Open Source Campus Agreement

Modul Pelatihan

# ADMINISTRASI JARINGAN LINUX

oleh :

R. Anton Raharja

Afri Yunianto

Wisesa Widyantoro

Editor:

I Made Wiryana



2001

# Administrasi jaringan Linux

R. Anton Raharja <**anton@ngoprek.org**>
Afri Yunianto <**afri@ngoprek.org**>
Wisesa Widyantoro <**pondokbambu@yahoo.com**>

Editor : I Made Wiryana **<mwiryana@rvs.uni-bielefeld.de**>

2001

# Daftar Isi

# Daftar Gambar

# Daftar Tabel

# Kata Pengantar

Rasa syukur yang sangat mendalam, kami panjatkan kehadirat Allah SWT, sehingga melalui rahmat-Nya yang tiada terkira rilis pertama dari modul Linux Basic ini dapat terselesaikan.

Pada mulanya kami menggunakan modul ini dalam rangka pelatihan Linux yang diadakan di Telematics Indonesia. Seluruh rangkaian modul yang tersedia ada 3 versi, Basic, System Administrator dan Network Administrator. Modul ini di release menggunakan lisensi **OPL (Open Public License)**, yang berarti siapapun, dengan tujuan apapun, boleh dan secara legal dapat membuat salinan, dapat memperbanyak, dan dapat mendistribusikannya kembali ke masyarakat.

Kami sadar dengan banyaknya keterbatasan yang kami miliki, modul ini jauh dari sempurna. Masih butuh sentuhan tangan-tangan yang lebih expert dalam mengembangkannya. Kami mengharapkan input dari semua masyarakat, terutama dari komunitas Linux di Indonesia, karena modul ini adalah sebagai sedikit sumbangsih kami untuk komunitas.

## Penyusun

- R Anton Raharja
- Afri Yunianto
- Wisesa Widyantoro

# Tentang penulis

**Anton Raharja**, seorang anak muda. Perkenalannya dengan Linux menjadikan perubahan yang drastis pada dirinya sehingga kini menjadi aktif mengutak-atik sistem dan melakukan kegiatan dengan giat tanpa kenal lelah. Siang malam dihabiskan untuk melakukan pekerjaan mengoprek mesin-mesin komputer, baik milik teman ataupun milik sendiri. Dapat dikontak dengan email : anton@ngoprek.org

**Afri Yunanto**, seorang anak muda. Perkenalannya dengan Linux menjadikan perubahan yang drastis pada dirinya sehingga kini menjadi aktif mengutak-atik sistem dan melakukan kegiatan dengan giat tanpa kenal lelah. Siang malam dihabiskan untuk melakukan pekerjaan mengoprek mesin-mesin komputer, baik milik teman ataupun milik sendiri. Dapat dikontak dengan email : afri@ngoprek.org

**Wisesa Widyantoro**, seorang anak muda. Perkenalannya dengan Linux menjadikan perubahan yang drastis pada dirinya sehingga kini menjadi aktif mengutak-atik sistem dan melakukan kegiatan dengan giat tanpa kenal lelah. Siang malam dihabiskan untuk melakukan pekerjaan mengoprek mesin-mesin komputer, baik milik teman ataupun milik sendiri. Dapat dikontak dengan email : pondokbambu@yahoo.com

**I Made Wiryana SSi SKom MSc** menamatkan S1 di jurusan Fisika FMIPA Universitas Indonesia pada bidang instrumentasi dan fisika terapan. Dengan beasiswa dari STMIK Gunadarma juga menamatkan S1 Teknik Informatika di STMIK Gunadarma. Melanjutkan studi S2 di Computer Science Department Edith Cowan University - Perh dengan beasiswa ADCSS dan STMIK Gunadarma pada bidang fuzzy system dan artificial neural network untuk pengolahan suara. Menangani perancangan dan implementasi jaringan Internet di Universitas Gunadarma dengan memanfaatkan sistem Open Source sebagai basisnya. Pernah mewakili IPKIN dalam kelompok kerja Standardisasi Profesi TI untuk Asia Pasifik (SEARCC). Saat ini dengan beasiswa dari DAAD melanjutkan studi doktoral di RVS Arbeitsgruppe Universität Bielefeld Jerman di bawah bimbingan Prof. Peter B Ladkin PhD. Aktif menjadi koordinator beberapa proyek penterjemahan program Open Source seperti KDE, SuSE, Abiword dan juga sebagai advisor pada Trustix Merdeka, distibusi Linux Indonesia yang pertama. Terkadang menyumbangkan tulisannya sebagai kolumnis pada media online DETIK.COM dan SATUNET. Juga kontributor pada KOMPUTEK, MIKRODATA, ELEKTRO dan INFOLINUX. Kontribusi ke komunitas Open Source sering dilakukan bersama-sama kelompok Tim PANDU. Star pengajar tetap Universitas Gunadarma.

# Pernyataan

Beberapa merk dagang yang disebutkan pada buku ini merupakan merk dagang terdaftar dari perusahaan tersebut, kecuali bila disebutkan lain.

Pembuatan modul ini disponsori oleh :

**TELEMATICS INDONESIA**

Jl. Adhyaksa Raya No.11

Bandung 40267

Homepage : `http://www.telematicsindonesia.com`

Email : `support@telematicsindonesia.com`

Proses pengeditan dan pemformatan dilakukan editor yang secara tidak langsung disponsori oleh :

- **Deutscher Akademischer Austauschdiesnt (DAAD**)
  Homepage : `http://www.daad.de`
- **Universitas Gunadarma**
  Homepage : `http://www.gunadarma.ac.id`

Beberapa merk dagang yang disebutkan pada buku ini merupakan merk dagang terdaftar dari perusahaan tersebut, kecuali bila disebutkan lain.

# Bab 1

# Pengenalan jaringan

Network atau jaringan, dalam bidang komputer dapat diartikan sebagai dua atau lebih komputer yang dihubungkan sehingga dapat berhubungan dan dapat berkomunikasi, sehingga akan menimbulkan suatu effisiensi, sentralisasi dan optimasi kerja. Pada jaringan komputer yang dikomunikasikan adalah data, satu komputer dapat berhubungan dengan komputer lain dan saling berkomunikasi (salah satunya bertukar data ) tanpa harus membawa disket ke satu komputer ke komputer lainnya seperti yang biasa kita lakukan.

Ada beberapa jenis jaringan komputer dilihat dari cara pemrosesan data dan pengaksesannya.

- **Host-Terminal.** Dimana terdapat sebuah atau lebih server yang dihubungkan dalam suatu dumb terminal. Karena Dumb Terminal hanyalah sebuah monitor yang dihubungkan dengan menggunakan kabel RS-232, maka pemrosesan data dilakukan di dalam server, oleh karena itu maka suatu server haruslah sebuah sistem komputer yang memiliki kemampuan pemrosesan data yang tinggi dan penyimpanan data yang sangat besar.
- **Client - Server**. Dimana sebuah server atau lebih yang dihubungkan dengan beberapa client. Server bertugas menyediakan layanan, bermacam-macam jenis layanan yang dapat diberikan oleh server, misalnya adalah pengaksesan berkas, peripheral,database, dan lain sebagainya. Sedangkan client adalah sebuah terminal yang menggunakan layanan tersebut. Perbedaannya dengan hubungan dumb terminal, sebuah terminal client melakukan pemrosesan data di terminalnya sendiri dan hal itu menyebabkan spesifikasi dari server tidaklah harus memiliki performansi yang tinggi, dan kapasitas penyimpanan data yang besar karena semua pemrosesan data yang merupakan permintaan dari client dilakukan di terminal client.
- **Peer to Peer**. Dimana terdapat beberapa terminal komputer yang dihubungkan dengan media kabel. Secara prinsip, hubungan peer to peer ini adalah bahwa setiap komputer dapat berfungsi serbagai server (penyedia layanan) dan client, keduanya dapat difungsikan dalam suatu waktu yang bersamaan.

Sedangkan apabila kita lihat dari sisi lingkupannya atau jangkauannya, jaringan dapat di bagi menjadi beberapa jenis, yaitu :

- **LAN** (**Local Area Network**). Hanya terdapat satu atau dua server dan ruang lingkupnya hanya terdapat dalam satu lokasi atau gedung, Hal ini akan mendapat pembahasn tersendiri pada sub bahasan berikutnya.
- **WAN** (**Wide Area Network**). Merupakan gabungan dari LAN, yang ruang lingkupnya dapat saja satu lokasi, misalnya gedung bertingkat, atau dapat tersebar di beberapa lokasi di seluruh dunia, jaringan jenis ini membutuhkan minimal satu server untuk setiap LAN, dan membutuhkan minimal dua server yang mempunyai lokasi yang berbeda untuk membentuknya.
- **Internet.** Internet adalah sekumpulan jaringan yang berlokasi tersebar di seluruh dunia yang saling terhubung membentuk satu jaringan besar komputer. Dalam jaringan ini dibatasi layanannya sebagai berikut : FTP, E-Mail, Chat, Telnet, Conference, News Group, Mailing List. Biasanya jaringan ini menggunakan protoko; TCP/IP (Jenis protokol ini akan dibahas selanjutnya), walaupun ada sebagian kecil yang menggunakan jenis lain (IPX Novell Netware, NetBios, dan lain-lainnya)
- **Intranet**. Jenis jaringan ini merupakan gabungan dari LAN/WAN dengan Internet. Apabila kita lihat dari lingkupannya atau jangkauannya maka jaringan ini adalah jenis LAN/WAN yang memberikan layanan seper-

ti layanan internet kepada terminal clientnya. Perbedaan menyolok Intranet dengan Internet adalah Intranet melayani satu organisasi tertentu saja.

Dari jenis-jenis jaringan yang telah dijelaskan diatas, yang akan dijelaskan dalam pelatihan ini adalah jenis LAN (Local Area Network), karena LAN merupakan jaringan terkecil dan yang paling penting, karena jenis-jenis jaringan yang lain hanya merupakan pengembangan dari LAN saja.

## 1.1 Topologi jaringan

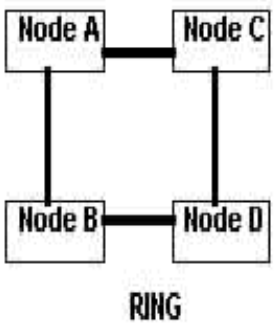
Apabila dilihat dari jenis hubungannya, maka topologi jaringan dapat dibagi menjadi tiga, yaitu :

- Topologi cincin (*ring topology*)
- Topologi bus (*bus topology*)
- Topologi bintang (*star topology*)

Berikut adalah ilustrasi dari ketiga topologi di atas :

### 1.1.1 Topologi cincin (*ring topology*)

Topologi jenis cincin ini menghubungkan satu komputer di dalam suatu loop tertutup. Pada topologi jenis ini data atau message berjalan mengelilingi jaringan dengan satu arah pengiriman ke komputer selanjutnya terus hingga mencapai komputer yang dituju. Waktu yang di butuhkan untuk mencapai terminal tujuan disebut *walk time* (waktu transmisi).




Gambar 1.1: Topologi jaringan tipe cincin

Ada dua hal yang dilakukan oleh suatu terminal ketika menerima data dari komputer sebelumnya, yaitu :

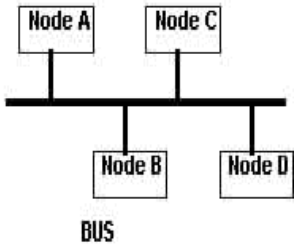
1. Memeriksa alamat yang dituju dari data tersebut dan menerimanya jika terminal ini merupakan tujuan data tersebut.
2. Terminal akan meneruskan data ke komputer selanjutnya dengan memberikan tanda negatif ke komputer pengirim.

