi dan konsumsi sumber daya yang digunakan software itu.
9. Fasilitas apa yang mampu didukung software itu.
10. Dukungan Teknis (mempunyai site-site atau mailis untuk bertanya bila terjadi masalah).
11. Dukungan Platform (jenis sistem operasi apa saja yang dapat menjalankan software tersebut)
12. Dukungan terhadap third party (Apakah software ini dapat ditambahkan software tambahan sebagai pelengkap)

2. APACHE WEB SERVER

Banyak sekali software web server yang berada di internet. Dengan berdasarkan pada 12 macam pertimbangan diatas, maka dapat dipilih software mana saja yang cocok dengan kebutuhan kita. Misalnya : Kita memasang web server untuk keperluan suatu perusahaan jasa internet (ISP), maka pertimbangan yang harus diambil adalah apakah mereka menginginkan software yang gratis atau komersial. Keuntungan dari software komersial adalah mereka punya dukungan teknis dan dokumentasi yang lengkap, sedang pada kebanyakan software gratis tidak punya. Namun ada juga software gratisan yang mempunyai dukungan teknis dari pembuatnya dan dengan dokumentasi yang lengkap. Salah satu software web server gratisan seperti itu adalah Web server Apache.

Web server Apache mempunyai kelebihan dilihat dari beberapa pertimbangan diatas:

Apache termasuk dalam kategori freeware (software gratisan)

Apache mudah sekali proses instalasinya jika dibanding web server lainnya seperti NCSA, IIS, dan lain-lain.

Mampu beroperasi pada berbagai platform sistem operasi seperti : AIX 3.1, BSDI 2.0, FreeBSD 2.1, HP-UX 9.07, IRIX 5.3, Linux, NetBSD 1.1, NEXTSTEP, SolarisX86 2.5, Solaris 2.4, Solaris 2.5, SunOS 4.1.3, UnixWare 1.1.2.

Apache mudah untuk mengkonfigurasinya karena hanya mempunyai satu file konfigurasi.

Apache Web server mudah dalam menambahkan periferal lainnya ke dalam platform web servernya, misalnya : untuk menambahkan modul, cukup hanya menset file konfigurasinya agar mengikutsertakan modul itu kedalam kumpulan modul lain yang sudah dioperasikan.

Features atau ciri khas dari web server Apache adalah :

1. Dapat dijadikan pengganti bagi NCSA web server.
2. Perbaikan terhadap kerusakan dan error pada NCSA 1.3 dan 1.4

3. Apache web server dalam merespon client sangat cepat jauh melebihi server NCSA.
4. Mampu di kompilasi sesuai dengan spesifikasi HTTP yang sekarang.
5. Apache menyediakan feature untuk multihomed dan virtual server.
6. Kita dapat men-set respon error yang akan dikirim web server dengan menggunakan file atau skrip.
7. Server Apache dapat otomatis berkomunikasi dengan client browsernya untuk menampilkan yang tampilan terbaik pada client browsernya. Misalnya browser ingin menampilkan dalam bahasa Spanyol, maka Apache web server otomatis mencari dalam servicenya halaman-halaman dengan bahasa Spanyol.
8. Webserver Apache secara otomatis menjalankan file index.html, halaman utamanya, untuk ditampilkan secara otomatis pada clientnya.
9. Webserver Apache mempunyai level-level pengamanan.
10. Apache mempunyai komponen dasar terbanyak di antara webserver-webserver lain, yang berarti bahwa webserver Apache termasuk salah satu dari webserver yang lengkap.
11. Ditinjau dari segi sejarah perkembangan dan prospeknya di masa yang akan datang, Apache web server mempunyai prospek yang cerah. Apache berasal dari webserver NCSA yang kemudian dikembangkan karena NCSA masih mempunyai kekurangan dibidang kompatibilitasnya dengan sistem operasi lain. Sampai saat ini, webserver Apache terus dikembangkan oleh team dari apache.org.
12. Performansi dan konsumsi sumberdaya (resource) dari webserver apache tidak terlalu banyak, hanya sekitar 20 MB untuk file-file dasarnya dan setiap daemonnya hanya memerlukan sekitar 950 KB memory per-child.
13. Mendukung transaksi yang aman (secure transaction) menggunakan SSL (Secure Socket Layer).
14. Mempunyai dukungan teknis melalui web.
15. Mempunyai kompatibilitas platform yang tinggi.
16. Mendukung third party berupa modul-modul tambahan.

