Networking Linux 1.0

Yantisa Akhadi
(iyan@greits.com)

Bab I
Pengantar GNU/Linux untuk Jaringan Komputer

1.1. Pendahuluan

GNU/Linux(selanjutnya akan disebut sebagai Linux) dan jaringan komputer. Pinguin dan air. Rasanya kedua perbandingan tersebut tidaklah jauh berbeda. Mengingat dukungan terhadap jaringan sangat menyatu ke dalam sistem operasi Linux, dan kehadiran ribuan aplikasi jaringan yang terus dikembangkan, sehingga Linux merupakan pilihan yang amat tepat untuk sebuah jaringan komputer.

Tapi sebelum berbicara tentang Pinguin sebelumnya kita akan bahas airnya terlebih dahulu. Kita akan berbicara tentang jaringan komputer. Apa itu jaringan komputer? Secara umum kita dapat mendefinisikan jaringan komputer sebagai jaringan koneksi antar komputer yang terhubung melalui suatu media. Lalu mengapa kita membutuhkan jaringan komputer? terdapat beberapa alasan kuat mengapa kita membutuhkan jaringan komputer, diantaranya :

- **Berbagi pakai sumber daya (*resource sharing*)**
  Dengan adanya jaringan komputer maka kita dapat berbagi pakai program, peralatan (printer, scanner, media penyimpanan), dan terutama data. Walaupun sumber daya tersebut berada pada jarak ribuan kilometer tapi bagi pemakainya seolah-olah sumber daya tersebut dekat dengannya.
- **Keandalan tinggi**
  Jaringan komputer membuat suatu data atau program yang penting, dapat digandakan pada beberapa komputer(*mirroring*). Hal ini sangat berguna apabila salah satu penyedia layanan data atau program tersebut mati maka dapat segera dialihkan pada penyedia yang lain yang memberi layanan yang sama.
- **Efisiensi**
  Akan jauh lebih efisien bagi suatu perusahaan untuk mempunyai satu printer yang digunakan bersama-sama, daripada memiliki satu printer pada tiap komputer. Langkah efisiensi lain dengan memasang suatu media penyimpanan jaringan (NAS-*Network Attached Storage*), sehingga akan menghemat media penyimpanan yang dibutuhkan tiap komputer.
- **Media komunikasi**
  Jaringan komputer saat ini sangat berperan dalam menjalin komunikasi. Baik melalui teks, suara maupun video, setiap orang yang terhubung ke dalam jaringan yang sama dapat berinteraksi dengan mudah satu sama lain.

1.2 Protokol

Mengapa komputer dengan berbagai jenis sistem operasi, berbagai jenis arsitektur dan media transmisi dapat terhubung satu sama lain? Jawabnya adalah mereka berkomunikasi dengan "bahasa" yang sama. "Bahasa" yang dimaksud disini adalah protokol. Protokol adalah seperangkat aturan yang mengatur komunikasi data antar peralatan. Berikut adalah beberapa protokol yang dikenal dalam dunia jaringan komputer :

- **TCP/IP(*Transfer Control Protocol / Internet Protocol*)**
  Dikenal sebagai "bahasa Inggris" untuk jaringan komputer. Pertama kali dikembangkan oleh Departemen Pertahanan Amerika Serikat(DoD-*Departement of Defense*) melalui ARPA(*Advanced Research Project Agency*). Merupakan protokol yang secara "de facto" menjadi fondasi bagi Internet. Terdiri dari sekumpulan protokol yang menyediakan berbagai macam layanan. Boleh dikata jika anda ingin bergabung dengan komunitas Internet maka mau tak mau anda harus menggunakan TCP/IP. Sekarang memasuki versi ke 6(disebut juga IPng - *IP Next Generation*), dengan penambahan kemampuan yang cukup signifikan.

- **NetWare Protocol**
  NetWare adalah *Network Operating System*(NOS) yang dikembangkan oleh Novell. Saat ini Novell NetWare telah sampai versi 5. Merupakan protokol yang merajai LAN(*Local Area Network*) generasi pertama. Dikembangkan berdasarkan protokol XNS (*Xerox Network Systems*) yang muncul di akhir 70-an. Sering disebut juga sebagai protokol IPX/SPX(*Internet Packet Exchange/Sequenced Packet Exchange*). NetWare dapat berjalan mulai dari PC sampai mainframe, dan mendukung topologi Bus, Token Ring dan FDDI.

- **AppleTalk Protocol Suite**
  AppleTalk adalah protokol yang terdapat pada setiap komputer Apple. Dikembangkan di awal tahun 80-an. Bertujuan agar komputer Apple dapat berbagi pakai file dan printer. Versi AppleTalk yang terbaru disebut AppleTalk Phase 2.