Apabila ada komputer yang tidak berfungsi maka hal tersebut tidak akan mengganggu jalannya jaringan, tapi apabila satu kabel putus akan mengakibatkan jaringan tidak berfungsi.

### 1.1.2 Topology bus (*bus topology*)

Topologi jaringan jenis ini menggunakan sebuah kabel pusat yang merupakan media utama dari jaringan. Terminal-terminal yang akan membangun jaringan dihubungkan dengan kabel utama yang merupakan inti dari jaringan.

Data yang dikirimkan akan langsung menuju terminal yang dituju tanpa harus melewati terminal-terminal dalam jaringan, atau akan di routingkan ke *head end controller.* Tidak bekerjanya sebuah komputer tidak akan menghentikan kerja dari jaringan, jaringan akan tak bekerja apabila kabel utamanya dipotong atau putus.

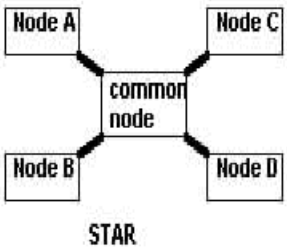



Gambar 1.2: Topologi jaringan tipe bus

Jaringan ini merupakan jaringan yang banyak digunakan karena hanya dalam beberapa meter kabel dapat dihubungkan ke banyak terminal client. Jaringan ini biasanya menggunakan kabel coaxial sebagai media transmisinya. Kabel coaxial dilihat adri bentuk fisiknya mirip dengan kabel antena. Kabel ini mempunyai kapasitas bandwidth yang besar (2MB), sehingga apabila dihubungkan dengan banyak terminal akan terlayani dengan baik.

### 1.1.3 Topologi bintang (*star topology*)

Jenis topologi jaringan ini menggunakan satu terminal sebagai terminal sentral yang mengubungkan ke semua terminal client. Terminal sentral ini yang mengarahkan setiap data yang dikirimkan ke komputer yang dituju. Jenis jaringan ini apabila ada salah satu terminal client tidak berfungsi atau media transmisi putus atau terganggu makan tidak akan mempengaruhi kerja dari jaringan, karena gangguan tersebut hanya mempengaruhi terminal yang bersangkutan.




Gambar 1.3: Topologi star (bintang)

Kelemahan dari jenis topologi jaringan ini adalah ketergantungan terhadap suatu terminal sentral. Hal tersebut merupakan suatu gangguan yang sangat berarti apabila terminal sentral tersebut mendapatkan gangguan, sehingga dicari suatu solusi yang dapat mengatasi masalah tersebut. Salah satu solusi yang banyak dilakukan adalah dengan menggunakan dua buah terminal sebagai server, sehingga apabila satu server dalam keadaan down dapat dialihkan ke server yang kedua dan begitu seterusnya.

## 1.2 TCP/IP (Transfer Control Protocol/Internet Protocol)

TCP/IP terdiri dari lapisan-lapisan protokol. Untuk memudahkan dalam memahaminya maka akan diambil contoh pengiriman email. Dalam pengiriman email yang diperlukan adalah protokol untuk email. Protokol ini mendefinisikan perintah-perintah yang diperlukan dalam pengiriman email, dan protokol ini juga mengasumsikan bahwa ada hubungan antara terminal yang mengirim dengan terminal yang dituju. Dalam hal ini perintah-perintah tersebut diatur oleh TCP dan IP. TCP mengatur masalah perintah-perintah pengiriman data, mengawasi jalannya data dan memastikan data tersebut sampai ke tujuannya, apabila ada bagian dari data yang tidak mencapai tujuan maka TCP akan mengirimkan ulang. Proses tersebut terus berlangsung sampai data yang dikirimkan sampai ke tujuannya. Apabila ada data yang sangat besar untuk dimuat dalam satu datagram maka TCP akan memecahnya menjadi beberapa datagram dan kemudian mengirimkan ke tujuan dan memastikan sampai dengan benar. TCP dapat dianggap sebagai suatu pembentuk kumpulan - kumpulan routine (perintah) yang dibutuhkan oleh aplikasi untuk dapat berhubungan dengan terminal lain dalam jaringan.

Tidak semua perintah yang dibutuhkan oleh aplikasi terdapat dalam TCP/IP. IP adalah protokol yang memuat semua kebutuhan aplikasi dalam berhubungan antar terminal. Seperti telah disampaikan sebelumnya bahwa TCP bertanggungjawab di masalah pengiriman dan dalam memecah data menjadi bagian-bagian kecil, maka IP merupakan pembuka jalan hingga sampainya data ke terminal tujuan. Pelapisan-pelapisan protokol tersebut berguna untuk menjaga agar data dapat sampai dengan sempurna.

Beberapa layanan dasar tapi merupakan layanan yang penting diberikan oleh TCP/IP adalah :

- File Transfer (FTP)
- Remote Login (menggunakan fasilitas TELNET)
- Mail elektronik

Sebenarnya masih banyak lagi layanan yang dapat diberikan oleh TCP/IP, hanya tidak akan kita bahas sekarang.

## 1.3 LAN (Local Area Network)

Local Area Network merupakan salah satu arsitektur jaringan yang paling sederhana dan dapat dikembangkan menjadi arsitektur jaringan yang lebih luas cakupannya. Luas cakupan LAN itu sendiri tidak melebihi dari satu area yang terdiri dari beberapa terminal yang saling dihubungkan sehingga menambahkan fungsi dari terminal itu sendiri Layanan-layanan yang dapat diberikan LAN adalah penggunaan file bersama (*file sharing)* atau penggunaan printer bersama, *(printer sharing*).

Biasanya LAN menggunakan satu server untuk melayani kebutuhan clientnya, tetapi tidak menutup kemungkinan untuk menggunakan >1 server, tergantung kebutuhan dari client itu sendiri. Biasanya yang menjadi pertimbangan adalah jenis layanan yang dibutuhkan dan performansi jaringan itu sendiri. Apabila jenis layanan yang dibutuhkan banyak (mail, web, ftp server), maka sebaiknya server yang digunakan lebih dari satu dan hal tersebut akan mempengaruhi kinerja jaringan yang menggunakan layanan-layanan terserbut.

Penamaan terminal dalam suatu jaringan menggunakan apa yang disebut IP Address (Internet Protocol Address). Sedang penamaan penamaan server berdasarkan nama domainnya disebut DNS (Domain Name Server). Kedua cara penamaan ini merupakan cara penamaan yang biasa digunakan dalam jaringan. Hal-hal lebih lanjut akan kita bahas langsung pada pengaplikasian instalasi jaringan pada bahasan selanjutnya.

### 1.3.1 Penamaan alamat IP

IP Address digunakan untuk mengidentifikasi interface jaringan pada host dari suatu mesin. IP Address adalah sekelompok bilangan biner 32 bit yang di bagi menjadi 4 bagian yang masing-masing bagian itu terdiri dari 8 bit (sering disebut IPV4). Untuk memudahkan kita dalam membaca dan mengingat suatu alamat IP, maka umumnya penamaan yang digunakan adalah berdasarkan bilangan desimal.

Misal :

```
11000000.10101000.00001010.00000001
 192    . 168    .  10    .   1
```

### 1.3.2 Pembagian kelas IP

Alamat IP dibagi menjadi kelas-kelas yang masing-masing mempunyai kapasitas jumlah IP yang berbeda-beda. Pada Tabel 1.1 ditampilkan kelas-kelas pengalamatan IP. Pada tabel tersebut `x` adalah **NetID** dan `y` adalah **HostID**

| Kelas | Format | Kisaran | Jumlah IP |
|---|---|---|---|
| A | `0xxxxxxx.yyyyyyyy.yyyyyyyy.yyyyyyy` | `0.0.0.0 - 127.255.255.255` | 16.777.214 |
| B | `10xxxxxx.yyyyyyyy.yyyyyyyy.yyyyyyy` | `128.0.0.0 - 191.255.255.255` | 65.532 |
| C | `110xxxxx.yyyyyyyy.yyyyyyyy.yyyyyyy` | `192.0.0.0 - 223.255.255` | 254 |

Tabel 1.1: Pembagian kelas IP

### 1.3.3 Subnetting

Subnetting adalah pembagian suatu kelompok alamat IP menjadi bagian-bagian yang lebih kecil lagi. Tujuan dalam melakukan subnetting ini adalah :

- Membagi suatu kelas jaringan menjadi bagian-bagian yang lebih kecil.
- Menempatkan suatu host, apakah berada dalam satu jaringan atau tidak.
- Keteraturan
  - Kelas A subnet : `11111111.0000000.00000000.00000000` (`255.0.0.0`)
  - Kelas B subnet : `11111111.11111111.00000000.00000000` (`255.255.0.0`)
  - Kelas C subnet : `11111111.11111111.11111111.00000000` (`255.255.255.0`)

Misal suatu jaringan dengan IP jaringan `192.168.10.0` ingin membagi menjadi 5 jaringan kecil (masing-masing 48 host), yang artinya harus dilakukan proses subnetting dalam jaringan tersebut. Langkah pertama yang harus kita lakukan adalah membagi IP jaringan tersebut (`192.168.10.0` <- kelas C) menjadi blok-blok yang masing-masing blok minimal terdiri dari 48 host. Seperti kita telah ketahui bahwa tiap-tiap kelas C mempunyai 255 IP maka perhitungannya adalah sebagai berikut :

```
255/5 = 51
```

Masing-masing subnet mempunyai 49 alamat IP (masing-masing diambil 2 untuk IP broadcast dan IP network). Berikut adalah pengelompokan dari jaringan-jaringan tersebut :

- `192.168.10.0` - `192.168.10.50` digunakan oleh jaringan 1
- `192.168.10.51` - `192.168.10.101` digunakan oleh jaringan 2
- `192.168.10.102` - `192.168.10.152` digunakan oleh jaringan 3
- `192.168.10.153` - `192.168.10.203` digunakan oleh jaringan 4
- `192.168.10.204` - `192.168.10.224` digunakan oleh jaringan 5

Subneting diperlukan agar host pada satu jaringan tidak dapat mengakses host pada jaringan lain secara langsung. Untuk pembagian 51 host : 51 = `00110011` (biner). Nilai 8 bit tertinggi dari subnetting kelas C adalah : 255 = `11111111`

```
00110011
-------------  (negasi)
11001100 (8 bit terakhir dari subnet kelas C) = 204
```

maka IP subnetmask nya : `255.255.255.204`

## 1.4 Instalasi perangkat jaringan

Pertama-tama kita harus terlebih dahulu mengetahui jenis ethetnet card kita agar mempermudah kita dalam memilih modul apa yang akan digunakan. Misalnya jenis NE2000 compatible, dan juga harus kita ketahui pula variabel-variabel pendukungnya (`io` dan `irq`). Langkah-langkah yang harus dilakukan adalah :

### 1.4.1 Inisialisasi module

```
#modprobe ne io=0x300
```

Bila kartu ethernet yang kita gunakan adalah jenis `ne2000` dengan `io=0x300`

```
#modprobe ne2k-pci io=0x300 irq=5
```

Bila kartu ethernet yang kita gunakan adalah jenis `ne2000` PCI dengan `io=0x300` dan irq=5