3. MODUL PHP PADA APACHE WEB SERVER

Apache web server mendukung penambahan modul-modul. Diantara modul yang sering dipakai adalah modul PHP. PHP (Personal Home Page Tool) adalah salah satu jenis alat yang digunakan untuk membuat halaman web anda menjadi lebih menarik, lebih aman, dan lebih dinamik. Pada dasarnya PHP mirip dengan bahasa script yang lainnya seperti asp, Javascript, Visual Basic atau yang lainnya, namun keuntungannya adalah pada PHP tidak diperlukan tambahan pada sisi webclient seperti halnya Javascript dan Visual Basic script, sehingga lebih luas penggunaannya.

Beberapa keunggulan PHP dibandingkan dengan yang lain :

- Autentikasi http menggunakan PHP menggunakan fungsi `header()` yang ada dalam modul apache PHP.
- Pembuatan file GIF menggunakan library GD yang dikompilasi saat menginstall php.
- PHP dapat menerima metoda upload file
- Mendukung penggunaan cookie
- Mendukung integrasi dengan database
- Mendukung ekspresi regular seperti `ereg()`, `ereg_replace()` dan lainnya.
- Penanganan kesalahan berdasarkan tingkat kesalahan.

INSTALASI APACHE dan PHP

Proses instalasi dari source agak sedikit memakan waktu sebab anda harus melakukan proses tambahan, tapi dengan begitu akan membuat anda lebih mengetahui apa saja yang anda masukkan ke dalam webserver Apache anda nantinya, disamping itu juga dapat memberikan kesempatan untuk memasang sendiri modul-modul ke dalam webserver Apache anda. Banyak dari modul-modul ini menyediakan fungsi-fungsi yang akan menambah kehandalan webserver anda. Diantaranya adalah modul PHP yang akan diinstall bersamaan dengan instalasi Apache. Instalasi yang dilakukan di modul ini juga akan mengintegrasikan Apache dan PHP dengan MySQL sebagai database server.

Sebelum instalasi anda harus menyediakan terlebih dahulu source yang diperlukan yang dapat didownload di :

<http://www.apache.org>
<http://www.php.net>

Anda memilih sendiri versi apache dan php yang ingin diinstall di mesin anda. Tentunya memilih versi yang lebih stabil hasilnya. Setelah anda download source tersebut sebaiknya semua source ditaruh dalam satu direktori untuk memudahkan proses instalasi, misalnya di direktori /tmp, lalu diekstrak.

```
tar xvf apache_verisi.tar.gz
tar xvf php-versi.tar.gz
```

Untuk apache harus dikonfigurasi awal.

```
cd apache_verisi/
./configure
```

Setelah konfigurasi awal sudah selesai, anda pindah ke direktori php untuk konfigurasi modul php dan mengintegrasikannya ke dalam apache dan mysql

```
cd php-versi/
./configure --with-apache=../apache_verisi \
            --with-mysql --enable-track-vars
```

Setelah itu php dikompilasi dan diinstall

```
# make && make install
```

Kemudian anda copykan file php.ini-dist yang ada di source php ke direktori /usr/local/lib dan diberi nama php.ini

```
# cp php.ini-dist /usr/local/lib/php.ini
```

Kemudian anda kembali ke direktori apache untuk konfigurasi ulang dan mengaktifkan modul php (yaitu php4).

```
# cd ../apache_verisi
# ./configure --activate-module=src/modules/php4/libphp4.a
```

Lalu kompilasi dan install apache

```
# make && make install
```

Sampai disini instalasi apache dan PHP sudah selesai. Selanjutnya anda harus melakukan konfigurasi apache agar berjalan sesuai dengan kebutuhan anda

5. MEMONITOR AKTIFITAS WEB SERVER

Pada bagian sebelumnya, telah dipelajari penginstallasian dan pengoperasian WebServer. Pengoperasian WebServer termasuk juga penggunaan pencatatan atau logger. Ada tiga macam log yang harus diperhatikan yaitu server log, access log dan error log.

server.log

Adalah logging yang dilakukan oleh server yang dicatat dalam file /var/log/messages. Fungsi file ini adalah mencatat kejadian-kejadian tertentu pada server anda. Namun demikian, file ini hanya diperiksa apabila ternyata webserver tidak jalan karena sesuatu kesalahan. Biasanya tidak hanya melihat file ini saja, tapi juga harus melihat file log yang lainnya.

access.log

File ini mencatat semua akses yang dilakukan terhadap web server anda. Sebenarnya program penganalisa juga mengambil data dari file ini kemudian diproses menjadi data statistik yang enak dibaca dan dimengerti orang.