1.3. Media Transmisi

Setelah mengenal beberapa protokol yang ada, sekarang kita beranjak ke media dan topologi. Pemilihan media dan topologi boleh dibilang saling mempengaruhi satu sama lain. Dimana pilihan kita akan suatu media mempengaruhi topologi yang bisa dibentuk dan sebaliknya. Kita lihat dulu jenis-jenis media transmisi

1. **Coaxial**
   Banyak digunakan untuk membuat LAN(*Local Area Network*) karena harga kabel ini relatif murah dan cukup mudah dalam instalasinya. *Radio Government*(RG) membuat standar kabel coaxial seperti RG-8, RG-9 dan RG-11 digunakan untuk *Thick Ethernet* (karena diameter kabel yang relatif besar/tebal), RG-58 digunakan untuk *Thin Ethernet* dan RG-59 digunakan untuk antena TV. Kabel ini dikenal tahan terhadap interferensi. Mendukung transmisi data hingga 10 Mbps. Kekurangannya, kurang fleksibel dan cukup mudah untuk dibajak.
2. **Twisted Pair**
   Terbagi menjadi dua, *Unshielded Twisted Pair*(UTP) dan *Shielded Twisted Pair* (STP). Perbedaanya, pada STP ditambahkan pelindung logam untuk mencegah interferensi noise dan crosstalk. Namun jenis yang paling sering digunakan adalah jenis yang pertama(UTP). *Electronic Industry Association* membuat standar kabel UTP dari katagori 1 sampai 5. Katagori 3 biasa digunakan untuk telepon, sementara untuk komputer biasa digunakan katagori 5, yang mendukung transmisi data hingga 100 MBps. Murah dan mudah dalam instalasinya. Kelemahannya lebih rentan terhadap interferensi, dan jangkauan lebih kecil dibandingkan coaxial.

3. **Serat Optik**

   Jika kedua jenis transmisi sebelumnya menggunakan sinyal listrik yang dihantarkan melalui kabel tembaga, pada serat optik, sinyal ditransmisikan dalam bentuk berkas sinar melalui serat kaca. Sehingga praktis kebal terhadap interferensi. Memang media ini relatif mahal dan cukup sulit dalam pemasangannya, namun mendukung transmisi data lebih dari 1 Gbps dengan jangkuan terjauh (± 2 km). Media ini biasa digunakan untuk sambungan kabel bawah laut. Saat ini mulai banyak diterapkan penggabungan media serat optik dan coaxial yaitu *Hybrid Fiber Coaxial*(HFC).

4. **Nirkabel**

   Pada nirkabel media yang digunakan adalah gelombang elektromagnetik, dengan kisaran frekuensi dari 30MHz sampai 200.000GHz. Pada dasarnya, mekanisme transmisi dibagi menjadi dua tipe : *directional* dan *omnidirectional*. Directional untuk pancaran gelombang yang terfokus pada ke antena penerima. Sebaliknya, pada omnidirectional sinyal ditransmisikan ke segala arah dan dapat diterima oleh beberapa antena sekaligus. Terdapat tiga range frekuensi sebagai berikut :

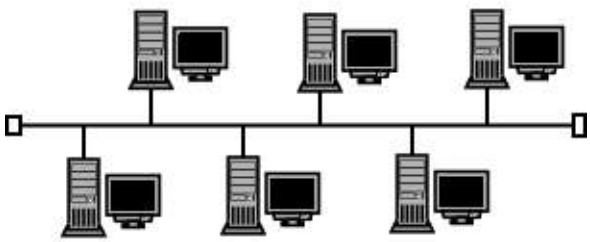
   1. Frekuensi 30 MHz - 1 GHz. Digunakan untuk aplikasi omnidirectional(segala arah), contohnya untuk penerapan broadcast radio. Dalam dunia jaringan komputer contohnya Internet radio paket.
   2. Frekuensi 2 - 40 GHz. Biasa digunakan untuk transmisi directional. Satelit dan WaveLAN adalah contoh penggunaan frekuensi ini.
   3. Frekuensi 300 - 200.000 GHz. Merupakan range spektrum infra merah. Dapat digunakan untuk transmisi directional dengan lingkup terbatas, misal dalam satu ruangan.

1.4. Topologi Jaringan

Yang dimaksud dengan topologi jaringan disini adalah susunan dari jaringan komputer. Atau bagaimana letak antara komputer yang satu dengan komputer yang lain dan jalur yang menghubungkan antar komputer-komputer tersebut. Topologi ini biasanya hanya dapat diterapkan pada jaringan dengan ukuran yang relatif kecil. Karena pada jaringan yang besar bisanya merupakan kombinasi dari beberapa topologi. Berikut beberapa jenis topologi yang umum digunakan :

1. **Bus**

   Pada topologi bus, jalur hanya terdiri dari satu kabel saja. Pada kabel tersebut dipasang T-connector sehingga workstation dapat menghubungkan diri dengan jalur tersebut. Gambaran dari topologi bus adalah sebagai berikut.

