

# Apache Web Server



Distro Linux besar seperti RedHat, Slackware, dan Mandrake sudah menyertakan Apache dalam iso atau CD-nya. Namun, Anda juga dapat menginstalasi paket Apache terbaru, terutama untuk menjaga keamanan dan kestabilannya.

Pada saat kita berselancar di Internet dengan menggunakan *browser* atau istilah kerennya *browsing*, kita pasti akan menuju pada suatu alamat situs yang di dalamnya memuat berbagai macam data, informasi, dan hiburan. Tentunya kita yang masih awam akan penasaran, bagaimana data atau file tersebut bisa dimunculkan? Dan bagaimana cara kerja atau proses sehingga data/file tersebut bisa diakses secara serentak pada PC kita? Marilah kita bahas hal ini bersama.

Pada umumnya, semua situs web di dunia ini digerakkan oleh suatu program aplikasi yang berjalan di *server*. Program tersebut, yang sering dipakai adalah *Apache*, karena selain berlisensi GPL (*General Public License*) atau *free software*, juga mudah dikonfigurasi. Sedangkan aplikasi yang menjalankan program apache tersebut biasa dinamakan *Web Server* atau *httpd*.

Apache Web Server merupakan program aplikasi yang berjalan di server, berfungsi untuk menjalankan aplikasi web sehingga bisa diakses oleh klien baik melalui jaringan intranet maupun Internet.

## Instalasi Apache Web Server

Ada dua cara untuk melakukan instalasi dan kompilasi Apache, yaitu dengan menggunakan metode *rpm* dan menggunakan metode kompilasi file *tar.gz*. Adapun langkah-langkahnya adalah sebagai berikut:

### Download Apache

Untuk melakukan *download* Apache, Anda bisa mengambil file-nya di alamat situs <http://www.apache.org/dist/httpd/>

### Instalasi Paket RPM

Biasanya pada distro-distro Linux, seperti: SuSE, TSL, RedHat, dan lain-lain Apache

sudah dipaketkan dalam bentuk file ber-ekstensi *rpm*. Untuk melakukan instalasi dengan metode ini, Anda bisa melakukan langkah-langkah sebagai berikut:

1. Pada konsol, lakukan login sebagai root.

```
login: root
```

```
Password: password (ganti dengan
password Anda)
```

Anda juga bisa menjadi *root*, pada saat Anda login sebagai user biasa atau pada waktu menggunakan X Window, dengan mengetikkan sintaks perintah pada konsol sebagai berikut:

```
[user@localhost: ~] $ su -l root
```

```
Password: (isi dengan password root Anda)
```

Pada contoh ini, kami menggunakan file *apache\_1.3.27.i-386.rpm*, Anda bisa menggunakan versi Apache lainnya, sesuai dengan kebutuhan.

2. Masuk ke direktori, di mana file *apache\_1.3.27.i-386.rpm* berada.
3. Lakukan kompilasi, dengan mengetikkan perintah sebagai berikut:

```
[root@localhost: ~] # rpm -ivh
```

```
apache_1.3.27.i-386.rpm
```

```
apache #####
#####
```

4. Kemudian, untuk menjalankan Apache Web Server, Anda bisa membaca pada bagian Menjalankan Apache Web Server.

## Instalasi paket tar.gz

Jika sudah melakukan *download source* Apache, Anda bisa melakukan kompilasi dan instalasi Apache sebagai berikut:

1. Pada konsol, lakukan login sebagai root.

```
login: root
```

```
Password: password (ganti dengan
password Anda)
```

Anda juga bisa menjadi *root*, pada saat Anda login sebagai user biasa atau pada waktu menggunakan X Window, dengan mengetikkan sintaks perintah pada konsol sebagai berikut:

```
[user@localhost: ~] $ su -l root
```

```
Password: (isi dengan password root Anda)
```

Pada contoh ini, kami menggunakan file *apache\_1.3.27.tar.gz*, Anda bisa menggunakan versi Apache lainnya, sesuai dengan keinginan dan kebutuhan Anda.