### 1.4.2 Menentukan alamat IP ethernet card

Misal kita tentukan IP Address Ethernet Card kita adalah `192.168.0.11`

```
#/sbin/ifconfig eth0 192.168.0.11
```

### 1.4.3 Memeriksa konfigurasi ethernet card

Kita jalankan lagi

```
#/sbin/ifconfig -a
```

Apabila keluar output seperti di bawah :

```
eth0      Link encap:Ethernet  HWaddr 00:00:1C:07:01:22
inet addr:192.168.0.11  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:869146 errors:0 dropped:0 overruns:0 frame:1104
          TX packets:871799 errors:0 dropped:0 overruns:0 carrier:0
          collisions:44040 txqueuelen:100
          Interrupt:5 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:109480 errors:0 dropped:0 overruns:0 frame:0
          TX packets:109480 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Maka berarti ethernet card telah terkonfigurasi dengan baik. Sekarang test koneksi dengan terminal lain, untuk ini kita gunakan perintah ping

```
#ping 192.168.0.12
```

Maka bila tidak ada masalah akan ditampilkan output seperti berikut ini :

```
PING 192.168.0.12 (192.168.0.12) from 192.168.0.11 : 56(84) bytes of data.
64 bytes from venus.planet.tzo.com (192.168.0.12): icmp_seq=0 ttl=128 time=1.5 ms
64 bytes from venus.planet.tzo.com (192.168.0.12): icmp_seq=1 ttl=128 time=0.8 ms
64 bytes from venus.planet.tzo.com (192.168.0.12): icmp_seq=2 ttl=128 time=0.8 ms
64 bytes from venus.planet.tzo.com (192.168.0.12): icmp_seq=3 ttl=128 time=0.8 ms
64 bytes from venus.planet.tzo.com (192.168.0.12): icmp_seq=4 ttl=128 time=0.8 ms

--- 192.168.0.12 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.8/0.9/1.5 ms
```

## 1.5 Koneksi Internet

Disini akan di bahas koneksi ke Internet secara dial up menggunakan PPP. Kita akan menggunakan `wvdial` untuk mempermudah dalam penggunaan `pppd`

### 1.5.1 Instalasi PPP & wvdial

Pertama-tama install pppd terlebih dahulu,

```
#rpm -ivh pppd-x.rpm
```

Lalu install program pembantu, misalkan `wvdial`,

```
#rpm -ivh wvdial-x.rpm
```

### 1.5.2 Konfigurasi client

Konfigurasi `wvdial` menggunakan `wvdialconf`

```
#wvdialconf  /path/to/wvdial.conf
```

Pertama-tama `wvdialconf` akan menanyakan di serial prot mana modem yang digunakan terpasang, berikut adalah daftar serial port yang biasa di pakai :

- `/dev/ttyS0` atau COM 1 di DOS
- `/dev/ttyS1` atau COM 2 di DOS dan seterusnya

setelah kita tentukan maka program `wvdialconf` akan mengetes port serial. Selanjutnya akan di tanyakan nomor ISP yang akan di gunakan. Kemudian setelah itu `wvdialconf` akan meminta kita memasukkan login dan password account kita di ISP. Cara lain adalah dengan mengedit file `wvdial.conf` yang biasanya terletak di `/etc`. Berikut adalah isi dari `wvdial.conf` :

```
#----file wvdial.conf begin :

[Dialer Defaults]
phone = 112233
username = username
password = password
New PPPD = yes
Modem = /dev/ttyS0
Baud = 115200
Init = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
Init1 = ATZ

[Dialer ISP2]
phone = 223344
username = username-isp2
password = pass-isp2

[Dialer ISP3]
phone = 334455
username = username-isp3
password = pass-isp3

#--------------------EOF
```

Biasanya untuk menginisialisasi modem untuk pertama kali kita menggunakan `wvdialconf` terlebih dahulu, kemudian untuk menambahkan account dari ISP lain dapat dilakukan dengan meng-edit file `wvdial.conf`

### 1.5.3 Menjalankan program wvdial

Urut-urutan jalannya program adalah sebagai berikut, `wvdial` adalah program yang menggunakan fasilitas `pppd` untuk menghubungkan suatu host ke ISP, mungkin dapat dikatakan `wvdial` sebagai perantara proses jabat tangan (*handshaking*) antara dial in server ISP dengan ppp di mesin kita. Apabila telah terhubung dan `username` dan `password` nya cocok maka `wvdial` akan memanggil `pppd,` sehingga hubungan dapat terus di lakukan.

Untuk menjalankannya dapat dilakukan sbb :

```
#wvdial &
```

Ini dilakukan untuk menghubungkan dengan nomor isp defaultnya atau

```
#wvdial ISP2 &
```

atau

```
#wvdial ISP3 &
```

Keterangan : tanda "&" digunakan untuk mengirimkan proses `wvdial` ke background, ISP1,2,?dst adalah variable yang diambil dari `wvdial.conf`

## 1.6 Pengenalan Ipchains

`Ipchains` adalah tool administrasi yang digunakan untuk mengaktifkan dan mengawasi aturan-aturan tertentu yang diimplementasikan pada paket yang melewatinya. `Ipchains` biasanya dijalankan di komputer gateway. Aturan-aturan ini dapat dikatagorikan kedalam 4 buah katagori umum yaitu :

- `input`
- `output`
- `forward`
- `user defined`

Aturan firewall menentukan kriteria-kriteria untuk paket dan targetnya. Bila paket tidak memenuhi kriteria tersebut, aturan selanjutnya dalam `ipchains` dibandingkan. Kriteria yang dapat diterapkan pada paket antara lain :

- `ACCEPT.` Menerima paket
- `DENY`. Menolak paket tanpa memberi pesan atau return
- `REJECT`. Menolak paket namun memberi pesan penolakan
- `MASQ`. Membungkus paket seakan-akan paket berasal dari gateway
- `REDIRECT`. Membelokkan paket ke port tertentu
- `RETURN`. Sama dengan `REDIRECT`
- `user defined`. Buatan user sendiri

Parameter untuk `ipchains` cukup banyak, anda dapat mempelajarinya sendiri dengan mengetikkan :

```
# man ipchains
```

atau

```
# info ipchains
```

Secara default `ipchains` dalam sistem Linux dapat kita lihat sebagai berikut :

```
[root@digital /root]# ipchains -L
Chain input (policy ACCEPT):
Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
```

Keterangan :

`Chains` : Aturan
`Policy` : Kriteria
`- L` : parameter untuk melihat aturan yang diterapkan

**Kasus** : *Warnet dengan network* 192.168.0 *akan di masquerade agar client dapat berhubungan dengan dunia Internet seakan-akan client-lah yang terkoneksi langsung ke Internet (bukan melalui server)*

Maka untuk menentukan rule `ipchains` : tentukan bahwa forwarding by default adalah di `DENY`

```
[root@digital /root]# ipchains -P forward DENY
[root@digital /root]# ipchains -L
Chain input (policy ACCEPT):
Chain forward (policy DENY):
Chain output (policy ACCEPT):
```

Kemudian tambahkan pada aturan `forward` bahwa paket dengan `-s (source) 192.168.0.0/24` (192.168.0.0-255 netmask 255.255.255.0) adalah di `MASQUERADE`

```
[root@digital /root]# ipchains -A forward -s 192.168.0.0/24 -j MASQ
[root@digital /root]# ipchains -L
Chain input (policy ACCEPT):
Chain forward (policy DENY):
target     prot opt     source               destination           ports
MASQ       all  ------  192.168.0.0/24       anywhere              n/a
Chain output (policy ACCEPT):
```

**Kasus** : *Menolak semua paket berasal dari* IP 192.168.0.6 *ke port telnet*

Maka untuk menentukkan aturan `ipchains`, yang pertama kali dilakukan adalah membersihkan seluruh aturan :

```
[root@digital /root]# ipchains -F
```

Menolak paket dengan -p (protokol) `tcp --destination-port 23` (target port telnet) dan `-s` (source) `host 192.168.0.6 netmask 255.255.255.255` (32 bit).

```
[root@digital /root]# ipchains -A input -p tcp --destination-port 23 -
s 192.168.0.6/32 -j DENY
[root@digital /root]# ipchains -L
Chain input (policy ACCEPT):
target     prot opt     source               destination           ports
DENY       tcp  ------  digital.adhyaksa.net anywhere              any ->   telnet
Chain forward (policy DENY):
Chain output (policy ACCEPT):
```

## 1.7 Pengenalan superserver Inetd

`Inetd` disebut super server karena didalamnya terdapat banyak daemon yang dijalankan. Konfigurasinya terdapat dalam file `/etc/inetd.conf.` Karakteristik daemon yang dijalankan melalui `inetd` adalah proses turunan akan muncul seiring dengan bertambahnya koneksi pada daemon tersebut.

Konfigurasi biasanya dilakukan dengan mengedit file `/etc/inetd.conf` secara langsung, kemudian merestart inetd dengan cara mengirim sinyal `HUP` pada proses `inetd`, seperti contoh :

**Kasus** : *Non aktifkan daemon ftp dan telnet*

Cara melakukannya : daemon FTP dan Telnet dijalankan melalui super server `inetd`, maka hal yang perlu dilakukan adalah mengedit file `/etc/inetd.conf` dan memberi tanda # pada awal baris `ftp` dan `telnet`. Kemudian jalankan :

```
[root@digital /root]# killall -HUP inetd
```

# Bab 2

# Server Samba

## 2.1 Pendahuluan SAMBA

Samba merupakan implementasi dari protokol **SMB** (Server Message Block) pada sistem UNIX. Protokol ini digunakan oleh MS Windows NT untuk File dan Printing Sharing Service. Dengan mengaktifkan samba pada mesin Linux kita maka kita dapat berbagi file dan printer dengan Windows 95/98 atau Windows NT. Dengan kata lain, dengan menjalankan Samba, maka suatu server Linux dapat tampak seperti suatu Windows NT Server bagi mesin Windows lainnya.

Pada Linux kita dapat me-mounting direktori yang di-share pada Windows juga dapat mengakses secara langsung pada direktori tersebut. Sedangkan pada Windows, kita dapat melihat direktori yang di-share berupa icon yang terdapat dalam **Network Neighborhood**.

## 2.2 Instalasi Samba

Pada RedHat 6.2 paket samba telah diikutsertakan dalam bentuk file-file rpm sebanyak 3 buah, antara lain :

- `samba-common-2.0.6-9.i386.rpm`
- `samba-2.0.6-9.i386.rpm`
- `samba-client-2.0.6-9.i386.rpm`

Install ketiga paket diatas menggunakan perintah rpm :

```
# rpm -ivh samba-common-2.0.6-9.i386.rpm
# rpm -ivh samba-2.0.6-9.i386.rpm
# rpm -ivh samba-client-2.0.6-9.i386.rpm
```

File-file yang ter-install yang sering digunakan untuk mengkonfigurasi dan menjalankan samba antara lain :

- `/usr/bin/smbd`. Merupakan daemon yang menyediakan File and Printing Sharing Service di sistem UNIX untuk SMB Client seperti Windows 95/98 atau Windows NT. Untuk menjalankan daemon ini :

  ```
  # /usr/bin/smbd -D
  ```

- `/usr/bin/nmbd`. Merupakan daemon yang menyediakan penamaan NetBIOS dan kemampuan browsing bagi SMB Client. Untuk menjalankan daemon ini :

  ```
  # /usr/bin/nmbd -D
  ```

- `/usr/bin/smbclient`. Untuk mengakses direktori yang di-share di Windows dengan model FTP. Untuk menggunakannya :