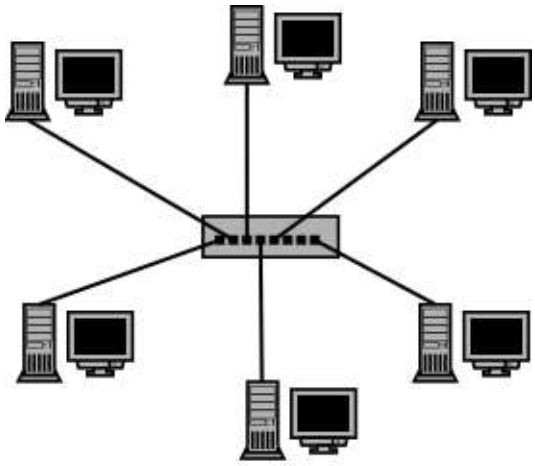


Gb. 1.1. Topologi Bus

Pada topologi ini hanya terdapat satu jalur transmisi yang digunakan bersama-sama. Terdapat terminator pada tiap ujungnya, agar sinyal tidak terus berada pada jalur transmisi. Media transmisi yang menggunakan topologi bus hanyalah coaxial. Biaya untuk membangun jaringan dengan topologi bus paling murah diantara topologi lainnya. Namun saat ini tidak banyak ethernet card yang mendukung topologi ini.

2. **Star**

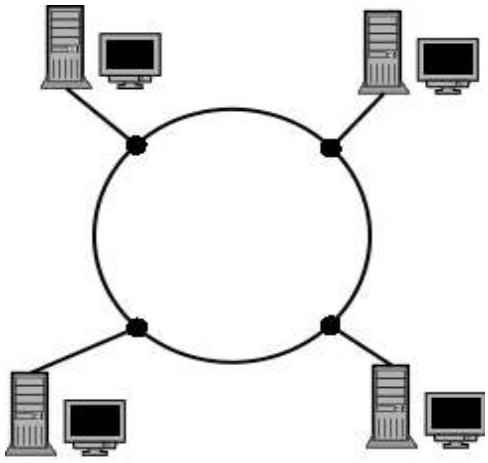
Pada topologi star, terdapat peralatan tambahan yang harus disertakan yaitu pusat dari topologi ini. Pusat disini dapat berupa hub atau switch. Tiap komputer akan terhubung ke pusat ini. Topologi star biasa digunakan pada implementasi ethernet 10Base-T, Fast Ethernet(100 MBps), Gigabit Ethernet(1000 MBps), ATM(*Asynchronus Transfer Mode*) LAN dan Wireless LAN. Untuk lebih jelasnya dapat dilihat pada gambar 1.2.



Gb. 1.2. Topologi Star

3. **Ring**

Topologi ring ini dapat dianggap sebagai topologi bus yang dipertemukan kedua ujung-ujungnya. Setiap komputer akan "terkait" pada jalur jaringan yang berbentuk seperti cincin. Jalur berjalan secara searah, searah jarum jam atau sebaliknya



Gb. 1.3. Topologi Ring

Sebenarnya masih ada beberapa topologi selain topologi diatas, sebut saja mesh, tree, campuran, dan sebagainya, namun pada umunya merupakan pengembangan dari ketiga topologi diatas.

1.5. Linux dan Jaringan Komputer

Karena Linux lahir dan berkembang melalui jaringan komputer, maka Linux sejak awal pengembangannya telah memberi dukungan terhadap jaringan komputer. Baik dari segi protokol, perangkat keras maupun media transmisi. Beberapa kelebihan Linux dalam dunia jaringan adalah sebagai berikut :

- **Fasih dalam banyak protokol**
  Sebut saja mulai dari TCP/IP, IPv6, AppleTalk, NetWare, sampai ISDN dan X.25. Dan Linux disini dapat berfungsi ganda, sebagai client maupun server untuk protokol-protokol tersebut.
- **Dukungan luas terhadap hardware**
  Dari segi hardware, Linux memiliki dukungan yang amat luas terhadap berbagai piranti jaringan. Dari kartu ethernet, token ring, ATM, nirkabel, Bluetooth dan masih sederet piranti jaringan lain. Boleh dikata Linux selalu *up to date* terhadap tren piranti jaringan.
- **Dukungan berbagi pakai file dan printer**
  Pada berbagai lingkungan jaringan, Linux dapat ikut berbagi pakai file jaringan. Seperti lingkungan Windows, Apple dan NetWare serta tak lupa Unix.
- **Sangat siap untuk Internet**
  Dari segala sesuatu tentang Mail(Mail Server, Mail User Agents, Mailing List Software, Remote acces to Mail), Web(Web Server, Web Browser, Web Scripting), FTP, DNS, dan masih banyak lagi layanan yang dapat Linux berikan untuk Internet.
- **Dapat menjalankan aplikasi secara remote.**
  Ini merupakan fitur yang sering tidak diketahui oleh pengguna baru. Dimana dengan Linux, kita dapat menjalankan aplikasi yang mungkin berada ribuan kilometer dari tempat kita berada. Dukungan ini diberikan oleh beberapa aplikasi seperti Telnet, SSH, X Window System, dan VNC.
- **Kaya akan fitur**
  Sebagian dari fitur yang ditawarkan seperti penerapan routing protokol yang lengkap, IP Masqurade, IP Accounting, IP Aliasing, Firewall, Intrusion Detection System dan masih banyak lagi.

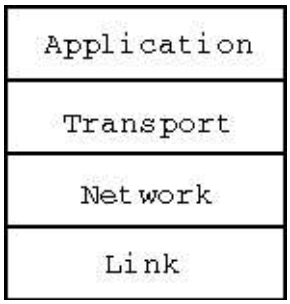
Dari berbagai uraian diatas, kiranya tak salah jika kita katakan Linux dan Jaringan Komputer layaknya seperti Pinguin dan Air.