2. Pindahkan file *apache\_1.3.27.tar.gz* ke direktori */usr/src*, dengan mengetikkan sintaks perintah sebagai berikut:

```
[root@localhost: user/] # mv
```

```
apache_1.3.27.tar.gz /usr/src
```

3. Masuk ke direktori */usr/src*, di mana file *apache\_1.3.27.tar.gz* berada, dengan mengetikkan perintah sebagai berikut:

```
[root@localhost: ~] # cd /usr/src
```

4. Lakukan ekstraksi file dengan mengetikkan perintah sebagai berikut:

```
[root@localhost: src/] # tar -zxvf
```

```
apache_1.3.27.tar.gz
```

5. Masuk ke subdirektori *apache\_1.3.27*

```
[root@localhost: src/] # cd apache_1.3.27
```

6. Lakukan kompilasi dengan mengetikkan sintaks perintah sebagai berikut:

```
[root@localhost: apache_1.3.27/] # ./
```

```
configure \
```

```
--prefix=/usr/local/apache
```

**Tip:** Anda bisa mengganti *--prefix=/usr/local/apache* dengan direktori yang sesuai dengan kebutuhan Anda.

Lanjutkan dengan mengetikkan perintah berikut ini:

```
[root@localhost: apache_1.3.27/] # make
```

```
[root@localhost: apache_1.3.27/] #
```

```
make install
```

## Setting Apache secara manual

Untuk melakukan setting Apache secara manual, Anda bisa melakukan editing pada file `/etc/httpd/conf/httpd.conf`.

Sekarang mari kita bahas bersama setting manual konfigurasi Apache Web Server ini. Adapun langkah-langkahnya adalah sebagai berikut:

1. Pada konsol, lakukan login sebagai root

Login: root

Password: password

2. Lakukan editing file `/etc/httpd/conf/httpd.conf`, sebagai berikut:

```
[root@localhost: ~] # vi /etc/httpd/conf/httpd.conf
```

3. Biasanya file `httpd.conf` sudah berisi file konfigurasi, namun Anda bisa juga menambahkan sesuai dengan kalimat di bawah ini:

ServerType stAndalone

ServerRoot /etc/httpd

ServerName server.net

ServerAdmin user@localhost

LockFile /etc/httpd/httpd.lock

PidFile /var/run/httpd.pid

ScoreBoardFile /etc/httpd/httpd.scoreboard

ErrorLog logs/error\_log

LogLevel warn

DocumentRoot /var/www/html

Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 15

MinSpareServers 4

MaxSpareServers 10

StartServers 4

MaxClients 150

MaxRequestsPerChild 500

Include conf/addon-modules/php.conf

Include conf/ssl/mod\_ssl.conf

#BindAddress \*

<IfDefine APACHEPROXIED>

Port 8080

Listen 8080

</IfDefine>

Adapun penjelasan dari konfigurasi di atas adalah sebagai berikut:

**ServerType stAndalone**

→ Pilihan untuk menjalankan Server

Apache sebagai daemon atau melalui inetd. Pilihan inetd biasanya digunakan pada platform Unix.

**ServerRoot /etc/httpd**

→ Direktori utama, di mana file konfigurasi Apache diletakkan.

**ServerName server.net**

→ Dipergunakan untuk meletakkan nama domain Anda, Anda juga bisa menggantikan dengan nomor IP.

**ServerAdmin user@localhost**

→ Alamat yang dipergunakan untuk memberitahukan halaman-halaman yang dibuat secara otomatis oleh server, seperti pesan kesalahan.

**LockFile /etc/httpd/httpd.lock**

→ Direktori yang berisi file lock dari Apache.

**PidFile /var/run/httpd.pid**

→ File yang dipergunakan oleh Server untuk merekam nomor identitas proses (PID).

**ScoreBoardFile /etc/httpd/httpd.scoreboard**

→ File yang berguna untuk merekam proses internal pada Server.

**ErrorLog logs/error\_log**

→ Letak file `error_log`, biasanya terdapat pada direktori lokal.

**LogLevel warn**

→ Jumlah pesan kesalahan yang dicatat dalam file `error_log`. Nilai yang dimasukkan, antara lain `debug`, `warn`, `notice`, `alert`, `error`, `info`, `crit`, `emerg`.

**DocumentRoot /var/www/html**

→ Direktori di mana file `index.html` akan ditempatkan, `index.html` merupakan file yang kali pertama dibaca oleh klien pada saat melakukan akses ke Server.

**Timeout 300**

→ Lama waktu/jeda sebelum Server mengeluarkan pesan time-out.

**KeepAlive On**

→ Untuk menentukan, apakah Server

mengijinkan lebih dari satu permintaan dalam satu koneksi.

**MaxKeepAliveRequests 100**

→ Jumlah maksimum permintaan yang diijinkan dalam satu koneksi.

**KeepAliveTimeout 15**

→ Waktu penantian, dihitung dalam detik untuk menunggu permintaan berikutnya dari klien.

**MinSpareServers 4**

→ Pilihan untuk mengatur proses yang dijalankan oleh Apache. Nilai spare minimum untuk membuat spare baru.

**MaxSpareServers 10**

→ Nilai spare maksimum spare yang diizinkan.

**StartServers 4**

→ Jumlah server yang berjalan pada saat diaktifkan.

**MaxClients 150**

→ Jumlah maksimum klien yang diijinkan untuk mengakses pada saat yang bersamaan.

