

Apache Web Server di Berbagai Distro

Anda yang berkecimpung di dunia *web server*, pasti pernah mendengar nama Apache, web server standar yang dipaket oleh berbagai macam distribusi Linux. Kemudahan konfigurasi, unjuk kerja yang bagus, dan arsitektur yang modular, menjadikannya sebagai web server yang terkenal.

Web server adalah *software* yang memberikan layanan web. Web server menggunakan protokol yang disebut dengan HTTP (*HyperText Transfer Protocol*). Anda mempunyai banyak pilihan di dunia *open source*, tergantung pada keperluan Anda. Dalam tulisan ini akan kita bahas salah satu web server yang sangat terkenal dan menjadi standar *de facto* setiap distribusi Linux, yaitu Apache.

Apache adalah nama web server yang dibuat berdasarkan kode sumber dan ide-ide yang ada pada web server leluhurnya, yaitu web server NCSA. Sesuai namanya, web server NCSA dibuat oleh National Center for Supercomputing Applications.

Tidak seperti poryek leluhurnya yang dibiayai oleh pemerintah Amerika, web server Apache dikembangkan oleh sekelompok programmer yang bekerja tanpa dibayar oleh siapapun. Mereka mengerjakan proyek ini dengan berbagai macam alasan, akan tetapi alasan yang paling mendasar adalah mereka senang jika perangkat lunak mereka digunakan oleh banyak orang.

Apache adalah web server yang kompak, modular, mengikuti standar protokol HTTP, dan tentu saja sangat digemari. Kesimpulan ini bisa didapatkan dari jumlah pengguna yang jauh melebihi para pesaingnya. Sesuai hasil survei yang dilakukan oleh Netcraft, bulan Januari 2005 saja jumlahnya tidak kurang dari 68% pangsa web server yang berjalan di Internet. Ini berarti jika semua web server selain Apache digabung, masih belum bisa mengalahkan jumlah Apache. Saat ini ada dua versi Apache yang bisa dipakai untuk server produksi, yaitu versi mayor 2.0 dan

versi mayor 1.3. Anda bisa menggunakan salah satu dari keduanya. Tapi sangat dianjurkan Anda memakai versi 2.0.

Mengapa kita harus memilih Apache? Tentu itu pertanyaan yang terfikir dalam benak Anda. Ya, tentu saja kita harus mempunyai alasan untuk segala hal, termasuk memilih Apache sebagai web server pilihan. Mungkin daftar berikut bisa menjadi jawabannya.

- Arsitektur modular.
- Mendukung banyak sistem operasi, termasuk di dalamnya adalah Windows NT/2000/XP dan berbagai varian Unix.
- Mendukung IP versi 6 (IPv6).
- Mendukung CGI (Common Gateway Interface) dan SSI (Server Side Include).
- Mendukung otentifikasi dan kontrol akses.
- Mendukung SSL (Secure Socket Layer) untuk komunikasi terenkripsi.
- Konfigurasi yang mudah dipahami.
- Mendukung Virtual Host.
- Pesan kesalahan multi bahasa dan bisa dimodifikasi.

Instalasi Apache

Proses instalasi dari paket binary, berbeda dalam berbagai distribusi atau distro Linux. Tergantung pada sistem pemaketan yang digunakan. Berikut ini pembahasan cara instalasi binary untuk tiga distro besar dan cara instalasi dari kode sumber. Untuk pembahasan selanjutnya akan kita pakai contoh instalasi dari kode sumber. Tujuannya, agar mudah diikuti dan seragam untuk berbagai variasi distribusi Linux.

Debian Sarge

Debian menggunakan sistem pemaketan

.deb. Sistem pemaketan yang mudah tapi berbasis text ini memungkinkan untuk melakukan proses pendeteksian ketergantungan paket secara otomatis. Paket Apache versi 2.0 terdapat dalam CD kedua dari lima belas CDROM distribusi Debian. Dalam contoh ini dipakai versi *unstable* yang diberi nama kode Sarge.

Untuk memasukkan daftar paket dari CD distro, gunakan perintah berikut ini:

```
# apt-cdrom add
Using CD-ROM mount point /cdrom/
Unmounting CD-ROM
Please insert a Disc in the
drive and press enter
```

Masukkan CD kedua ke dalam CD drive dan tekan tombol [Enter]. Tunggu sebentar, sistem Debian akan membaca informasi semua paket yang terdapat pada CD tersebut.