Bab II
Konfigurasi TCP/IP

2.1. Mengenal TCP/IP

Sebagaimana topik pembahasan kita sebelumnya, TCP/IP memegang peranan yang sangat penting di Internet. Di mana TCP/IP menjadi protokol "de facto" Internet. Bila anda ingin terkoneksi ke Internet maka sistem operasi anda harus mendukung TCP/IP. Namun sekali terkoneksi, anda dapat berkomunikasi dengan bermacam sistem operasi, dan bermacam arsitektur komputer.

TCP/IP sebenarnya merupakan sekumpulan protokol. Terdiri dari empat lapisan (*layer*). Tiap lapisan bertugas menerima data dari lapisan diatasnya dan mengolahnya, kemudian memberikannya pada lapisan di bawahnya. Konsep lapisan ini sangat memudahkan programmer dalam membuat aplikasi jaringan karena yang perlu dipikirkan hanyalah, bagaimana mengirimkan data kepada layer di bawahnya. Detail komunikasi, seperti bagaimana memulai komunikasi, routing jaringan, kontrol kesalahan, tidak perlu dipikirkan. Apa saja empat lapisan tersebut? Dapat dilihat pada gambar berikut.



Gb. 2.1. Empat Lapisan TCP/IP

Tiga lapisan terbawah(*Transport*, *Network*, dan *Link*) diatur oleh sistem operasi, tepatnya oleh kernel. Dan layer paling atas berupa program atau aplikasi yang dijalankan oleh user pada sistem operasi. Penjelasan fungsi masing-masing lapisan adalah sebagai berikut :

Application Layer

Di sini aplikasi jaringan berjalan dengan memanfaatkan lapisan di bawahnya. Aplikasi jaringan akan memilih jenis protokol transport apa yang akan digunakan. Contoh beberapa aplikasi jaringan yang berjalan pada protokol TCP/IP adalah

Telnet, untuk *remote login*

FTP(*File Transfer Protocols*), untuk pertukaran file

SMTP(*Simple Mail Transfer Protocol*), untuk e-mail

SNMP(*Simple Network Management Protocols*), untuk manajemen jaringan

Transport Layer

Lapisan ini memecah data yang dikirim dari layer diatasnya menjadi unit-unit data yang lebih kecil. Tugas lain layer ini adalah untuk membuka dan menutup komunikasi diantara dua komputer. Pada layer ini terdapat dua pilihan protokol yang dapat digunakan. TCP(*Transmission Control Protocols*) dan UDP(*User Datagram Protocols*).

Network Layer

Tugas utama layer ini adalah bertanggung jawab dalam proses pengiriman paket ke alamat yang benar. Seperti pemilihan rute pengiriman. Macam protokolnya , yaitu IP (*Internet Protocols*), ARP(*Address Resolution Protocols*), ICMP(*Internet Control Messages Protocols*) dan IGMP(*Internet Group Management Protocols*).

Link Layer

Layer terbawah ini berfungsi mengirim dan menerima data dari dan ke media jaringan. Dengan kata lain merubah data digital menjadi sinyal listrik atau cahaya dan sebaliknya.

Berikut kita akan membahas pengalamatan dalam TCP/IP.

2.2. Pengalamatan IP

Alamat adalah hal yang sangat penting dalam proses pengiriman dan penerimaan data. Sebagaimana pada semua protokol jaringan lainnya, alamat pengirim dan penerima data merupakan suatu hal yang mutlak harus ada. Begitu pula dalam protokol TCP/IP. Setiap komputer pada sebuah jaringan TCP/IP memiliki sebuah alamat unik dengan ukuran 32 bit. Secara umum alamat ini dibagi menjadi dua, alamat jaringan dan alamat komputer/host. Alamat ini dikeluarkan oleh *Internet Network Information Center* (InterNIC), jika komputer tersebut hendak bergabung dengan Internet. Di Indonesia alamat ini dikeluarkan oleh IDNIC. Sebuah *Internet Service Provider*(ISP) dapat saja membeli satu blok alamat, dan kemudian menjualnya atau mengalokasikannya pada konsumennya. Jika komputer tersebut tidak tergabung dengan Internet maka ia dapat menentukan alamatnya sendiri.

Lalu bagaimana formatnya ?

Alamat IP (*IP Address*) yang berukuran 32 bit ini dipecah menjadi empat kelompok. Masing-masing berukuran 8 bit(disebut juga *octet*)=1 byte, dipisahkan oleh titik dan direpresentasikan dalam bentuk desimal(bahasa kerennya *dotted decimal notation*). Nilai minimal untuk sebuah octet adalah 0 dan nilai maksimalnya adalah 255. Untuk lebih jelasnya lihat gambar di bawah

Terdapat lima kelas alamat IP dengan spesifikasi sebagai berikut :

Kelas A
Oktet pertama dalam desimal : 1 sampai 126
Subnet mask default : 255.0.0.0
Jumlah alamat IP tiap kelas : 16.387.064

Kelas B
Oktet pertama dalam desimal : 128 sampai 191
Subnet mask default : 255.255.0.0
Jumlah alamat IP tiap kelas : 64.516

Kelas C
Oktet pertama dalam desimal : 192 sampai 223
Subnet mask default : 255.255.255.0
Jumlah alamat IP tiap kelas : 254

Untuk kelas D merupakan alamat multicast dan kelas E termasuk kelas eksperimen.