```
# /usr/bin/smbclient
```

Contoh :

```
[root@namec samba-2.0.6]# /usr/bin/smbclient //Planet-3/pic
added inter-
face ip=192.168.0.1 bcast=192.168.0.255 nmask=255.255.255.0
Got a positive name query re-
sponse from 192.168.0.13 ( 192.168.0.13 )
Password:
smb: \>
```

- `/usr/bin/smbmount`. Untuk mounting direktori yang di-share di Windows sehingga dapat dibaca layaknya CDROM yang di mount pada `/mnt/cdrom`. Untuk menggunakannya :

```
# /usr/bin/smbmount
```

Contoh :

```
[root@namec samba-2.0.6]# /usr/bin/smbmount //Planet-
3/oky /mnt/share
Password:
[root@namec samba-2.0.6]# cd /mnt/share
[root@namec share]# ls
```

- `/usr/bin/smbumount`. Untuk unmounting setelah selesai bekerja dengan direktori yang di-mount. Untuk menggunakannya :

```
# /usr/bin/smbumount.
```

Contoh :

```
[root@namec /]# /usr/bin/smbumount /mnt/share
[root@namec /]# cd /mnt/share
[root@namec share]# ls
```

- `/usr/bin/smbstatus`. Melaporkan status koneksi samba. Pada contoh dibawah ini, user `anton` sedang terhubung dengan Home Directory `anton`. Contoh :

```
[root@namec /]# /usr/bin/smbstatus
Samba version 2.0.6
Service uid gid pid machine
----------------------------------------------
anton anton 1004 15514 planet-3 (192.168.0.13) Fri Sep 6 10:150
No locked files
Share mode memory usage (bytes):
1048464(99%) free + 56(0%) used + 56(0%) overhead = 1048576(100%) to-
tal
```

- `/usr/bin/smbadduser`. Menambahkan user ke file user samba (`/etc/smbusers`) dan file password samba (`/etc/smbpasswd`). Contoh :

```
[root@namec /]# /usr/bin/smbadduser pelatihan:training
Adding: pelatihan to /etc/smbpasswd
Adding: {pelatihan = training} to /etc/smbusers
------------------------------------------------------------
ENTER password for pelatihan
New SMB password:
Retype new SMB password:
Password changed for user pelatihan.
```

- `/usr/bin/smbpasswd.` Merubah password user. Contoh :

```
[root@namec /etc]# /usr/bin/smbpasswd pelatihan
New SMB password:
Retype new SMB password:
Password changed for user pelatihan.
```

- `/usr/bin/mksmbpasswd.sh`. Shell script untuk menambahkan user pada `/etc/passwd` milik sistem Linux ke `/etc/smbpasswd` milik samba. Cara menggunakannya :

```
# cat /etc/passwd | mksmbpasswd.sh > /etc/smbpasswd
```

- `/usr/doc/samba-2.0.6/`. Berisi seluruh dokumentasi samba contoh-contoh konfigurasi samba.
- `/etc/smb.conf`. Merupakan file konfigurasi samba.
- `/etc/smbpasswd.` Merupakan password file yang akan digunakan samba untuk proses otentikasi.
- `/etc/smbusers`. Berisi pemetaan user Linux dengan user Windows yang akan digunakan samba untuk proses otentikasi.

Masih ada lagi file-file yang lain yang diikutsertakan dalam paket samba ini, namun tidak sering digunakan. Anda dapat mempelajarinya sendiri dengan membaca petunjuk manual pada :

```
# man samba
# info samba
```

Juga pada direktori `/usr/doc/samba-2.0.6`.

## 2.3 Konfigurasi Samba

Saat daemon-daemon samba dihidupkan, daemon-daemon tersebut akan membaca file `/etc/smb.conf` untuk mendapatkan berbagai informasi yang diperlukan untuk menghubungkan jaringan Windows dengan UNIX. Informasi tersebut antara lain, nama workgroup, password file, direktori yang di-share, hak akses. Berikut ini konfigurasi samba standar pada `/etc/smb.conf` :

```
[global]
# workgroup = NT-Domain-Name atau Workgroup-Name
workgroup = PLANET
# server string = NT Description atau deskripsi server samba
server string = Samba Server
# hanya mengizinkan network 192.168.0 dan network 127 untuk
# mengakses server samba
hosts allow = 192.168.0. 127.
# samba menggunakan file log berbeda untuk tiap mesin yang connect
log file = /var/log/samba/log.%m
# besar file log maksimum
maksimum max log size = 50
# security level, user level atau share level
# User level mengakibatkan proses otentikasi dilakukan 1 kali
# direktori yang di share diakses berdasarkan priviledge user.
# Share level mengakibatkan proses otentikasi berulang-ulang
# direktori yang di share menentukan sendiri permission-nya
security = user
# enkripsikan password bila terkoneksi dengan WIN9x/NT
encrypt passwords = yes
# file password yang digunakan untuk proses otentikasi
```

```
smb passwd file = /etc/smbpasswd
# sinkronisasikan perubahan UNIX password dengan SAMBA password
unix password sync = Yes
# bagian ini dibiarkan default
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
# ==== Share Definitions ====
[homes]
comment = Home Directories
browseable = no
writable = yes
[doc]
comment = Linux Documentation
path = /usr/doc
public = yes
writable = yes
printable = no
[source]
comment = Linux Source
path = /home/ftp/pub
public = yes
[upload]
comment = Upload file
path = /home/ftp/upload
public = no
writable = yes
browseable = yes
readonly = no
```

Keterangan singkat :

- `Comment` : merupakan deskripsi lebih lengkap dari sebuah share
- `Path` : menentukan direktori lokal yang di-share
- `Public` : bila '`yes`' berlaku seperti `anonymous` pada FTP

# Bab 3

# Server FTP

## 3.1 Instalasi server FTP

FTP Server di aktifkan dengan mengeksekusi program ftp daemon yang kemudian akan sipa di *background* dan mendengarkan (*listen*) di port 21 (default) untuk siap menerima request. Yang biasa digunakan adalah `wu-ftpd` yang di kembangkan oleh Washington University. Seperti biasa :)

```
#rpm -ivh wu-ftpd-x.rpm
```

## 3.2 Konfigurasi server FTP

File-file konfigurasi untuk Wu-Ftpd terdapat di `/etc` sebagai berikut :

- `services`
- `inetd.conf`
- `ftpaccess`
- `ftphosts`
- `ftpusers`

FTP Server ini dijalankan oleh internet super server yang disebut `inetd`, suatu file yang akan menentukan cara penanganan port tertentu oleh program daemon

### `/etc/services`

Isi dari file services di `/etc` digunakan untuk mendefinisian port-port yang tersedia dan dapat di gunakan.

```
ftp-data        20/tcp
ftp             21/tcp
```

Dua baris diatas yang mendefinisikan kepada inetd bahwa data dan command di ftp server menggunakan port tersebut.

### `/etc/inetd.conf`

Baris yang menjalankan perintah inisialisasi ftp daemon adalah :

```
ftp     stream          tcp     nowait  root    /usr/sbin/tcpd  in.ftpd  -l -a
```

Keterangan dari baris diatas adalah :

- `ftp` menerangkan nama service
- `stream` menerangkan jenis soket yang digunakan
- `tcp` menerangkan jenis protokol yang digunakan
- `nowait` atau `wait`
- `root` pengguna yang menjalankan daemon tersebut, hal ini akan berkaitan dengan hak pengguna tersebut. Biasanya dituliskan dalam format `user[.group]`
- `/usr/sbin/tcpd` program daemon server yang dijalankan
- `in.ftpd -l -a` argumen yang diberikan pada program server

## `/etc/ftpaccess`

Hal-hal yang penting dalam konfigurasi akses ftp server adalah `class`, `deny`, `limit`, `noretrieve`, `login-fails`, `private`, `autogroup`, dan `guestgroup`. Masing-masing fungsi akan dijelaskan kemudian.

### `class`

Syntax :

```
class  <classname>   typelist   addrglobal
```

Keterangan :

- `classname` : adalah nama sebutan untuk sebuah class
- t`ypelist` : daftar jenis user yang terdiri dari real, guest, anonymous
- `addrglobal` : dapat berupa ip address ataupun host.domain.name dapat menggunakan wildcard "*".

Misal :

```
class  all   real,guest,anonymous *
```

yang berarti ftp server menerima setiap request dari `real user`, `guest`, dan `anonymous` dan dari mesin mana saja

### `deny`

Sintaksnya adalah :

```
deny   addrglobal    message_file
```

Misal :

```
deny    *.planet.tzo.com   /etc/nggak-boleh-masuk.txt
```

Yang berarti ftp server menolak koneksi yang berasal dari semua host di bawah domain `planet.tzo.com.`

### limit

Sintaks :

```
limit   <classname>   n   times   message_file
```

Keterangan :

- n : adalah jumlah user yang diperbolehkan akses ke ftp server secara simultan
- times : Jarak waktu yang di tetapkan biasanya dalam hitungan hari

Misal :

```
class     lokal    real    *
limit  lokal  100   0700-1300   /etc/kebanyakan.txt
```

Yang berarti real user yang di perbolehkan mengakses ftp server adalah sejumlah 100 orang dari jam 7 AM-1 PM. Dan bila pengakses tersebut gagal melakukan login maka ditampilkan pesan pada file /etc/kebanyakan.txt

### loginfails

syntax :

```
loginfails number
```

Fungsinya adalah untuk menentukan berapa kali seorang user boleh salah memasukkan login dan password sebelum disconnect. Misal :

```
loginfails 3
```

Artinya user diberikan mencoba 3 kali memasukkan password sebelum akhirnya koneksi diputuskan bila password tidak tepat.

## /etc/ftphosts

File ftphosts digunakan untuk akses kontrol dari ftp server. syntax :

```
allow   <username>   <addrglobal>
deny    <username>   <addrglobal>
```

## /etc/ftpusers

File ini berisi daftar user yang tidak boleh akses ke ftp server

# Bab 4

# Server DNS

Server DNS bertugas menerjemahkan IP ke nama alamat dan sebaliknya dari nama alamat ke nomor IP. Beberapa cara untuk menerjemahkan alamat Internet antara lain :

- Dengan membaca file lokal `/etc/hosts`
- Dengan memanfaatkan pelayanan DNS Server
- Dengan memanfaatkan pelayanan **NIS** (Network Information System) Server

File `/etc/hosts` ini berisi daftar penerjemahan nama mesin ke alamat IP mesin yang bisa digunakan juga untuk melakukan penerjemahan slsmst IP ke nama. Dengan memiliki file ini, mesin Linux dapat menggunakan nama yang lebih mudah diingat untuk memanggil atau mengakses mesin lain dalam jaringan, daripada harus menggunakan nomor IP. File ini amat sederhana isinya seperti dalam contoh berikut :