Misal konfigurasi log anda pada file httpd.conf adalah sebagai berikut :

```
ErrorLog /web/logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog /web/logs/access_log common
```

Maka akan tampil pada file httpd-access.log format seperti ini :

```
167.205.206.101 - - [26/Jun/2001:07:09:46 +0700] "GET / HTTP/1.0" 200
1310
167.205.22.105 - - [26/Jun/2001:07:10:50 +0700] "GET /usage/index.html
HTTP/1.0"
403 296
167.205.22.105 - - [26/Jun/2001:07:11:27 +0700] "GET /test.php HTTP/1.0"
200 387
46
167.205.15.205 - - [26/Jun/2001:07:17:08 +0700] "GET / HTTP/1.0" 200 1310
167.205.15.198 - - [26/Jun/2001:07:17:20 +0700] "GET / HTTP/1.0" 200 1310
167.205.206.104 - - [26/Jun/2001:09:42:31 +0700] "GET /icons/unknown.gif
HTTP/1.
1" 404 305
167.205.206.104 - - [26/Jun/2001:09:42:31 +0700] "GET /
icons/compressed.gif HTTP
/1.1" 404 308
167.205.206.104 - - [26/Jun/2001:09:42:31 +0700] "GET /icons/binary.gif
HTTP/1.1
" 404 304
167.205.206.104 - - [26/Jun/2001:09:42:36 +0700] "GET / HTTP/1.1" 200
1310
167.205.206.104 - - [26/Jun/2001:09:42:36 +0700] "GET /apache_pb.gif
HTTP/1.1" 2
00 2326
167.205.206.104 - - [26/Jun/2001:09:42:43 +0700] "GET /~wahid HTTP/1.1"
301 338
167.205.206.104 - - [26/Jun/2001:09:42:43 +0700] "GET /icons/blank.gif
HTTP/1.1"
404 303
167.205.206.104 - - [26/Jun/2001:09:42:43 +0700] "GET /icons/unknown.gif
HTTP/1.
1" 404 305
167.205.206.104 - - [26/Jun/2001:09:42:43 +0700] "GET /icons/back.gif
HTTP/1.1"
404 302
```

```

167.205.206.104 - - [26/Jun/2001:09:42:43 +0700] "GET /
icons/compressed.gif HTTP
/1.1" 404 308
167.205.206.104 - - [26/Jun/2001:09:42:43 +0700] "GET /icons/binary.gif
HTTP/1.1
" 404 304

```

Tiap-tiap baris menunjukkan format seperti yang dijelaskan sebelumnya. Semua hasil log tidak mempunyai remote Logname dan remote user, karena secara default kedua log environment ini memang tidak diisi, namun ada juga yang mengisinya untuk tujuan tertentu misalnya pada aplikasi yang menggunakan cookies.

Penggunaan terlalu banyak aksesoris pada web akan ikut mempengaruhi log anda. Tentu akan semakin mempercepat pembengkakan access.log karena apache akan mencatat setiap proses pengambilan data baik gambar, maupun tulisan. Penggunaan script atau Active-X justru menghemat space untuk log anda.

error.log

File ini mencatat setiap kesalahan yang terjadi pada web server, apakah kesalahan itu pada file konfigurasinya sehingga apache tidak mau jalan, atau hanya berupa kesalahan pada pembuatan webnya. Pada error.log, LogLevel berpengaruh dari segi seberapa banyak informasi kesalahan yang akan dituliskan dalam file error.log ini.