Setelah proses tersebut selesai, jalankan perintah berikut:

```
# apt-get install apache2
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will
be installed:
  apache2
0 upgraded, 1 newly installed,
0 to remove and 0 not upgraded.
Need to get 0B/31.2kB of
archives.
After unpacking 77.8kB of
additional disk space will be
used.
Media Change: Please insert the
disc labeled
'Debian GNU/Linux testing
```

```
_Sarge_ - Official Snapshot i386
Binary-2 (20041225)'in the drive
'/cdrom/' and press enter
```

Masukkan CD ROM kedua distribusi Debian Sarge, kemudian tekan [Enter]. Setelah selesai jalankan skrip untuk menjalankan server web.

```
# /etc/init.d/apache2 start
```

Fedora Core 3

Fedora Core 3 adalah proyek lanjutan dari Red Hat setelah Red Hat versi 9.0. Yang berarti juga memakai sistem pemaketan perangkat lunak sama dengan Red Hat yaitu RPM (RedHat Package Manager). Untuk menginstal paket Apache, Anda memerlukan CD pertama distribusi Fedora Core 3.

Masukkan CDROM pertama tersebut, aktifkan terminal dan jalankan perintah:

```
# mount /media/cdrom/
# rpm -ivh /media/cdrom/Fedora/
RPMS/httpd-2.0.52-3.i386.rpm
```

Menjalankan server Apache dengan cara menjalankan skrip:

```
# /etc/init.d/httpd start
```

Kemudian akses dengan browser web kesukaan Anda ke <http://localhost>.

Mandrake 10

Seperti Fedora Core 3, Mandrake memakai

sistem manajemen paket RPM. Karena memang Mandrake juga distro turunan dari Red Hat. Perbedaannya pada pemaketannya saja.

Untuk menginstall Apache di Mandrake sangat mudah dengan menjalankan perintah berikut di terminal (masukkan CD yang diminta dan tekan tombol [Enter], maka pengecekan dependensi dan proses instalasi akan dijalankan oleh URPMI):

```
# urpmi apache2
To satisfy dependencies, the
following packages are going to
be installed (1 MB):
apache-conf-2.0.48-2mdk.i586
apache2-2.0.48-6mdk.i586
apache2-common-2.0.48-6mdk.i586
apache2-modules-2.0.48-6mdk.i586
Is this OK? (Y/n) Please insert
the medium named "Mandrakelinux
10.0 Official Download CD 1" on
device [/dev/hdc]
Press Enter when ready...
installing /mnt/cdrom/Mandrake/
RPMS/apache2-modules-2.0.48-
6mdk.i586.rpm /mnt/cdrom/
Mandrake/RPMS/apache-conf-
2.0.48-2mdk.i586.rpm /mnt/
cdrom/Mandrake/RPMS/apache2-
2.0.48-6mdk.i586.rpm /mnt/cdrom/
Mandrake/RPMS/apache2-common-
2.0.48-6mdk.i586.rpm
Preparing... #####
```

```
#####
1:apache2-common #####
#####
2:apache-conf #####
#####
3:apache2-modules #####
#####
4:apache2 #####
#####
```

Untuk menjalankan server gunakan perintah:

```
# /etc/init.d/httpd start
```

Instalasi dari kode sumber

Proses instalasi dari kode sumber memang sedikit lebih merepotkan daripada instalasi dari paket-paket binary semacam rpm atau deb. Akan tetapi instalasi dengan cara ini tidak bergantung dari distro Linux yang Anda pakai. Jadi bisa dipastikan berjalan di semua distribusi.

Sebelum melakukan proses instalasi menggunakan kode sumber sebaiknya Anda periksa sistem Linux Anda apakah sudah terinstal semua perangkat lunak dalam kategori "development". Karena ini akan diperlukan dalam proses kompilasi. Anda tidak perlu takut-takut melakukan kompilasi dari kode sumber. Karena dalam proses tersebut akan dilakukan pengecekan perangkat lunak yang diperlukan.

Bila terjadi kegagalan, perhatikan pesan kesalahan yang terakhir. Penyebab kesalahan yang umum adalah belum terinstalnya paket pengembangan yang diperlukan.

Sebelum melakukan proses kompilasi, dapatkan terlebih dahulu file kode sumber yang diperlukan dari url berikut:

```
http://apache.the.net.id/httpd/
httpd-2.0.52.tar.bz2
```

Jika url tersebut tidak bisa diakses, Anda bisa mencari mirror-mirror lain dari situs resmi Apache. Salin file tersebut ke direktori `/usr/local/src/`.