```
[root@digital modul]# cat /etc/hosts
192.168.0.6     digital.adhyaksa.net    digital
127.0.0.1       localhost.localdomain   localhost
```

Keterangan :

- Kolom 1 adalah nomor IP
- Kolom 2 adalah **FQDN** (Fully Qualified Domain Name)
- Kolom 3 adalah nama host

File `/etc/hosts` diatas menunjukkan bahwa nama `digital.adhyaksa.net` dan `digital` dipetakan ke nomor IP `192.168.0.6`, nama `localhost.localdomain` dan `localhost` dipetakan ke nomor IP `127.0.0.1`. Kelemahan menggunakan file `/etc/hosts` :

- Semua mesin atau host dalam jaringan harus memiliki file `/etc/hosts` yang identik isinya
- Setiap kali ada perubahan nama host atau nomor IP, maka seluruh file di tiap host harus di-update isinya
- Sangat tidak praktis untuk jaringan dengan host banyak

## 4.1 Pendahuluan DNS

Menggunakan DNS tidak seperti menggunakan file `/etc/hosts`. DNS bersifat client-server sehingga administrasi cukup dilakukan di sisi server saja, sedangkan pada client cukup dikonfigurasi 1 kali yaitu memberi cara agar mesin client dapat menghubungi DNS server. Dalam jaringan Internet, DNS server di seluruh dunia saling bekerja sama dalam rangka menerjemahkan alamat Internet. Network yang lebih besar memiliki DNS server yang menjadi sumber data bagi DNS server pada network dibawahnya. Kerjasama yang dijalin ini dapat digambarkan pada contoh kasus berikut :

**Kasus** : *Proses penampilan gambar atau isi sebuah situs pada browser Netscape yang digunakan seorang pengguna Linux dengan akses dial-up ke sebuah ISP di Indonesia misalnya* `comnet.net.id`. *Saat itu DNS client mengarah pada DNS server dengan* IP `202.150.128.64` *dan* IP `202.150.128.65`.

Perjalanan yang ditempuh untuk menerjemahkan IP secara umum dapat dijelaskan seperti berikut :

1. Browser diarahkan ke situs `http://mail.ngoprek.org`
2. DNS client menghubungi DNS server agar mendapatkan IP domain `mail.ngoprek.org`
3. DNS server mencari data mengenai `mail.ngoprek.org` dengan cara menghubungi DNS server tertinggi yaitu `.` (`dot`) atau root server
4. DNS `root` server menghubungi DNS server `org`
5. DNS server `org` menghubungi DNS server `ngoprek.org`
6. DNS server `ngoprek.org` mengenali subdomain `mail.ngoprek.org` dan berhasil menerjemahkan `mail.ngoprek.org` ke IP `202.135.0.9`
7. IP tersebut dikirimkan kembali ke DNS client kemudian diberikan ke browser
8. Browser mengarahkan langsung langsung ke IP `202.135.0.9` untuk menghubungi web server pada IP tersebut

DNS server terdiri dari 2 jenis server, yaitu :

- **Primary Name Server** (**PNS**) adalah DNS server yang bertanggung jawab atas resolusi domain dan subdomain yang dikelolanya
- **Secondary Name Server** (**SNS**) adalah DNS server yang secara hirarki setara dengan PNS namun data-data domain dan subdomain diperoleh dengan cara menyalin dari PNS

## 4.2 Instalasi BIND 8.2.2

Program DNS yang digunakan oleh Linux RedHat 6.2 adalah `BIND 8.2.2` yang terdiri dari file-file rpm sebagai berikut :

- `bind-8.2.2_P5-9.i386.rpm`
- `bind-utils-8.2.2_P5-9.i386.rpm`
- `bind-devel-8.2.2_P5-9.i386.rpm`
- `caching-nameserver-6.2-2.noarch.rpm`

Gunakan `rpm -ivh` untuk menginstal `bind` pada mesin server Linux.

## 4.3 Client DNS

Client DNS bertugas untuk menentukan server DNS yang digunakan untuk menerjemahkan alamat Internet yang perlu dihubungi oleh program dalam mesin client. Dalam sistem Linux, DNS client merupakan file biasa seperti `/etc/hosts` bernama `/etc/resolv.conf`, namun dengan isi berbeda seperti contoh berikut :

```
[root@digital /root]# ls -l /etc/resolv.conf
-rw-r--r--    1 root     root           66 Dec  3 10:23 /etc/resolv.conf
[root@digital /root]# cat /etc/resolv.conf
search adhyaksa.net
nameserver 192.168.0.4
nameserver 192.168.0.1
```

Pada contoh diatas DNS yang dihubungi adalah IP `192.168.0.4` dan SNS yang dihubungi adalah IP `192.168.0.1`. Tag search berisi sebuah nama yang digunakan sebagai default domain bila resolusi sebuah nama gagal.

## 4.4 Server DNS

DNS membaca data-data resolusi pada sekumpulan file konfigurasi yang terdapat pada komputer lokal. File-file tersebut antara lain :

### `/etc/named.conf`

Berisi konfigurasi DNS server BIND 8.x.x.

```
[root@digital /root]# cat /etc/named.conf
// generated by named-bootconf.pl
options {
        directory "/var/named";
        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query-source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
        // query-source address * port 53;
};
//
// a caching only nameserver config
//
zone "." {
        type hint;
        file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
        type master;
        file "named.local";
};
```

Format `/etc/named.conf` terdiri dari 2 bracket dasar yaitu :

- **Blok Options** Berisi kompulan opsi-opsi global untuk bind 8.x.x, gunakan man `named.conf` untuk mendapatkan informasi lebih detil mengenai opsi-opsi yang tersedia.
- **Blok Zone** Berisi tag-tag yang digunakan untuk menentukan tipe server untuk 1 domain atau subdomain tertentu dan file zona yang berisi konfigurasi 1 domain atau subdomain tertentu.
  - Bila kita bertujuan membuat zona file untuk pemetaan NAME-TO-IP gunakan nama domain sebagai nama zona.
  - Bila kita bertujuan membuat zona file untuk pemetaan IP-TO-NAME gunakan nama domain dengan format sebagai berikut :
    - IP-TO-NAME untuk network 192.168.0, nama zona file ditulis `0.168.192.in-addr.arpa`
    - IP-TO-NAME untuk network 202.150.128, nama zona file ditulis `128.150.202.in-addr.arpa`

### `/var/named/*`

Direktori `/var/named` berisi file-file zona yang namanya bersesuaian dengan tag file pada bracket zone dalam `/etc/named.conf`

```
[root@digital /root]# ls -l /var/named
total 4
-rw-r--r--    1 root     root         2769 Feb  4  2000 named.ca
-rw-r--r--    1 root     root          422 Feb  4  2000 named.local

[root@digital /root]# cat /var/named/named.local
@       IN      SOA     localhost. root.localhost.  (
                                      1997022700 ; Serial
```

```
                                    28800      ; Refresh
                                    14400      ; Retry
                                    3600000    ; Expire
                                    86400 )    ; Minimum
              IN      NS      localhost.

1       IN      PTR     localhost.

[root@digital /root]# cat /var/named/named.ca
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.RS.INTERNIC.NET
;       -OR- under Gopher at    RS.INTERNIC.NET
;           under menu          InterNIC Registration Services (NSI)
;              submenu          InterNIC Registration Archives
;           file                named.root
;
;       last update:    Aug 22, 1997
;       related version of root zone:   1997082200
;
;
; formerly NS.INTERNIC.NET
;
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000      A     128.9.0.107
;
; formerly C.PSI.NET
;
.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                       3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                       3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.     3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                       3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.     3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                       3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.     3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                       3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.     3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                       3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.     3600000      A     192.36.148.17
```

```
;
; temporarily housed at NSI (InterNIC)
;
.                        3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A     198.41.0.10
;
; housed in LINX, operated by RIPE NCC
;
.                        3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A     193.0.14.129
;
; temporarily housed at ISI (IANA)
;
.                        3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A     198.32.64.12
;
; housed in Japan, operated by WIDE
;
.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
; End of File
```

Sedangkan SNS membaca data copy dari PNS melalui sebuah mekanisme transfer data melalui protokol DNS. Data yang dicopy disimpan dalam bentuk file zona yang diletakkan di direktori `/var/named` pada mesin SNS.

Format file zona terdiri dari kumpulan record yang berisikan keterangan yang detil tentang sebuah domain atau subdomain. Record-record tersebut antara lain :

- `SOA` **Start Of Authority** mengawali file zona, berisi data-data waktu sebuah domain atau subdomain. Lebih jelasnya seperti berikut :

```
@       IN      SOA     localhost. root.localhost.  (
                                      1997022700 ; Serial
                                      28800      ; Refresh
                                      14400      ; Retry
                                      3600000   ; Expire
                                      86400 )    ; Minimum
```

Keterangan :

| Isian | Keterangan |
|---|---|
| `@` | Shortcut yang menyatakan nama domain yang bersesuaian dengan zona ini |
| `IN` | Kata kunci protokol INTERNET |
| `SOA` | Nama record SOA |
| `localhost` | Name Server yang menangani domain ini |
| `root.localhost` | Kontak administratif berupa email administrator, dalam hal ini `root@localhost` |
| ( dan ) | Bila ditulis lebih dari 1 baris |
| `Serial` | Nomor urut yang dibangkitkan setiap kali ada perubahan konfigurasi |
| `Refresh` | Interval yang digunakan SNS untuk mengontak PNS |
| `Retry` | Waktu tunggu yang digunakan oleh SNS bila PNS down atau crash |
| `Expire` | Masa berlaku zona untuk SNS tanpa harus melakukan refresh pada PNS jika PNS down |
| `Minimum` | Nilai default untuk masa berlaku data yang disimpan dalam cache |

- `NS`. menyatakan **Name Server** yang berlaku.

```
@               IN      NS      localhost.
```

- `A`. menyatakan **Address Internet** atau alamat IP dari mesin yang ditangani oleh DNS ini proses penerjemahan namanya.

```
@               IN      A       192.168.0.1
digital         IN      A       192.168.0.4
```

- `CNAME`, menyatakan nama **Alias** (**Canonical Name**). Contoh berikut ini menyatakan bahwa `mail` adalah nama alias dari `digital`

```
mail            IN      CNAME   digital.
```

- `PTR`, menyatakan **pointer**, yaitu reversed-address. Contoh berikut ini menmyatakan bahwa IP `192.168.0.4` dipetakan ke nama domian atau subdomain `digital`

```
digital                         IN      A       192.168.0.4
4.0.168.192.in-addr.arpa.       IN      PTR     digital.
```

- `MX`, menyatakan **Mail Exchanger**, digunakan untuk menunjuk mail server yang menangani email domain atau subdomain ini. Contoh berikut ini menentukankan bahwa email untuk `digital.adhyaksa.net` akan diterima oleh mail server dengan prioritas lebih tinggi (`super.adhyaksa.net`). Angka yang lebih kecil merupakan prioritas yang lebih tinggi. Angka yang dimaksud adalah kolom ke-3 pada MX. Mail server pada prioritas selanjutnya akan dihubungi apabila mail server sebelumnya down atau crash.