2.3 Subnet

Untuk memahami konsep subnet, sebelumnya kita lihat terlebih dahulu contoh sebuah alamat kelas A :
112. 88. 221. 9

Alamat diatas biasa dipasangkan dengan subnet mask, sehingga bentuk lengkapnya menjadi
Alamat IP : 112.88.221.9
Subnet Mask : 255. 0 . 0 .0

Subnet mask ini sangat memudahkan beberapa peralatan dan software jaringan(contoh: Router) dalam mengirimkan suatu paket data. Analoginya, seorang pegawai pos ketika menyortir surat tidak perlu membaca seluruh alamat surat (seperti nam jalan dan nomor rumah) namun cukup membaca kota tujuannya saja. Mirip dengan kasus diatas peralatan atau software jaringan cukup membaca alamat yang memiliki subnet mask 255. Pada contoh diatas maka alamat yang dibaca hanyalah 112.

Adanya subnet mask ini juga sangat membantu dalam memecah suatu alamat pada kelas A atau B, bahkan juga C. Cara ini disebut *subnetting*. Contoh *subnetting* pada alamat kelas A :

Alamat IP : 17 .189.10 .7
Subnet mask : 255.255.255.0

Dengan adanya perintah diatas maka alamat jaringan yang tadinya 17 menjadi 17.189.10 sedangkan alamat komputer yang tadinya 189.10.7 menjadi hanya 7. Dengan demikian lebih banyak alamat jaringan yang bisa dibentuk dari satu alamat kelas A(dengan kata lain kita membentuk subnet atau sub jaringan).

2.4 Setting alamat IP di Linux

Untuk mengkonfigurasi alamat IP di Linux, kita dapat melakukannya melalui beberapa cara, yaitu:

1. Dengan menggunakan program `netconfig`. Program ini biasanya telah terinstall secara otomatis pada beberapa distro seperti RedHat dan Mandrake. Terdapat empat hal yang bisa dikonfigurasikan; alamat IP, netmask, gateway dan nameserver. Cukup ketikkan saja perintah berikut pada prompt
   ```
   # netconfig
   ```

2. Edit file `/etc/sysconfig/network-scripts/ifcfg-eth0`
   File diatas berlaku pada distro RedHat dan turunannya. Angka 0 menunjukkan urutan kartu jaringan. Jika anda memiliki 2 kartu jaringan pada 1 komputer maka akan terdapat dua file `ifcfg-eth0` dan `ifcfg-eth1`.
   Bila dilakukan pada modem, maka nama file yang diedit harus diubah menjadi `ifcfg-ppp0`. Isi file ini kurang lebih sebagai berikut :

```
#cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
BROADCAST=10.0.0.255
IPADDR=10.0.0.4
NETMASK=255.255.255.0
NETWORK=10.0.0.0
ONBOOT=yes
```

   DEVICE berarti nama peralatan jaringan yang dikonfigurasikan
   BOOTPROTO disini terbagi menjadi menjadi dua; *static* dan *dynamic*. *Static* artinya alamat IP bersifat tetap, *dynamic* jika alamat IP didapat dari DHCP server. Dengan kata lain berubah-ubah(dinamis).
   BROADCAST adalah alamat untuk mengirim data ke semua komputer yang pada alamat jaringan tersebut.
   IPADDR inilah alamat ip kartu ethernet yang terpasang pada komputer yang bersangkutan.
   NETMASK disini sebagaimana yang telah dijelaskan diatas. Untuk menandai mana yang termasuk alamat jaringan dan alamat komputer suatu alamat IP.
   NETWORK berarti alamat jaringan dimana device tersebut terhubung.
   ONBOOT jika diset yes maka sewaktu boot, device akan dihidupkan.

3. Mengubah langsung dari prompt. Untuk cara yang ketiga ini, kita menggunakan utilitas yang cukup umum digunakan pada Unix dan sejenisnya, yaitu `ifconfig`. Namun perubahan ini sifatnya hanya sementara. Karena bila sistem di reboot maka semua konfigurasi akan hilang. Contoh perintahnya sebagai berikut :
   `# ifconfig eth0 182.168.10.17 netmask 255.255.0.0`

Bab III
Utilitas Jaringan

3.1 Latar Belakang

Di Linux, terdapat program-program utilitas untuk jaringan komputer seperti ping, tracert, dan telnet. Dari ukurannya sekilas tampak kecil, namun fungsionalitasnya sangat berperan untuk mengarungi jaringan. Ada yang berfungsi untuk mengetahui kondisi jaringan, untuk adminstrasi jaringan jarak jauh, dan untuk penentuan routing paket. Tak jarang ada pula program untuk merangkum fungsi-fungsi tersebut menjadi satu utilitas yang cukup lengkap. Adanya program-program ini sangat membantu seorang administrator untuk merawat jaringan komputer yang menjadi tanggung jawabnya. Kesemua program tersebut menggunakan protokol TCP/IP, sehingga dapat diterapkan secara luas pada banyak *platform*. Bab ini akan membahas beberapa utilitas yang esensial untuk diketahui seorang administrator jaringan.