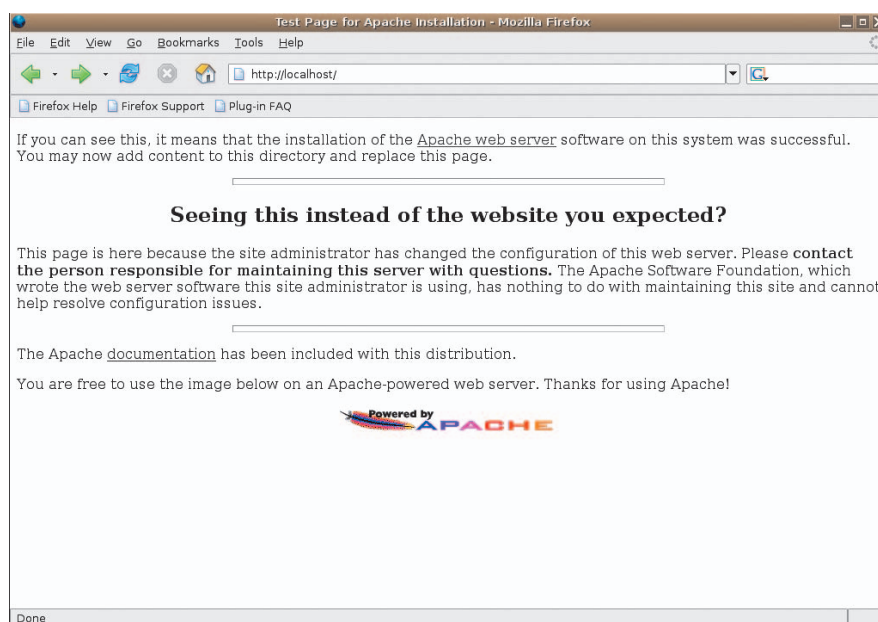
```
# cp httpd-2.0.52.tar.bz2 /usr/
local/src/
```

Setelah itu ikuti langkah-langkah berikut:

1. Uraikan file terkompresi di direktori `/usr/local/src/`

```
# cd /usr/local/src/
```

```
# tar -xjvf httpd-2.0.52.tar.
bz2
```



Gambar 1. Test Page web server Apache.

2. Konfigurasi kode sumber sesuai dengan lingkungan pengembangan di komputer tempat Anda bekerja.

```
# cd httpd-2.0.52/
# ./configure --prefix=/usr/
  local/apache --enable-so
```

Dengan konfigurasi ini root direktori httpd server akan terletak pada direktori `/usr/local/apache/`. Pilihan `--enable-so`, mengaktifkan dukungan terhadap modul *shared object*. Tujuannya agar nanti Anda bisa menambahkan modul-modul DSO (*Dynamic Shared Object*). DSO adalah modul-modul Apache yang diletakkan dalam file-file terpisah, tidak dikompilasi ke dalam Apache web server itu sendiri. Modul-modul tersebut bisa dimuat ketika Apache dijalankan.

Yakinkan dalam proses ini tidak ada masalah sebelum Anda melanjutkan ke tahap selanjutnya. Dan kegagalan yang sangat mungkin terjadi adalah di tahap ini.

3. Jalankan proses kompilasi dan instalasi.

```
# make
# make install
```

Perintah `make` akan mengompilasi semua kode sumber, kemudian perintah `make install` akan menyalin program-program hasil kompilasi ke direktori masing-masing sesuai tempat yang telah ditentukan pada proses konfigurasi.

4. Salin skrip startup ke direktori startup.

```
# cp support/apachectl /etc/
  init.d/
# chmod 755 /etc/init.d/
  apachectl
```

Jika direktori `/etc/init.d/` tidak ada dalam distribusi Linux Anda, lewati saja langkah ini.

5. Jalankan server web Apache dengan perintah berikut:

```
# /etc/init.d/apachectl start
```

Jika distro yang Anda pakai tidak memiliki direktori `/etc/init.d/` gunakan perintah ini:

```
# /usr/local/apache/bin/
  apachectl start
```

Kemudian coba akses dengan browser alamat `http://localhost`. Mestinya Anda su-

dah bisa melihat halaman “Test Page for Apache” seperti ditunjukkan pada gambar 1. Tambahkan perintah ini ke dalam file `/etc/rc.local` agar setiap kali sistem booting, server web akan dijalankan.

Konfigurasi Apache

Apache mempunyai direktori-direktori khusus. Masing-masing digunakan untuk keperluan yang berbeda, yaitu:

- `conf/` untuk menyimpan file-file konfigurasi.
- `logs/` untuk menyimpan file-file log.
- `htdocs/` untuk menyimpan dokumen-dokumen HTML dan yang lainnya yang akan diakses oleh pengunjung situs Anda.
- `modules/` berisi module-module yang akan dimuat ke memory oleh Apache ketika diperlukan.
- `cgi-bin/` berisi skrip-skrip CGI.