```
digital IN      MX      0       super.adhyaksa.net.
        IN      MX      10      drutz.adhyaksa.net.
```

- `HINFO`, memberikan keterangan tentang perangkat keras yang digunakan server

```
digital IN      HINFO   "Intel PIII 550 - Linux Redhat 6.2"
```

- `TXT`. menyatakan infomasi umum

```
digital IN      TXT     "Server location : Sukapura - Bandung"
```

**Kasus** : *Konfigurasikan sebuah host menjadi PNS dengan nama domain* `adhyaksa.net` *mempunyai range* IP `192.168.0.1 - 192.168.0.15`.

- Penentuan IP untuk host-host tertentu, misalnya Untuk IP yang lainnya disimpan untuk keperluan mendatang.:

```
ns1.adhyaksa.net   -> IP 192.168.0.1
ns2.adhyaksa.net   -> IP 192.168.0.2
www.adhyaksa.net   -> IP 192.168.0.3
mail.adhyaksa.net  -> IP 192.168.0.3
ftp.adhyaksa.net   -> IP 192.168.0.3
mp3.adhyaksa.net   -> IP 192.168.0.9
vhost.adhyaksa.net -> IP 192.168.0.13
```

- Membuat `/etc/named.conf`

```
options {
        directory "/var/named";
 };
zone "." {
        type hint;
        file "named.ca";
};
zone "adhyaksa.net" {
        type master;
        file "db.adhyaksa.net";
};
zone "0.168.192.in-addr.arpa" {
        type master;
        file "db.192.168.0";
};
```

- Membuat file zona untuk domain `adhyaksa.net` diberi nama `db.adhyaksa.net`

```
@               IN      SOA     ns1.adhyaksa.net. admin.adhyaksa.net.  (
                        200022700 ; Serial
                        28800      ; Refresh
                        14400      ; Retry
                        3600000    ; Expire
                        86400 )    ; Minimum
@               IN      NS      ns1.adhyaksa.net.
@       IN      NS      ns2.adhyaksa.net.
                IN      MX      10      mail.adhyaksa.net.
ns1     IN      A       192.168.0.1
ns2             IN      A       192.168.0.2
www     IN      A       192.168.0.3
mail            IN      A       192.168.0.3
mp3             IN      A       192.168.0.9
vhost           IN      A       192.168.0.13
ftp             IN      CNAME   192.168.0.3
```

- Membuat file zona `reverse-lookup` untuk network `192.168.0` diberi nama `db.192.168.0`

```
@       IN      SOA     ns1.adhyaksa.net. root.localhost.  (
                        200022700 ; Serial
                        28800      ; Refresh
                        14400      ; Retry
                        3600000    ; Expire
                        86400 )    ; Minimum
                IN      NS      ns1.adhyaksa.net.
                IN      NS      ns2.adhyaksa.net.
1               IN      PTR     ns1.adhyaksa.net.
2               IN      PTR     ns2.adhyaksa.net.
13              IN      PTR     vhost.adhyaksa.net.
```

- Tes hasil konfigurasi dngan cara mengaktifkan `bind` dengan perintah berikut :

```
[root@digital /root]# /etc/rc.d/init.d/named start
```

- Lakukan pemeriksaan pada `/var/log/messages`

```
[root@digital /root]# cat /var/log/messages
```

- Cek dengan tool `nslookup` dan `dig`

# Bab 5

# Server Web

## 5.1 Instalasi server Web

Paket program web server dapat kita ambil dari CD instalasi yaitu menggunakan Apache Web Server, yaitu Apache versi 1.3.12. Paket Apache WebServer dapat diinstal dengan menggunakan perintah sebagai berikut :

```
#rpm -ivh apache-1.3.12.rpm
```

Ketika instalasi selesai maka file-file yang perlu di perhatikan adalah:

- `httpd.conf`
- `access.conf`
- `srm.conf`

masing-masing file diatas akan dibahas kemudian.

## 5.2 Konfigurasi Apache

### File `httpd.conf`

Berikut adalah isi dari file `httpd.conf` :

- `ServerType`. Konfigurasi yang menerangkan server, apakah dijalankan melalui `inetd` atau dijalankan secara berdiri sendiri. Bila secara berdiri sendrii, maka server akan dijalankan secara manual.
- `ServerRoot`. Suatu path directory tempat disimpannya file konfigurasi, file error.
- `PidFile`. File yang menyimpan nomor proses dari apache yang dijalankan.
- `ResourceConfig` dan `AccessConfig.` Isi dari file tersebut adalah konfigurasi untuk directory sumber (`access.conf`) dan konfigurasi izin akses (`srm.conf`). Kedua file tersebut adalah bersifat tambahan (optional) , karena keduanya dapat diletakkan di `httpd.conf`, ataupun dapat diletakkan di masing-masing file (`access.conf` dan `srm.conf`)
- `Timeout.` Batas waktu yang digunakan untuk suatu menganggap suatu koneksi terputus, yaitu ketika tidak ada respon dari client.
- `KeepAlive`. Kemampuan server menerima request secara simultan yang berasal dari satu koneksi.
- `MaxKeepAliveRequests`. Jumlah request maksimum yang diterima server secara simultan.
- `KeepAliveTimeout`. Waktu yang ditentukan untuk menunggu request selanjutnya dari satu koneksi.

- `MinSpareServers` dan `MaxSpareServers`. Jumlah server yang dibutuhkan untuk melayani setiap request yang masuk.(biasa digunakan untuk situs web yang sangat sibuk)
- `StartServers`. Jumlah server yang dijalankan oleh apache untuk inisialisasi ketika apache pertama kali di eksekusi.
- `MaxClients`. Jumlah koneksi yang diizinkan secara simultan di server.
- `MaxRequestsPerChild`. Jumlah request yang akan dilayani oleh child server sebelum child server tersebut dimatikan.
- `BindAddress`- Server akan otomatis menerjemahkan setiap alamat semua alamat ip yang ada di server
- `Port`. Port yang akan di "dengarkan" oleh apache web server.
- `User` dan `Group`. User dan group yang menjalankan apache web server.
- `ServerAdmin`. Alamat email dari administrator web server.
- `ServerName`. Nama server yang disesuaikan dengan FQDN (Full Qualified Domain Name), berfungsi sebagai nama dari web server kita.
- `ErrorLog`. Direktori dan nama file dimana kita menempatkan error log dari apache.
- `LogLevel`. Jenis pesan-pesan log yang akan dicatat oleh web server. Ada beberapa kategori yang akan dicatat ke dalam log file, yaitu `emerg` (emergency), `alert`, `crit` (critical), `errors`, `warn`, `debug`.
- `LogFormat`. Bagian ini menenetukan format log file dan juga memberikan "*nickname*" untuk format tersebut. Bagian ini telah diberikan secara default oleh apache, dan sebaiknya tidak usah dirubah (kecuali anda secara pasti tahu apa yang anda lakukan :) )
- `CustomLog`. Log yang di konfigurasikan untuk mencatat setiap access request dari client. Juga digunakan untuk mencatat secara default konfigurasi virtualhost (lihat virtualhost)
- `ServerSignature.` Web Server Signature yang biasa muncul ketika file yang dicari tidak ditemukan dan pada saat ftp. (`on`,`off` dan `email`)
- `UseCanonicalName`. Bagian ini mengkonfigurasikan server apache agar membuat referensi sendiri menggunakan `ServerName` dan `Port` sesuai yang ada di `httpd.conf,` apabila diset `off`, maka server akan merespons sesuai dengan request dari client.
- `HostnameLookups`. Perintah ini akan melog alamat IP dari setiap client yang mengakses server kita. Direkomendasikan bagian ini di set `off` untuk site2 yang sangat sibuk, karena akan dapat membuat log file menjadi besar.

## File `srm.conf`

File `srm.conf` juga digunakan untuk mengatur masalah yang berhubungan dengan directory, DocumentRoot, `UserDir`, `DirectoryIndex` , `MIME` (Multi purpose Internet Mail Extension), `ScriptAlias`, `CGI-Scrip`t dan definisi-definisi dari extension lainnya, misal PHP, Perl dan lain-lain. Isi dari `srm.conf` adalah :

- `DocumenRoot.` Path dari direktori yang berisi file-file dokumen utama dari situs kita.
- `UserDir`. Direktori untuk user yang biasanya terletak di home directory dari user tersebut. URL yang digunakan untuk mengakses adalah `http://server.kita.com/~user` <– menggunakan tanda `~.`
- `DirectoryIndex.` File yang pertama kali akan diakses oleh client ketika client akan mengakses web server kita.
- `FancyIndexing`. Suatu cara menampilkan isi dari suatu directory di server. Menggunakan fancy style atau tidak. Setelah konfigurasi ini biasanya diikuti dengan mendefinisikan path cari icon-icon yang akan digunakan untuk indexing.

- `AccessFileName`. Mendefinisikan akses file yang akan digunakan untuk memproteksi suatu direktori. Biasanya dinamakan `.httaccess` (tanda . Berarti file tersebut di hidden dan untuk kepentingan sekuritas, nama file httaccess diganti)
- `Alias`. Directory alias yang biasa digunakan untuk beberapa direktori khusus, misalkan `cgi-bin.`
- `DefaultType.` Default type dari setiap dokumen yang ada di webserver kita.

### File `access.conf`

File ini digunakan untuk mengatur hak pengaksesan di Web Server atau lebih sering disebut **ACF** (**Access Control File**). Contoh :

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>

<Directory "/usr/local/apache/htdocs">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
    ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"
<Directory "/usr/local/apache/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

Option-option yang umum digunakan untuk hak akses direktori adalah :

- `Indexes` : Perizinkan setiap indexing dari suatu direktori, seperti `AddDescription`, `AddIcon`, `AddIconByEncoding`, `AddIconByType`, `DefaultIcon`, `DirectoryIndex`, `FancyIndexing`, `HeaderName`, `IndexIgnore`, `IndexOptions`, dan `ReadmeName`
- `Limit` : Perizinan untuk mengakses suatu path disesuaikan dengan hostnamenya, seperti `allow`, `deny` dan `order`.
- `Option` : Perizinan untuk menggunakan option-option yang ada dalam masing-masing direktori. Contohnya `Option` dan `XbitHack`.

Perhatian : Apabila pada `httpd.conf` baris dibawah di berikan comment (`#`) berarti semua konfigurasi dari `srm.conf` dan `access.conf` diletakkan di file `httpd.conf.`

```
#ResourceConfig conf/srm.conf
#AccessConfig conf/access.conf
```

## 5.3 VirtualHost

VirtualHost merupakan salah satu fasilitas yang didukung oleh Apache. Fungsi dari Vhost ini adalah untuk membuat multiple host dalam satu mesin. Ada dua cara dalam mengkonfigurasi VirtualHost, atau dengan cara

- **IP-base**, yaitu menggunakan banyak ip dalam satu mesin dan masing-masing ip digunakan untuk satu domain, Contoh dari IP-Based VirtualHost :

```
<VirtualHost 192.168.0.50>
        DocumentRoot /path/to/document
        ServerName test.vhost1.com
</VirtualHost>
```

- **Name-based**. menggunakan satu IP yang kemudian digunakan untuk banyak nama domain. Contoh dari Name-Based VirtualHost :

```
NameVirtualHost 192.168.0.50

<VirtualHost 192.168.0.50>
        DocumentRoot /path/to/document1
        ServerName test.vhost1.com
</VirtualHost>

<VirtualHost 192.168.0.50>
        DocumentRoot /path/to/document2
        ServerName test.vhost2.com
</VirtualHost>
```

## 5.4 Konfigurasi modul-modul Apache

Modul-modul pada Apache adalah interface dimana modul-modul tersebut menentukan fitur-fitur apa saja yang akan dijalankan pada Apache Web Server. Untuk melihat modul-modul yang aktif adalah dengan menggunakan perintah :

```
#/usr/local/apache/bin/httpd -l
```

Untuk meload modul-modul di apache ada dua cara :

- Static Module
- DSO (Dynamic Shared Object)

Apabila kita menginstal Apache mengunakan format rpm maka secara default cara loading modul secara default secara DSO. Untuk me-load modul secara DSO adalah kita tambahkan pada **httpd.conf** baris sebagai berikut (misalkan kita akan meload podul php4 dengan menggunakan DSO) :

```
LoadModule  php4_module     libexec/libphp4.so
AddModule mod_php4.c
```

Contoh lengkap `httpd.conf` dapat dilihat pada lampiran

## 5.5 Menjalankan server Web

Untuk menjalankan Apache yang perlu kita lakukan adalah :

```
#apachectl start
```

Apabila kita mengadakan perubahan file-file konfigurasi maka apache harus di-restart terlebih dahulu agar perubahan yang kita lakukan dapat memberikan pengaruh.