Contoh tampilan file error.log adalah :

```

[Tue Jun 26 04:24:32 2001] [error] [client 167.205.206.107] no acceptable
varian
t: /web/htdocs/index.html
[Tue Jun 26 04:25:00 2001] [error] [client 167.205.206.107] no acceptable
varian
t: /web/htdocs/index.html
[Tue Jun 26 04:25:17 2001] [error] [client 167.205.206.107] File does not
exist:
/web/htdocs/~index.html
[Tue Jun 26 04:25:29 2001] [error] [client 167.205.206.107] File does not
exist:
/web/htdocs/ndex.html
[Tue Jun 26 04:29:25 2001] [error] [client 167.205.206.104] client denied
by ser
ver configuration: /web/htdocs/usage
[Tue Jun 26 04:34:03 2001] [notice] SIGHUP received. Attempting to
restart
[Tue Jun 26 04:34:03 2001] [notice] Apache/1.3.17 (Unix) mod_perl/1.24_01
PHP/4.
0.4p11 configured -- resuming normal operations
[Tue Jun 26 06:07:55 2001] [notice] caught SIGTERM, shutting down
[Tue Jun 26 06:09:25 2001] [notice] Apache/1.3.17 (Unix) mod_perl/1.24_01
PHP/4.
0.4p11 configured -- resuming normal operations
[Tue Jun 26 06:25:50 2001] [notice] caught SIGTERM, shutting down
[Tue Jun 26 06:27:25 2001] [notice] Apache/1.3.17 (Unix) mod_perl/1.24_01
PHP/4.
0.4p11 configured -- resuming normal operations
[Tue Jun 26 07:10:50 2001] [error] [client 167.205.22.105] client denied
by serv

```

```

er configuration: /web/htdocs/usage/index.html
[Tue Jun 26 09:42:02 2001] [error] [client 167.205.206.104] File does not
exist:
/web/htdocs/icons/image2.gif
[Tue Jun 26 09:42:12 2001] [error] [client 167.205.206.104] File does not
exist:
/web/htdocs/icons/blank.gif
[Tue Jun 26 09:42:12 2001] [error] [client 167.205.206.104] File does not
exist:
/web/htdocs/icons/back.gif
[Tue Jun 26 09:42:12 2001] [error] [client 167.205.206.104] File does not
exist:
/web/htdocs/icons/image2.gif
[Tue Jun 26 09:42:31 2001] [error] [client 167.205.206.104] File does not
exist:
/web/htdocs/icons/back.gif
[Tue Jun 26 09:42:31 2001] [error] [client 167.205.206.104] File does not
exist:
/web/htdocs/icons/blank.gif
[Tue Jun 26 09:42:31 2001] [error] [client 167.205.206.104] File does not
exist:
/web/htdocs/icons/unknown.gif

```

Jadi isi file ini adalah pesan-pesan kesalahan (error) dan juga pemberitahuan pengoperasian httpdnya (notice). Dengan melihat error.log, akan ketahuan apakah web anda mengalami kesalahan pada waktu pembuatan, misal pada contoh file does not exist menandakan bahwa ada file yang hilang atau tidak sempat di ambil.

6. MENGAMATI LOG MENGGUNAKAN SOFTWARE WEBALIZER

Selain melihat secara langsung file-file log tersebut, ada cara yang lebih mudah, yaitu menggunakan software logger seperti webalizer. Software ini akan membaca file acces.log dan melakukan perhitungan. Sebelum menginstallnya anda ambil sourcenya di ftp sitenya (<ftp://ftp.mrunix.net/pub/webalizer>). Untuk modul ini digunakan webalizer versi 1.30

Kemudian instalasi dapat dilakukan dengan perintah sebagai berikut :

```

# tar zxvpf webalizer-1.30-04-src.tar.Z
# cd webalizer-1.30-04/
# ./configure --enable-dns
Lebih jelasnya lihat helpnya (./configure -help)

# make && make install

```

Setelah selesai installasinya, maka anda copy file /etc/webalizer.conf.sample ke /usr/local/etc/webalizer.conf kemudian editlah file ini. Ubah bagian yang penting saja, misalkan direktori web anda ada di /web, maka perubahannya adalah :

```

LogFile          /web/logs/access_log
OutputDir        /web/htdocs/stat

```

```
HistoryName      webalizer.hist
IncrementalName  webalizer.current
HostName         nama.host.anda
HideSite        *nama.host.anda
```

Setelah itu, maka jalankan saja perintah :

```
# /usr/local/bin/webalizer -c /usr/local/etc/webalizer.conf
```