3.2 Ping

Pada dasarnya, program ping berfungsi mengirimkan paket datagram dengan protokol ICMP (Internet Control Message Protocol). Paket yang dikirimkan adalah ECHO_REQUEST. Bila pesan ini dibalas dengan paket ECHO_RESPONSE, maka berarti host yang dituju sedang aktif. Namun demikian jika outputnya error atau gagal, belum pasti host yang sedang dituju sedang tidak aktif. Bisa saja terdapat *firewall*(kalem, akan dijelaskan pada bab 8)pada host tersebut, terjadi kesalahan dalam routing, atau jaringan sedang *down*. Berikut kita lihat contoh penggunaannya :

```
$ ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5) from 10.0.0.4 : 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=128 time=0.779 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=128 time=0.406 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=128 time=0.447 ms

--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2011ms
rtt min/avg/max/mdev = 0.406/0.544/0.779/0.167 ms
```

Terdapat beberapa informasi yang dapat kita ambil dari keluaran di atas.
Terlihat alamat pengirim (10.0.0.4) dan alamat tujuan ping(10.0.0.5). Tampak pula tiga paket yang dikirimkan kesemuanya mendapat balasan. `icmp_seq` menampilkan urutan paket yang dikirimkan. `ttl` menampilkan lama hidup paket dalam jaringan, setiap kali melewati router ttl akan berkurang satu. Hal yang menarik di sini, jika kita melakukan ping pada sistem operasi Win* maka hasilnya biasanya 128 sedang jika pada Linux dan kebanyakan sistem Unix lainnya, hasil yang muncul biasanya 255. `time` menunjukkan waktu yang dibutuhkan paket untuk pulang pergi dari pengirim ke penerima.

Sedangkan baris kedua merupakan statistik dari paket-paket yang dikirim, tampak banyaknya paket yang dikirim dan diterima, jumlah kehilangan, total waktu sejak ping dijalankan sampai dihentikan. `rtt` singkatan dari round-trip time, yaitu waktu pulang pergi paket.

3.3 Telnet dan ssh

Telnet digunakan untuk berkomunikasi dengan host lain, dalam hal ini untuk login pada mesin tersebut. Jadi dengan perintah ini kita bisa menggunakan host lain asalkan kita memiliki user dan password pada mesin tersebut. Yang dimaksud dengan host disini adalah sebuah server yang menjalankan daemon telnet-server. Hampir semua sistem operasi mendukung adanya telnet client, namun tidak semua dapat menjadi telnet server.
Contoh sebuah telnet client menghubungi telnet server pada alamat 10.0.0.4 :

```
[iyan@IyaNet iyan]$ telnet 10.0.0.4
Trying 10.0.0.4...
Connected to 10.0.0.4.
Escape character is '^]'.
Red Hat Linux release 7.3 (Valhalla)
Kernel 2.4.18-3 on an i586
login: iyan
Password:
Last login: Mon Apr 28 16:05:56 on tty2
[iyan@IyaNet iyan]$
telnet>
```

Harap dibedakan antara prompt pertama dan kedua. Prompt pertama adalah prompt tempat menjalankan perintah telnet. Sedangkan prompt kedua adalah keluaran dari perintah telnet, jadi kita seperti halnya login biasa ke mesin tujuan. Bila pada prompt kedua kita tekan *escape character* (tombol ctrl diikuti]) maka kita akan keluar dari prompt dan masuk ke mode perintah. Untuk melihat perintah apa saja yang bisa dijalankan pada mode perintah, ketik saja ?.
Telnet memang tampak mudah dan sederhana. Namun kelemahan terbesar dari utilitas ini adalah nama user dan password yang dimasukkan langsung dikirmkan ke telnet server dalam bentuk sebenarnya(*plaintext*) atau tidak disandikan. Sehingga orang dapat saja mencuri dengar paket data yang berisikan nama user dan password.

Solusi atas masalah ini adalah ssh. Ssh merupakan kependekan dari secure shell, dimana semua komunikasi antara client dan server dienkripsi atau disandikan. Sehingga sangat sulit bagi pihak yang tidak berkepentingan untuk mencuri dengar paket yang lewat.
Berikut contoh penggunaannya :

```
[iyan@IyaNet iyan]$ ssh -l rafik 10.0.0.4
rafik@10.0.0.4's password:
Last login: Tue Apr 29 17:22:46 2003 from 10.0.0.4
[rafik@IyaNet rafik]$
```

Sedikit berbeda dengan telnet, jika pada telnet kita bertemu dengan layar login, maka pada ssh jika anda tidak memberitahukan nama user (dengan opsi `-l rafik`), anda otomatis akan login sebagai user anda saat ini. Pada ssh ini tidak dikenal adanya mode perintah.

3.4 Route

Sebelum kita tahu perintah route, kita harus tahu dulu apa itu routing. Routing adalah proses pemindahan paket data dari pengirim ke penerima. Routing sangat berguna pada jaringan dengan banyak komputer, jaringan dengan bermacam kelas alamat dan jaringan kompleks dengan banyak topologi.