**MaxRequestsPerChild 500**

→ Maksimum permintaan untuk mempergunakan child process.

**Include conf/addon-modules/php.conf**

→ Penyertaan file konfigurasi PHP.

**Include conf/ssl/mod\_ssl.conf**

→ Penyertaan file konfigurasi `mod_ssl`.

**#BindAddress \***

→ Dukungan virtual host dengan memerintahkan Apache untuk mengikat beberapa alamat.

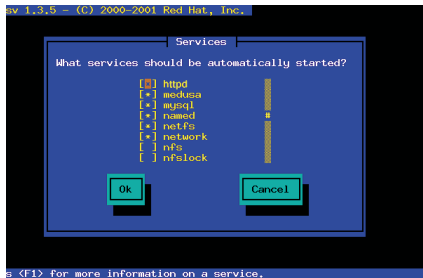
**<IfDefine APACHEPROXIED>**

Port 8080

Listen 8080

**</IfDefine>**

→ Bila Anda menjalankan Proxy Server, Apache akan menggunakan alamat port 8080, angka ini bisa diganti sesuai keinginan Anda, jangan lupa untuk mengubah NAT-nya sekalian.



▲ Gambar 1. Ntsysv memilih aplikasi yang jalan saat boot

4. Anda juga bisa menambahkan virtual host jika ada, caranya dengan menambahkan kalimat berikut ini:

```
<VirtualHost 192.168.0.1>
ServerAdmin webmaster@ai.co.id
DocumentRoot/www/docs/masaji.ai.co.id
ServerName masaji.ai.co.id
</VirtualHost>
```

5. Ini merupakan cara pembuatan virtual domain, jika ada. Anda tinggal mengganti domain di atas dengan domain Anda sendiri.
6. Jika sudah, selesai Anda bisa menjalankan Apache Web Server, untuk menjalankannya Anda tinggal mengetikkan perintah sebagai berikut:

```
[root@localhost: ~] # /etc/init.d/httpd start
```

## Menjalankan Apache

Setelah Anda selesai melakukan konfigurasi, Apache sudah siap dijalankan dengan menggunakan perintah sebagai berikut:

```
[root@localhost: ~] # httpd &
```

Pada paket rpm dari *default* instalasi RedHat, file-file log ini terdapat pada direktori `/var/log/httpd`.

Supaya Apache ini dapat otomatis langsung dijalankan ketika Anda *booting* Linux kali pertama, ada tiga cara:

## Menggunakan metode konvensional

Dalam hal ini, kita akan memasukkan program Apache pada file `/etc/rc.d/rc.local`. Adapun caranya adalah sebagai berikut:

1. Lakukan login sebagai *root* atau menggunakan *substitute user*:

```
[user@localhost: ~] $ su -r root
```

Password:

root in: user #

2. Edit file `/etc/rc.d/rc.local`

```
[root@localhost: ~] # vi /etc/rc.d/rc.local
```

3. Sisipkan kalimat berikut ini:

```
/usr/sbin/httpd &
```

4. Simpan hasil pekerjaan Anda
5. Restart PC Anda

## Menggunakan chkconfig

1. Lakukan login sebagai *root* atau menggunakan *substitute user*:

```
[user@localhost: ~] $ su -r root
```

Password:

root in: user #

2. Masuk ke direktori `/etc/init.d/`.

```
[root@localhost: ~] # cd /etc/init.d/
```

3. Mengaktifkan skrip `httpd`.

Untuk menjalankannya secara otomatis pada saat PC dinyalakan, ketikkan perintah sebagai berikut:

```
[root@localhost: /etc/init.d] # chkconfig -l 3 5 httpd on
```

4. *Restart* PC Anda atau jalankan skrip `/etc/init.d/httpd`, jika Anda ingin langsung menjalankan Apache Web Server.

Selanjutnya sistem Anda akan langsung menjalankan Apache Web Server, setiap kali PC Anda restart.

## Menggunakan metode init

Bila Anda menginstalasi dari paket rpm redhat, akan disertakan pula skrip di direktori `/etc/rc.d/init.d/httpd`, yang secara otomatis akan memanggil `/usr/sbin/httpd`, skrip ini dapat Anda buat aktif ketika booting dengan cara sebagai berikut:

1. Lakukan login sebagai *root* atau menggunakan *substitute user*:

```
[user@localhost: ~] $ su -r root
```

Password:

[root@localhost: user] #

2. Menjalankan `ntsysv`.

Untuk mengaktifkan skrip `/etc/init.d/`

`httpd`, pada konsol ketikkan perintah sebagai berikut:

```
[root@localhost: ~] # ntsysv
```

3. Mengaktifkan Apache pada `Ntsysv`. Kemudian pilih `httpd` dan tekan tombol space-bar, seperti tampak pada Gambar 1.
4. Tekan tombol *OK*.