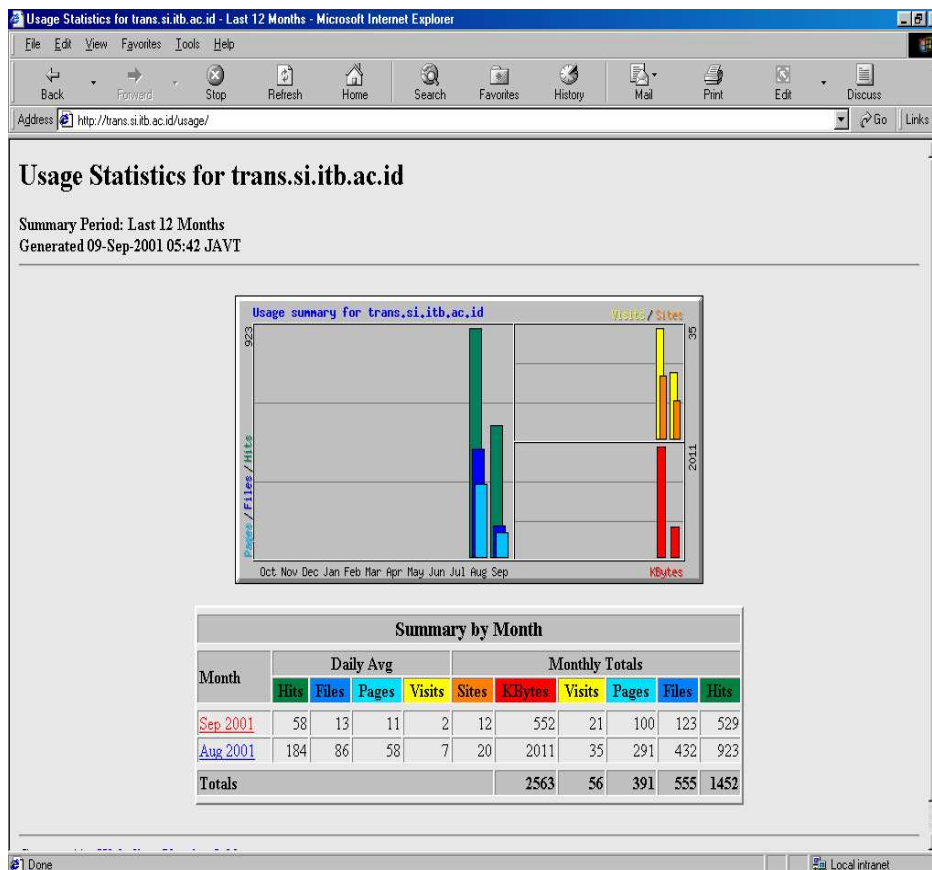
secara otomatis, program akan menganalisa data-data pada access.log dan mengubahnya menjadi data stasitik. Akan muncul pesan sbb :

```
Webalizer V1.30-04 (Linux 2.2.14-5.0) English
Using logfile /var/httpd/logs/access_log
Creating output in /var/httpd/hdocs/usage
Hostname for reports is 'trans.si.itb.ac.id'
History file not found...
Warning: Truncating oversized request field
Warning: Truncating oversized request field
Warning: Truncating oversized request field
Generating report for September 2001
Generating summary report
Saving history information...
1452 records in 0.56 seconds
```

Setelah selesai, maka akan diperoleh pada direktori outputdir seperti berikut :

```
-rw-r--r--      1 root      root      16209 Sep  9 05:42
ctry_usage_200108.gif
-rw-r--r--      1 root      root      16261 Sep  9 05:42
ctry_usage_200109.gif
-rw-r--r--      1 root      root      23265 Sep  9 05:42
daily_usage_200108.gif
-rw-r--r--      1 root      root      25767 Sep  9 05:42
daily_usage_200109.gif
-rw-r--r--      1 root      root      18517 Sep  9 05:42
hourly_usage_200108.gif
-rw-r--r--      1 root      root      17620 Sep  9 05:42
hourly_usage_200109.gif
-rw-r--r--      1 root      root      4295 Sep  9 05:42 index.html
-rw-r--r--      1 root      root      19205 Sep  9 05:42 usage.gif
-rw-r--r--      1 root      root      69127 Sep  9 05:42 usage_200108.html
-rw-r--r--      1 root      root      61864 Sep  9 05:42 usage_200109.html
-rw-r--r--      1 root      root        69 Sep  9 05:42 webalizer.hist
```

Sedangkan tampilan pada index.htmlnya adalah sebagai berikut :



Selesailah installasi program logger webalizer. Perlu diingat bahwa program ini harus dijalankan periodik secara manual atau dapat ditambahkan pada script /etc/monthly yang akan menjalankan program ini secara otomatis tiap bulannya.