```
#apachectl restart
```

Untuk lebih lengkap melihat pilihan-pilihan dari `apachectl`

```
#apachectl --help
```

File `httpd` yang di hasilkan dari proses instalasi adalah program daemon yang membuka port `80` (default) untuk `LISTEN` (mendengar) setiap request untuk web. Untuk dapat memanggil web server adalah sebagai berikut :

```
http://namec            <---
```

jika nama web server kita adalah `namec` Apabila kita menggunakan port yang tidak dari biasanya (misalkan 8000) maka pemanggilannya sebagai berikut :

```
http://namec:8000
```

Atau pemanggilannya dapat menggunakan IP Address dari mesin yang bersangkutan, misal :

```
http://192.168.0.1    <----
jika IP Address webserver kita 192.168.0.1
```

# Bab 6

# Mail server

## 6.1 Pengenalan server mail

Mail server adalah program daemon yang bekerja menampung dan mendistribusikan email dalam jaringan. Protokol yang umum digunakannya antara lain adalah protokol **SMTP**, **POP3** dan **IMAP**. SMTP (**Simple Mail Transfer Protocol**) digunakan sebagai standar untuk menampung dan mendistribusikan email, sedangkan POP3 (**Post Office Protocol v3**) dan IMAP (**Internet Mail Application Protocol**) digunakan agar user dapat mengambil dan membaca email secara remote, yaitu tidak perlu login ke dalam sistem shell mesin mail server, cukup menghubungi port tertentu dengan mail client yang mengimplementasikan protokol POP3 dan/atau IMAP. Lebih jelasnya, bila disebutkan 'mail server', hal ini dapat menunjukkan pada daemon-daemon yang bekerja dengan cara mengimplementasikan salah satu protokol di atas.

Pada dasarnya untuk membaca email pada mesin mail server terdapat 2 cara, yaitu :

- Secara **lokal,** yaitu dengan cara melakukan login ke dalam sistem shell pada mail server dan membaca langsung email dari mailbox (berupa file atau direktori yang berisi text terformat standar email). Bila hanya ini yang dapat dilakukan, maka mail server cukup menyediakan daemon SMTP tanpa daemon POP3 dan/atau IMAP.
- Secara **remote**, tanpa memasuki sistem shell tetapi melalui port POP3 atau IMAP tergantung mana yang disediakan, dengan menggunakan tool mail client yang mengimplementasikan salah satu protokol mail retrieval (mengambil email secara remote). Melalui cara ini, mail server selain harus menjalankan daemon SMTP, harus juga menjalankan daemon POP3 dan/atau IMAP.

Mengetahui hanya ada 2 cara di atas, tentu anda dapat menyimpulkan sendiri bagaimana suatu layanan email berbasiskan Web seperti `mail.yahoo.com` atau `hotmail.com` bekerja. Dan kini anda mengetahui bahwa pekerjaan membangun mail server, adalah pekerjaan menginstal, mengkonfigurasi dan mengoptimasi daemon SMTP sebagai **MTA** (**Mail Transport Agent**) dan/atau daemon POP3 dan/atau IMAP sebagai **mail retrieval.**

## 6.2 Instalasi server mail

Pada distribusi Linux RedHat 6.2 tersedia beberapa paket yang dapat digunakan untuk membangun mail server yang cukup handal yaitu `sendmail 8.9.3` sebagai MTA, `ipop3d` dan `imapd` sebagai daemon mail retrieval. Paket-paket rpm tersebut antara lain :

- `procmail-3.14-2.i386.rpm`
- `sendmail-8.9.3-20.i386.rpm`
- `sendmail-cf-8.9.3-20.i386.rpm`
- `sendmail-doc-8.9.3-20.i386.rpm`
- `imap-4.7-5.i386.rpm`

- `imap-devel-4.7-5.i386.rpm`

Lakukan instalasi melalui perintah `rpm -ivh` pada paket-paket diatas, penuhi keterkaitan (*dependency*) yang dibutuhkannya, kemudian gunakan perangkat lunak bantu `netconf` untuk mengkonfigurasi MTA sendmail.

## 6.3 Konfigurasi Sendmail

Untuk mempermudah pekerjaan, gunakan `netconf` yang merupakan bagian dari perangkat bantu administrasi `linuxconf` untuk mengkonfigurasi sendmail. Navigasinya sangat mudah, dan di setiap menu pilihan terdapat `HELP` yang mudah dimengerti. Hal ini akan diterangkan kemudian pada jalannya pelatihan.

# Bab 7

# Proxy

## 7.1 Squid sebagai server proxy

Squid adalah salah satu implementasi dari proxy server yang juga menyimpan cache dari setiap respon dari data yang bersangkutan. Singkatnya squid menerima permintaan akses data (request) dari client, dan kemudian meneruskan ke alamat yang dituju (misal : www.yahoo.com), kemudian menyimpan data dari alamat (misal: www.yahoo.com) tersebut disimpan ke dalam direktori squid cache yang kemudian juga diteruskan ke client. Kegunaan squid bila ada permintaan yang sama ke www.yahoo.com, karena sudah ada datanya pada cache maka dapat langsung diberikan tanggapan dari squid server kita tanpa harus meneruskan request tersebut ke www.yahoo.com, ini akan mempercepat akses sehingga dapat menghemat bandwidth.

## 7.2 Instalasi Squid

Instalasi squid dalam format rpm adalah sebagai berikut :

```
#rpm -ivh squid-2.2.3STABLE4.rpm
```

## 7.3 Konfigurasi Squid

Variabel-variabel yang dapat dianggap paling penting dalam konfigurasi squid adalah :

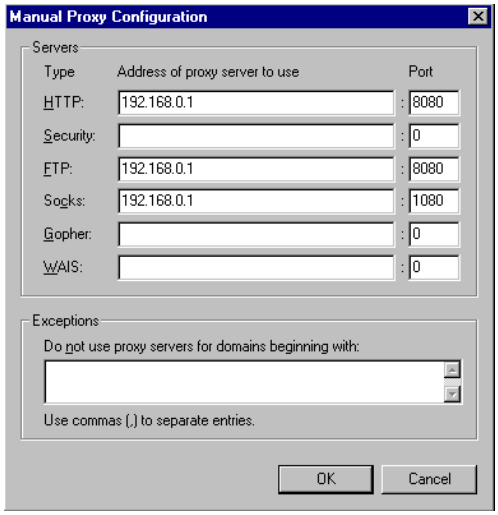
- http_port (nilai defaultnya 3128). Setelah squid dijalankan maka squid akan siap dan mendengarkan (LISTEN) listen di port 3128. Client yang akan menggunakan squid juga harus menggunakan port tersebut agar dapat mengakses squid. Untuk lebih jelas konfigurasi client akan dijelaskan lebih lanjut
- cache_mem (nilai defaultnya adalah 8 MB) .
- cache_swap_high (nilai defaultnya 95%)
- cache_swap_low (nilai defaultnya 90%)
- acl (accsess control list). Acl dapat menentukan user-user yang dapat mengakses squid
- http_access. ini akan mengatur siapa saja yang boleh mengkases squid berdasarkan access control list nya

Untuk konfigurasi lainnya dapat di lihat di file squid.conf pada Lampiran

## 7.4 Konfigurasi client squid

Konfigurasi pada client (Netscape 4.71) : dilakukan dengan urutan menu sebagai berikut :

- `Edit | Preferences | Advanced | Proxies | Manual Proxies Configuration | View`
- Lalu masukkan alamat IP dari server proxy dan port aktifnya.




Gambar 7.1: Konfigurasi client proxy

## 7.5 Menjalankan Squid

Skrip untuk menjalankan program squid terdapat di `/etc/rc.d/init.d`. Untuk menjalankannya dilakukan dengan :

```
#/etc/rc.d/init.d/squid start
```

Sedangkan untuk menghentikannya dapat dilakukan dengan :

```
#/etc/rc.d/init.d/squid stop
```

Anda dapat menguji apakah squid sudah berjalan dengan baik dengan mentesnya dari client atau menggunakan port scanner seperti `nmap` (`www.insecure.com/nmap`), apabila port yang kita tentukan (`http_port`) telah terbuka maka daemon squid telah berjalan dengan baik. File log squid dapat dilihat di `/var/log/squid/` sehingga dapat dimonitor setiap kegiatan yang dilakukan oleh squid ketika diakses oleh client. Hal ini juga ataupun dapat digunakan sebagai pendeteksi dari masalah-masalah yang mungkin timbul.

# Bab 8

# Dasar keamanan jaringan

## 8.1 Security ?

Ketika jaringan kita terhubung dengan sebuah WAN atau terhubung dengan Internet, maka kita tidak hanya harus mempertimbangkan masalah keamanan dari tiap-tiap komputer di dalam jaringan kita, tetapi juga harus memperhatikan keamanan jaringan secara keseluruhan. Kita tidak dapat menjamin bahwa semua orang di "luar sana" adalah orang baik-baik, sehingga permasalahan keamanan jaringan ini merupakan hal yang harus mendapat perhatian yang lebih dari seorang administrator jaringan. Kita juga sebaiknya tidak selalu berpikir bahwa keamanan jaringan bukan hanya berhubungan dengan *hacker* atau *cracker* dari "luar sana" tetapi sering kali ancaman tersebut juga datang dari sisi jaringan internal kita sendiri.

## 8.2 Kepedulian masalah security

Berikut diberikan beberapa contoh hal-hal yang harus diwaspadai dalam keamanan jaringan :

**Password Attack**

- Deskripsi : usaha penerobosan suatu sistem jaringan dengan cara memperoleh password dari jaringan tersebut.
- Pencegahannya : installah `shadow password`, suatu program enkripsi untuk melindungi password.

**Malicious Code**

- Deskripsi : kode-kode pada suatu program yang "tersamar" yang tidak diketahui fungsi dan manfaatnya, tetapi sewaktu-waktu dapat aktif dan beraksi membahayakan keadaan sistem.
- Pencegahan : gunakan program-program seperti `tripwire`, `TAMU`, `sXid` atau dengan menggunakan `MD5Checksum`.

**Sniffer**

- Deskripsi : suatu usaha untuk menangkap setiap data yang lewat dari suatu jaringan.
- Pencegahan : mengenkripsikan semua data yang akan kita lewatkan kedalam jaringan, misalnya menggunakan `ssh` (secure shell) yang mempunyai fungsi yang sama dengan `telnet` tetapi semua data yang dilewatkan kejaringan akan di enkrip dengan enkripsi 128 bit.

**Scanner**

- Deskripsi : merupakan utilitas bantu untuk mendeteksi celah-celah keamanan.

- Pencegahan : pada umumnya program-program scanner menggunakan paket SYN dan ACK untuk mendeteksi celah-celah sekuriti yang ada pada suatu sistem, SYN dan ACK menggunakan ICMP sehingga untuk pencegahannya adalah memfilter paket-paket ICMP dari sistem.

**Spoofing**

- Deskripsi : merupakan penyerangan melalui autentifikasi suatu sistem ke sitem lainnya dengan menggunakan paket-paket tertentu.
- Pencegahan : konfigurasikan sistem untuk menolak semua paket yang berasal dari localhost, memakai program enkripsi untuk akses remote (mis: ssh), mematikan service yang berhubungan dengan "dunia luar" apabila dirasakan kurang diperlukan.

**Denial of Service Attack (DoS)**

- Deskripsi : DoS merupakan serangan yang dilancarkan melalui paket-paket tertentu, biasanya paket-paket sederhana dengan jumlah yang sangat banyak/besar dengan maksud mengacaukan keadaan jaringan target.
- Pencegahan : dilakukan dengan mematikan alamat broadcast , dan memfilter paket-paket ICMP, UDP, serta selalu melakukan update kernel yang digunakan oleh sistem.

## 8.3 Setting beberapa file

Beberapa file perlu dikonfigurasi untuk mengamankan jaringan.

- `/etc/host.allow`. File ini digunakan untuk mengizinkan user dari luar untuk login ke dalam system berdasarkan hostname dan IP addressnya.
- `/etc/host.deny`. Berlawanan fungsi dengan `host.allow`, file ini berisi daftar hostname dan nomor IP address yang dilarang melakukan remote login ke dalam system.
- `/proc/sys/net/ipv4/icmp_echo_ignore_all`. File ini apabila bernilai "1" maka semua paket-paket yang menggunakan port icmp akan di tolak.
- `/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts`. Agak berbeda dengan file sebelumnya, apabila bernilai "1" file ini hanya menolak semua paket-paket icmp yang berasal dari IP broadcasts. jadi tidak seluruh paket icmp ditolak (*deny*)
- `/proc/sys/net/ipv4/conf/all/rp_filter`. File ini digunakan untuk menghindari usaha spoofing dari luar system. Set "`1`" untuk mengaktifkannya.
- `/proc/sys/net/ipv4/tcp_syncookies`. SYN attack adalah sebuah serangan DoS yang akan menghabiskan semua resource cpu dari system. Set "1" untuk mengaktifkannya.
- `/etc/pam.d/su`. Apabila system menggunakan `PAM`, dapat dikonfigurasikan untuk membatasi akses root berdasarkan user. Tambahkan dua baris di bawah pada `/etc/pam.d/su`,agar hanya user dibawah group `wheel` saja yang dapat login sebagai `root`.