Jika Instalasi dilakukan dari paket binary, letak masing-masing direktori ini tergantung pada kebijakan dari masing-masing distribusi Linux. Sebagai contoh letak direktori konfigurasi dan namanya ditunjukkan pada tabel 1.

Selanjutnya kita akan membahas konfigurasi Apache berdasarkan instalasi dari kode sumber.

Konfigurasi web server untuk domain

Anda mempunyai sebuah *domain*, katakan `domainku.lab`. Dan Anda ingin membuat sebuah situs web dengan nama `www.domainku.lab`. Sengaja diambil contoh domain yang tidak sesuai dengan top level domain, agar tidak mengacaukan jaringan yang sudah terstruktur dengan baik. Tentu saja untuk membuat web server yang bisa diakses oleh orang di seluruh dunia melalui Internet, Anda harus mendaftarkan domain di perusahaan-perusahaan pendaftaran nama domain. Dan tentu saja harus sudah dipersiapkan DNS server. Bagaimana cara mendaftarkan domain dan mempersiapkan DNS server tidak dibahas dalam tulisan ini.

Sistem operasi Linux akan menerjemahkan sebuah nama menjadi nomor IP, paling tidak bisa melalui dua cara, yaitu:

- File `/etc/hosts` (`hosts`).
- DNS server (`bind`).

Urutannya diatur dalam file konfigurasi `/etc/host.conf`. Contoh isi file ini adalah:

```
order hosts,bind
multi on
```

artinya komputer akan melakukan translasi nama ke nomor IP melalui file `/etc/hosts` terlebih dahulu. Jika gagal maka akan dicari dari DNS server. Tentu saja akan timbul pertanyaan, dari mana komputer mendapatkan informasi? Informasi ini didapatkan dari file `/etc/resolv.conf` atau resolver yang contoh isinya seperti berikut:

```
nameserver 202.159.32.2
nameserver 192.168.1.1
```

Komputer akan mengirimkan permintaan translasi pertama kali pada server dengan nomor IP 202.159.32.2, jika gagal akan mengulang permintaan ke server dengan nomor IP 192.168.1.1.

Dalam kasus kita kali ini, kita cukup memasukkan nama web server kita dalam file `/etc/hosts`. Tambahkan baris berikut dalam file tersebut.

```
192.168.1.192 www.domainku.lab
```

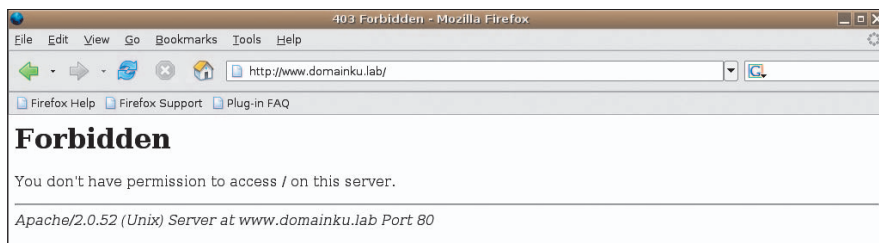
➔ Misalkan nomor IP komputer kita adalah 192.168.1.192.

Terlebih dahulu cek apakah nama ini bisa diterjemahkan ke nomor IP yang benar dengan perintah `ping`.

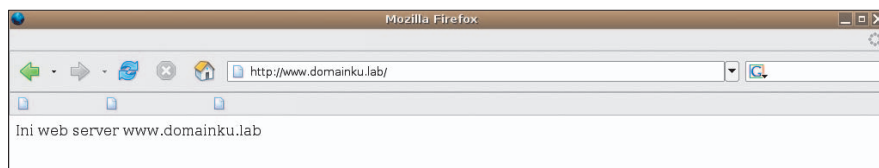
```
# ping -c3 www.domainku.lab
PING www.domainku.lab (192.168.1.192) 56(84) bytes of data.
64 bytes from www.domainku.lab (192.168.1.192): icmp_seq=1
ttl=64 time=0.052 ms
64 bytes from www.domainku.lab (192.168.1.192): icmp_seq=2
ttl=64 time=0.048 ms
64 bytes from www.domainku.lab (192.168.1.192): icmp_seq=3
ttl=64 time=0.059 ms
--- www.domainku.lab ping
```

Table 1. Letak konfigurasi web server Apache.

Distribusi	Letak File Konfigurasi
Debian Sarge	<code>/etc/apache2/</code>
Fedora	<code>/etc/httpd/</code>
Mandrake	<code>/etc/httpd/</code>
Kompilasi dari kode sumber (.tar.gz)	<code>/usr/local/apache/conf/</code>



Gambar 2. Pesan larangan akses dokumen web.