7. MENGATUR AKSES KE WEB SERVER APACHE

Terkadang ada beberapa bagian web yang tidak boleh diakses oleh sembarangan user. Bagaimana membatasi hak akses ini ? Ada beberapa cara yang dapat dilakukan, antara lain menggunakan pembatasan pada file konfigurasinya atau menggunakan file ACL.

✓ Menggunakan pembatasan pada file konfigurasi.

Dalam file konfigurasi (httpd.conf) terdapat tag Directory yang berfungsi mendefinisikan permission suatu direktori yang diatur oleh tag-tag didalam Tag Directory tersebut. Sintaknya :

```
<Directory directory> ... </Directory>
```

Tag-tag yang mengatur akses dalam tag directory antara lain Order, allow dan deny.

Tag Order mempunyai tiga kemungkinan :

- `order allow,deny` → perintah `allow` dievaluasi dahulu daripada `deny` (Default inisialisasi `forbidden`),
- `order deny,allow` → `deny` dievaluasi dahulu baru `allow` (Default inisialisasi `OK`),
- `order mutual-failure` → hanya `host` yang muncul di tag `allow` dan tidak muncul di tag `deny` yang boleh mengakses.

Tag `allow` dan `deny` mendefinisikan hak akses dari suatu `host` atau `domain` tertentu.

Kedua tag ini mempunyai sintak yang sama yaitu :

```
[Allow,deny] from [All,.domain.com,167.205.0.0/8]
```

Selain ketiga Tag ini, ada Tag lain yang menspesifikasi akses `user` dan `group` yaitu tag `require` dengan sintak :

```
require [user user_id ,group gourp_id ,valid-user].
```

Bila tag `allow/deny` dan `require` dipakai bersama-sama, ada satu tag lagi yang memastikan apakah kedua tag tersebut bekerja `cooperative` atau hanya salah satu saja, yaitu tag `satisfy` dengan sintak :

```
satisfy [any,all]
```

dimana `any` akan memberikan hak akses bila salah satu `permission` (`host` atau `user/group`) terpenuhi, sedang `all` memerlukan kedua tag terpenuhi.

Selain Tag `Directory`, masih ada tag-tag lain yang digunakan untuk membatasi hak akses.

Tag `files` membatasi hak akses terhadap file-file berektensi dan Tag `Location` membatasi hak akses terhadap URL tertentu. Sintaks kedua tag ini sama dengan sintak `directory`

```
<Files nama_file> ... </Files>  
<Location URL> ... </Location>
```

Contoh penggunaan tag-tag diatas sebagai berikut :

```
<Directory /home/*/public_html>  
    AllowOverride All  
    Options All  
    <Limit GET POST OPTIONS PROPFIND>  
        Order allow,deny  
        Allow from all  
    </Limit>  
    <LimitExcept GET POST OPTIONS PROPFIND>  
        Order deny,allow  
        Deny from all
```

```

        </LimitExcept>
    </Directory>
    <Files ~ "\.(MP3)$">
        Order deny,allow
        Deny from all
        Allow from 167.205
    </Files>
    <Files ~ "\.(asf)$">
        Order deny,allow
        Deny from all
        Allow from 167.205
    </Files>

```

Ini berarti semua direktori `public_html` user diperbolehkan untuk diakses. Sedangkan file-file dengan format `.mp3` dan `.asf` hanya bias diakses dari IP `167.205.*`.

✓ Penggunaan file ACL

Konfigurasi file ACL mengatur hak akses seperti halnya pada file `httpd.conf`. File ACL mengatur hak akses pada masing-masing direktori, jadi menggantikan Tag `Directory`. File ACL didefinisikan pada `httpd.conf` yaitu pada tag `AccessFileName`. Sintaks defaultnya :

```
AccessFileName .htaccess
```

Untuk dapat menggunakan file ACL ini, maka Pada Tag `AllowOverride` dalam tag `Directory` untuk `DocumentRoot`nya harus diberi option `All`, artinya tiap direktori mempunyai hak akses sendiri-sendiri tergantung file ACL atau mengikuti default permission dari `RootDirectory`nya.

Contoh filenya adalah :

```

IndexIgnore .htaccess */.??* *~ *# */HEADER* */README* */_vti*

<Limit GET POST>
order deny,allow
#allow from all
require user abdul
</Limit>
<Limit PUT DELETE>
order deny,allow
deny from all
</Limit>
AuthType Basic
AuthName salman.itb.ac.id
AuthUserFile /home/abdul/public_html/.pwd

```