Perintah route digunakan untuk memodifikasi tabel routing yang terdapat pada kernel. Fungsi utamanya adalah membuat rute statis ke komputer atau jaringan tertentu. Secara default pada Linux, jika jaringan telah berhasil diset, maka akan terbentuk tabel routing statis. Contohnya sebagai berikut :

```
# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.240 U     0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         10.0.0.4        0.0.0.0         UG    0      0        0 eth0
```

Keterangan output di atas adalah :
`Destination`adalah tujuan alamat, atau tujuan komputer. Maksudnya jika terjadi pengiriman paket ke alamat jaringan tersebut maka akan melalui gateway yang ada. Angka 0.0.0.0 artinya tanpa gateway atau langsung diberikan ke interface yang ada. `Genmask` menunjukkan netmask untuk jaringan tujuan yang dilayani. Bila angka yang muncul 0.0.0.0 maka itu artinya berlaku untuk semua alamat atau default selain alamat netmask yang ditentukan sebelumnya. Mengenai arti dari `Flags`, `Metric`, `Ref`, `Use`, `Iface` dapat melihat langsung ke manual dari route.

Jika anda ingin menambahkan satu gateway lagi, misalkan dengan alamat 10.0.0.5 maka anda dapat menambahkannya dengan perintah :

```
# route add default gw 10.0.0.5 netmask 255.255.255.240
```

Sedangkan untuk menghapusnya cukup mengganti perintah `add` dengan perintah `del`.

3.5 Traceroute

Utilitas ini digunakan untuk melacak rute yang ditempuh oleh paket sampai ke tujuan. Rute disini adalah gateway-gateway yang dilewati oleh paket untuk mencapai tujuan yang telah ditentukan. Secara default banyaknya gateway yang bisa dilewati dibatasi sampai 30 gateway. Berikut contoh penerapan dari perintah traceroute.

```
% traceroute nis.nsf.net.
  traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 38 byte packet
           1  helios.ee.lbl.gov (128.3.112.1)  19 ms  19 ms  0 ms
           2  lilac-dmc.Berkeley.EDU (128.32.216.1)  39 ms  39 ms  19 ms
           3  lilac-dmc.Berkeley.EDU (128.32.216.1)  39 ms  39 ms  19 ms
          4  ccngw-ner-cc.Berkeley.EDU (128.32.136.23)  39 ms  40 ms  39 ms
           5  ccn-nerif22.Berkeley.EDU (128.32.168.22)  39 ms  39 ms  39 ms
           6  128.32.197.4 (128.32.197.4)  40 ms  59 ms  59 ms
           7  131.119.2.5 (131.119.2.5)  59 ms  59 ms  59 ms
           8  129.140.70.13 (129.140.70.13)  99 ms  99 ms  80 ms
           9  129.140.71.6 (129.140.71.6)  139 ms  239 ms  319 ms
          10  129.140.81.7 (129.140.81.7)  220 ms  199 ms  199 ms
          11  nic.merit.edu (35.1.1.48)  239 ms  239 ms  239 ms
```

Dapat dilihat dari perintah tersebut, perjalanan paket melewati beberapa server yang bertindak sebagai gateway. Tampak beberapa server yang terlihat hanya alamat ip-nya saja, hal ini terjadi karena server tersebut tidak mendukung translasi dari alamat ke nama(tidak menjalankan/memiliki server DNS).

Bab IV
DNS Server dengan BIND

4.1. Pendahuluan

Apa itu DNS? DNS merupakan singkatan dari *Domain Name System*. Tugas utama DNS adalah menerjemahkan nama komputer menjadi sebuah alamat IP. DNS juga dapat berfungsi sebaliknya dengan menerjemahkan alamat IP menjadi suatu nama(domain). Contoh kasusnya misal sebuah alamat domain www.iyan.org memiliki alamat IP 123.31.1.82 DNS-lah yang melakukan penerjemahan bahwa www.iyan.org berada pada alamat 123.31.1.82. Atau sebaliknya alamat 123.31.1.82 adalah alamat dari www.iyan.org(*reverse lookup*).

DNS bekerja secara hierarki. Bagian paling atas adalah "." disebut juga root. Dibawahnya ada Top Level Domains, seperti .com, .org, .net, .edu, dan sebagainya. Jadi jika anda mengetikkan www.yahoo.com maka server named pertama akan mencari ke dalam cache apakah masih terdapat alamat tersebut, jika tidak maka ia akan mencari ke "." server, yang kemudian akan memberi petunjuk ke .com sebagai TLD, proses ini berlangsung sampai didapat alamat IP dari komputer tersebut.

Salah satu daemon pada Linux yang berfungsi sebagai DNS server atau name server adalah named. Named merupakan bagian dari paket program BIND(Berkeley Internet Name Domain), yang sekarang sudah mencapai versi 9. Selain BIND, ada juga djbdns yang dibuat oleh pembuat qmail, Daniel J. Bernstein.

4.2. Setting DNS Server

Untuk konfigurasi pada server kita perlu mengedit beberapa file sebagai berikut.