## Menguji web server

Sekarang Anda bisa mencoba web server Anda sendiri, dengan cara mengaksesnya melalui PC klien atau bisa juga Anda akses melalui PC Anda yang sudah diinstalasi Apache Web Server. Adapun caranya adalah sebagai berikut:

1. Login ke X Window sebagai user biasa.
2. Jalankan browser kegemaran Anda.
3. Isikan nama domain atau alamat IP Anda.

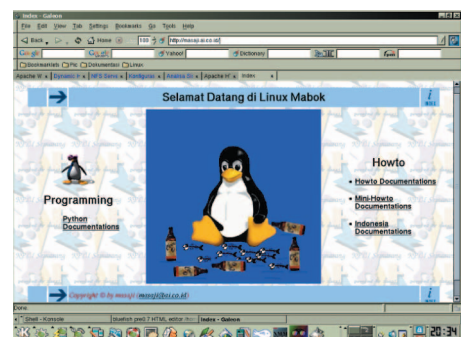
```
http://192.168.0.1
```

4. Maka hasilnya akan tampak seperti Gambar 2.

## Tip-tip keamanan Apache Web Server

Setelah Anda berhasil melakukan instalasi, kompilasi dan menjalankan Apache Web Server, maka ada baiknya jika Anda memperhatikan beberapa faktor keamanan yang akan kita bahas berikut ini. Sisi keamanan ini akan sangat vital, bilamana Apache Web Server yang kita bangun ini, juga diperuntukkan bagi akses umum, tidak hanya untuk akses pribadi.

Berikut ini, beberapa tip sekitar keamanan pada Apache Web Server.



▲ Gambar 2. Apache diakses dari browser galeon

### → Perizinan pada direktori ServerRoot

Program Apache pada dasarnya dijalankan oleh *root*, yang kemudian disalurkan pada masing-masing *user* yang terdaftar atau terdefinisi. Oleh karena itu, jagalah agar jangan sampai user biasa bisa melakukan modifikasi file-file pada Apache.

Misalnya, jika memilih untuk menempatkan Apache pada direktori `/usr/local/apache`, maka pastikan Anda membuat direktori tersebut dengan login sebagai *root*, dengan mengetikkan perintah sebagai berikut:

```
[root@localhost: ~] # mkdir /usr/local/apache
[root@localhost: ~] # cd /usr/local/apache
[root@localhost: ~] # mkdir bin conf logs
[root@localhost: ~] # chown 0 . bin conf logs
[root@localhost: ~] # chgrp 0 . bin conf logs
[root@localhost: ~] # chmod 755 . bin conf logs
```

Dengan menggunakan asumsi, bahwa direktori `/`, `/usr` dan `/usr/local` hanya bisa

dilakukan modifikasi oleh *root*. Pada saat melakukan instalasi program `httpd`, Anda harus memastikan untuk melindungi direktori terkait dengan menggunakan perintah sebagai berikut:

```
[root@localhost: ~] # cp httpd /usr/local/apache/bin
[root@localhost: ~] # chown 0 /usr/local/apache/bin/httpd
[root@localhost: ~] # chgrp 0 /usr/local/apache/bin/httpd
[root@localhost: ~] # chmod 511 /usr/local/apache/bin/httpd
```

Anda juga harus memastikan bahwa direktori yang berisi file log tidak bisa ditulis atau diubah oleh user lain selain *root*, bila tidak maka user biasa akan membuat symlink file ke direktori log dan selanjutnya mereka bisa melakukan perubahan terhadap file-file log tersebut.

### → Server Side Includes (SSI)

*Server Side Includes* mempunyai potensi

risiko keamanan yang tinggi, terutama jika digabungkan dengan modul CGI. Anda bisa menggunakan program `exec cmd`, SSI memungkinkan Anda untuk bisa menjalankan berbagai skrip CGI yang ada pada server.

### → Penggunaan CGI

Dalam penggunaan skrip CGI, Anda harus memastikan bahwa:

1. Anda harus memastikan bahwa user Anda tidak membuat skrip yang bisa merusak sistem.
2. Anda harus memastikan user Anda bisa dipercaya dalam membuat dan menggunakan fasilitas CGI tersebut.

Sampai di sini selesai sudah instalasi dan konfigurasi web server dengan menggunakan program aplikasi Apache. Pada saat PC Anda dihidupkan, Apache secara otomatis akan berjalan dan PC klien bisa melakukan akses ke Web Server Anda.🔗

**R. Kresno Aji** ([masaji@ai.co.id](mailto:masaji@ai.co.id))

# IKLAN