Gambar 3. Hasil tampilan web browser setelah web server selesai dikonfigurasi.

```
statistics ---
3 packets transmitted,
3 received, 0% packet loss,
time 1999ms
rtt min/avg/max/mdev =
0.048/0.053/0.059/0.004 ms
```

Selesai. Sekarang kita harus siapkan web servernya.

Edit file konfigurasi Apache dengan editor kesukaan Anda, misalnya vi.

```
# vi /usr/local/apache/conf/
httpd.conf
```

Semua karakter yang diawali dengan tanda sharp (#) dianggap komentar saja sampai akhir baris. Untuk mempermudah proses pengeditan gunakan fasilitas search editor vi pada mode perintah. Caranya adalah dengan mengetikkan karakter slash (/) diikuti kata kunci yang ingin dicari. Misalnya, /ServerName.

Untuk menentukan nama host web server, gunakan direktif ServerName. Seperti contoh ini,

```
ServerName www.domainku.lab
```

Ganti informasi administrator web server dengan direktif ServerAdmin. Ingat, gunakan alamat e-mail yang benar, agar jika terjadi masalah dengan web server Anda, pengunjung akan tahu siapa yang harus dihubungi.

Berikan tanda komentar pada baris lain yang mengandung kata kunci ini.

```
ServerAdmin webmaster@domainku.
lab
```

Tentukan juga direktori utama yang dipetakan dengan nama host kita dengan di-

rektif DocumentRoot.

```
DocumentRoot "/home/httpd/html"
```

Ketika pengunjung menuliskan nama host web server, maka yang diberikan adalah dokumen yang berada di direktori ini. Jangan lupa membuat direktori ini.

```
# mkdir -p /home/httpd/html
```

Restart web server dengan menjalankan perintah berikut.

```
# /usr/local/apache/bin/
apachectl restart
```

Sebelum mencoba mengakses web server, kita sediakan terlebih dahulu dokumen html sederhana pada direktori /home/httpd/html.

```
# echo "Ini web server www.
domainku.lab" >> /home/httpd/
html/index.html
```

Dan coba akses dengan browser dengan mengetikkan nama web server kita, yaitu *www.domainku.lab*. Jika browser menampilkan seperti gambar 2, Anda tidak perlu panik.

Pesan larangan akses ini disebabkan karena direktori yang disebutkan dalam direktif DocumentRoot tidak bisa diakses oleh web server. Sedangkan web server secara default dijalankan oleh user nobody.

Perhatikan hasil perintah ps berikut.

```
# ps aux|grep httpd
root      30521  0.0  0.2  1820
560 pts/1    S+   19:06   0:00
\_ grep httpd
root      21317  0.0  0.6  3616
```

```
1792 ?        Ss   16:15   0:00
/usr/local/apache/bin/httpd -k
start
nobody    29977  0.0  0.7  3616
1904 ?        S    18:56   0:00
\_ /usr/local/apache/bin/httpd
-k start
nobody    29978  0.0  0.7  3616
1816 ?        S    18:56   0:00
\_ /usr/local/apache/bin/httpd
-k start
nobody    29979  0.0  0.7  3616
1816 ?        S    18:56   0:00
\_ /usr/local/apache/bin/httpd
-k start
nobody    29980  0.0  0.7  3616
1816 ?        S    18:56   0:00
\_ /usr/local/apache/bin/httpd
-k start
nobody    29981  0.0  0.7  3616
1816 ?        S    18:56   0:00
\_ /usr/local/apache/bin/httpd
-k start
nobody    30026  0.0  0.7  3616
1816 ?        S    18:56   0:00
\_ /usr/local/apache/bin/httpd
-k start
```

Agar web server bisa mengakses direktori tersebut, ada dua cara:

- Mengubah kepemilikan direktori /home/httpd/html menjadi milik user nobody,
- Mengubah hak akses direktori tersebut agar bisa diakses oleh user nobody.

Gunakan perintah chown untuk mengubah kepemilikan direktori.

```
# chown -R nobody /home/httpd/
html/
```

Untuk mengubah hak akses direktori gunakan perintah chmod.

```
# chmod 755 /home/httpd/
# chmod 755 /home/httpd/html/
```

Gunakan salah satu saja dari alternatif tersebut, kemudian akses lagi web server dengan browser. Seharusnya browser akan menampilkan hasil seperti ditunjukkan pada gambar 3.

Selesai. Tugas selanjutnya bagi Anda adalah membangun situs Anda sendiri dengan format HTML. Selamat bekerja!

Salman AS (*sas@salman.or.id*)