```
auth    sufficient      /lib/security/pam_rootok.so debug
auth    required        /lib/security/pam_wheel.so group=wheel
```

- `/etc/lilo.conf`. Untuk lebih mengamankan system tambahkan password dan statement `restricted` pada `lilo.conf` agar tidak semua orang dengan mudah masuk ke dalam sistem dan mempunyai kekuasaan root.

## 8.4 Perangkat bantu IDS (Intrussion Detection System)

Berikut adalah beberapa perangkat lunak bantu yang bisa digunakan untuk pendeteksi penyusup :

- **Portsentry**. Sebuah program bantu yang cukup "ringan" dan tidak begitu sulit untuk mengkonfigurasikan dan menggunakannya. Cocok untuk sistem jaringan kecil.
- **Snort**. Program bantu ini berfungsi memeriksa data-data yang masuk dan melaporkan ke administrator apabila ada "gerak-gerik" yang mencurigakan. Bekerja dengan prinsip program *sniffer* yaitu mengawasi paket-paket yang melewati jaringan.
- **LIDS** (**Linux Intrussion Detection System**) merupakan salah satu tools IDS yang sangat baik dalam melindungi system. Ketika lids aktif, maka bahkan root sekalipun mempunyai akses yang sangat terbatas sekali dalam mengkonfigurasikan system.
- **Carnivore**. Sebenarnya tools ini lebih bisa dianggap sebagai sniffer daripada IDS. Dikembangkan di amerika, kini Carnivore oleh **FBI** dipasang di semua server yang berfungsi sebagai tulang-punggung (*backbone*) Internet yang ada di Amerika. Sehingga secara tidak langsung Amerika telah menyadap semua data yang lewat dari seluruh penjuru dunia. Perlu diketahui bahwa hampir semua server utama atau *backbone* yang ada di dunia ini berlokasi di Amerika Serikat.

Dan masih banyak tools untuk IDS lainnya yang dapat digunakan untuk lebih meningkatkan keamanan sistem.

## 8.5 Informasi sekuriti di Internet

Beberapa informasi tentang security yang dapat diperoleh di Internet :

- `http://www.linuxsecurity.com`
- `http://securityfocus.com`
- `http://www.cert.org`
- `http://atrittion.org`
- `http://packetstorm.securify.com`
- `http://www.karet.org`
- `http://www.securitylinux.net`
- `http://www.securityportal.com`

# Lampiran A. File httpd.conf

```
ServerType standalone
Port 80
HostnameLookups off

User www-data
Group www-data

ServerAdmin webmaster@drutz.adhyaksa.net

ServerRoot /etc/apache

BindAddress *

# The Debian package of Apache loads every feature as shared modules.
# Please keep this LoadModule: line here, it is needed for installation.
# LoadModule vhost_alias_module /usr/lib/apache/1.3/mod_vhost_alias.so
# LoadModule env_module /usr/lib/apache/1.3/mod_env.so
LoadModule config_log_module /usr/lib/apache/1.3/mod_log_config.so
# LoadModule mime_magic_module /usr/lib/apache/1.3/mod_mime_magic.so
LoadModule mime_module /usr/lib/apache/1.3/mod_mime.so
LoadModule negotiation_module /usr/lib/apache/1.3/mod_negotiation.so
LoadModule status_module /usr/lib/apache/1.3/mod_status.so
# LoadModule info_module /usr/lib/apache/1.3/mod_info.so
# LoadModule includes_module /usr/lib/apache/1.3/mod_include.so
LoadModule autoindex_module /usr/lib/apache/1.3/mod_autoindex.so
LoadModule dir_module /usr/lib/apache/1.3/mod_dir.so
LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so
# LoadModule asis_module /usr/lib/apache/1.3/mod_asis.so
# LoadModule imap_module /usr/lib/apache/1.3/mod_imap.so
# LoadModule action_module /usr/lib/apache/1.3/mod_actions.so
# LoadModule speling_module /usr/lib/apache/1.3/mod_speling.so
LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
LoadModule alias_module /usr/lib/apache/1.3/mod_alias.so
LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so
LoadModule access_module /usr/lib/apache/1.3/mod_access.so
LoadModule auth_module /usr/lib/apache/1.3/mod_auth.so
# LoadModule anon_auth_module /usr/lib/apache/1.3/mod_auth_anon.so
# LoadModule dbm_auth_module /usr/lib/apache/1.3/mod_auth_dbm.so
# LoadModule db_auth_module /usr/lib/apache/1.3/mod_auth_db.so
# LoadModule proxy_module /usr/lib/apache/1.3/libproxy.so
# LoadModule digest_module /usr/lib/apache/1.3/mod_digest.so
# LoadModule cern_meta_module /usr/lib/apache/1.3/mod_cern_meta.so
LoadModule expires_module /usr/lib/apache/1.3/mod_expires.so
# LoadModule headers_module /usr/lib/apache/1.3/mod_headers.so
# LoadModule usertrack_module /usr/lib/apache/1.3/mod_usertrack.so
LoadModule unique_id_module /usr/lib/apache/1.3/mod_unique_id.so
LoadModule setenvif_module /usr/lib/apache/1.3/mod_setenvif.so
# LoadModule sys_auth_module /usr/lib/apache/1.3/mod_auth_sys.so
# LoadModule put_module /usr/lib/apache/1.3/mod_put.so
# LoadModule throttle_module /usr/lib/apache/1.3/mod_throttle.so
# LoadModule allowdev_module /usr/lib/apache/1.3/mod_allowdev.so
# LoadModule auth_mysql_module /usr/lib/apache/1.3/mod_auth_mysql.so
# LoadModule pgsql_auth_module /usr/lib/apache/1.3/mod_auth_pgsql.so
# LoadModule eaccess_module /usr/lib/apache/1.3/mod_eaccess.so
# LoadModule roaming_module /usr/lib/apache/1.3/mod_roaming.so
```

```
ExtendedStatus on

ErrorLog /var/log/apache/error.log
LogLevel warn

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %T %v" full
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 /var/log/apache/access.log common

PidFile /var/run/apache.pid

LockFile /var/run/apache.lock

ServerName drutz.adhyaksa.net

UseCanonicalName on

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

MinSpareServers 5
MaxSpareServers 10

StartServers 5

MaxClients 150

MaxRequestsPerChild 30

#Listen 3000
#Listen 12.34.56.78:80
#<VirtualHost host.some_domain.com>
#ServerAdmin webmaster@host.some_domain.com
#DocumentRoot /var/www/host.some_domain.com
#ServerName host.some_domain.com
#ErrorLog /var/log/apache/host.some_domain.com-error.log
#TransferLog /var/log/apache/host.some_domain.com-access.log
#</VirtualHost>
```

# Lampiran B. File srm.conf

```
UserDir public_html
DirectoryIndex index.html
FancyIndexing on
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif */core
AddIcon /icons/deb.gif .deb Debian

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

DefaultIcon /icons/unknown.gif
ReadmeName README
HeaderName HEADER

IndexIgnore .??* *~ *# HEADER HEADER.html README README.html RCS CVS

AccessFileName .htaccess
DefaultType text/plain

AddEncoding x-compress Z
AddEncoding x-gzip gz
AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage da .da
AddLanguage it .it
AddLanguage es .es
AddLanguage br .br
AddLanguage ja .ja
AddLanguage dk .dk
AddLanguage pl .pl
AddLanguage kr .kr
```

```
LanguagePriority en fr de

AddDefaultCharset on
AddDefaultCharsetName iso-8859-1

Alias /icons/ /usr/share/apache/icons/
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

#AddType application/x-httpd-php3 .php
#AddType application/x-httpd-php3-source .phps

#AddHandler cgi-script .cgi
#AddHandler send-as-is asis

BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0

BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

Alias /doc/ /usr/doc/
```

# Lampiran C. File access.conf

```
<Directory /var/www>
Options Indexes FollowSymLinks
AllowOverride None
order allow,deny
allow from all
</Directory>

<Directory /usr/lib/cgi-bin>
AllowOverride None
Options ExecCGI FollowSymLinks
</Directory>

<Directory /usr/doc>
Options Indexes FollowSymLinks
AllowOverride None
order deny,allow
deny from all
allow from localhost
</Directory>

<DirectoryMatch ^/home/.*/public_html>
Options Indexes SymLinksIfOwnerMatch
AllowOverride None
</DirectoryMatch>

<Files .htaccess>
order allow,deny
deny from all
</Files>
```

# Lampiran D. File squid.conf

```
http_port 8080
cache_mem 2 MB

cache_swap_low 90
cache_swap_high 95
maximum_object_size 1024 KB
minimum_object_size 0 KB

cache_dir ufs /squid 70O 16 256
cache_access_log /usr/local/squid/logs/access.log
cache_log /usr/local/squid/logs/cache.log
cache_store_log /usr/local/squid/logs/store.log
mime_table /usr/local/squid/etc/mime.conf
log_mime_hdrs off
pid_filename /usr/local/squid/logs/squid.pid
#reference_age 1 year

#Defaults:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl localnet src 192.168.0.0/255.255.255.0
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

#Default configuration:
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

http_access allow localnet
http_access deny all

cache_mgr squid@planet.tzo.com
cache_effective_user squid
cache_effective_group squid

visible_hostname proxy.planet.tzo.com
memory_pools off
no_cache deny SSL_ports
reference_age 6 hour
```