4.2.1. /etc/named.conf

Pada file ini kita menambahkan sebuah zona. Zona di sini adalah nama domain yang kita kelola. Sintaks sebuah zona dalam file `/etc/named.conf` adalah

```
zone  "nama_zone" {
      type tipe;
      file "letak_file";
};
```

contoh ->

```
zone  "iyan.org"{
type master;
      file "iyan";
};
```

Jadi, `nama_zone` disini adalah nama domain. Anda tidak perlu mengetikkan www di awalan domain, karena nanti akan didefinisikan di dalam file zone. `type` adalah jenis dari DNS server apakah sebagai master atau secondary. Jika master, maka DNS server berfungsi sebagai *Primary Name Server*(PNS), dimana ia bertanggung jawab atas resolusi domain dan subdomain yang dikelolanya, jika secondary maka ia berfungsi sebagai *Secondary Name Server*, yang secara hirarki setara dengan PNS, namun data domain dan subdomain diperoleh dengan cara menyalin dari PNS. Sedangkan `file` menunjukkan letak file konfigurasi domain. Secara default file-file konfigurasi domain terletak di dalam direktori `/var/named`, sehingga pada contoh diatas, file konfigurasi terdapat pada `/var/named/iyan`. Berdasarkan

contoh diatas pula maka konfigurasi untuk *reverse-lookup* (mengubah alamat IP menjadi nama domain) adalah

```
zone  "0.0.10.in-addr.arpa" {
      type master;
      file  "0.0.10.in-addr.arpa.zone";
};
```

4.2.2. /var/named

Menurut contoh sebelumnya, pada zone "iyan.org", file konfigurasinya adalah "iyan". Di bawah ini adalah isi file iyan tersebut(`/var/named/iyan`).

```
$TTL 86400
@ IN  SOA   iyan.org.  root.iyan.org. (
            300         ; Serial
            7200        ; Refresh 2 Jam
            7200        ; Retry 2 Jam
            36000       ; Expire 10 Jam
            7200  )     ; Minimum 2 Jam

      IN MX       10 mx.iyan.org.
      NS    ns
ns    A     10.0.0.4
ftp   A     10.0.0.4
mx    A     10.0.0.4
www   A     10.0.0.4
mail  IN CNAME   mx.iyan.org.
```

Dari file diatas tampak bahwa komputer dengan alamat 10.0.0.4 menyediakan beberapa layanan yaitu name server, ftp, web, dan mail. Begitu juga domain iyan.org berada pada 10.0.0.4. Sedangkan isi file zone reverse lookup adalah sebagai berikut :

```
$TTL 86400
@ IN  SOA   iyan.org.  root.iyan.org. (
            300         ; Serial
            7200        ; Refresh 2 Jam
            7200        ; Retry 2 Jam
            36000       ; Expire 10 Jam
            7200  )     ; Minimum 2 Jam

      NS    ns.iyan.org.
4     IN    PTR  ns.iyan.org.
```

4.2.3. /etc/resolv.conf

Di sini anda hanya memastikan apakah baris di bawah ini sudah tercantum atau belum.

```
nameserver  10.0.0.4
```

4.3 Tes Konfigurasi

Sekarang kita akan melakukan penegetesan konfigurasi yang telah diset diatas. Dalam pengetesan ini kita menggunakan program `dig`.

```
$ dig www.iyan.org
```

```
; <<>> DiG 9.2.0 <<>> www.iyan.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29041
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.iyan.org.                IN    A

;; ANSWER SECTION:
www.iyan.org.           86400 IN    A     10.0.0.4

;; AUTHORITY SECTION:
iyan.org.         86400 IN    NS    ns.iyan.org.

;; ADDITIONAL SECTION:
ns.iyan.org.            86400 IN    A     10.0.0.4

;; Query time: 115 msec
;; SERVER: 10.0.0.4#53(10.0.0.4)
;; WHEN: Mon May  5 07:02:15 2003
;; MSG SIZE  rcvd: 79
```

Yang perlu kita perhatikan dari tes diatas adalah status dari query suatu alamat. Tampak bahwa www.iyan.org berada pada komputer 10.0.0.4. Selanjutnya akan dilakukan pengetesan *reverse lookup*.

```
$ dig -x 10.0.0.4

; <<>> DiG 9.2.0 <<>> -x 10.0.0.4
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6324
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;4.0.0.10.in-addr.arpa.       IN    PTR

;; ANSWER SECTION:
4.0.0.10.in-addr.arpa.  86400 IN    PTR   ns.iyan.org.

;; AUTHORITY SECTION:
0.0.10.in-addr.arpa.    86400 IN    NS    ns.iyan.org.

;; ADDITIONAL SECTION:
ns.iyan.org.            86400 IN    A     10.0.0.4

;; Query time: 67 msec
;; SERVER: 10.0.0.4#53(10.0.0.4)
;; WHEN: Mon May  5 07:04:50 2003
;; MSG SIZE  rcvd: 94
```

Bab V
Web Server dengan Apache

5.1. Pendahuluan

Web server adalah sebuah server yang menangani permintaan halaman web suatu domain atau alamat. Permintaan ini biasanya melalui port 80. Pada Linux, web server(Apache) dikenal juga sebagai http daemon. HTTP(Hypertext Transfer Protocols) merupakan protokol yang digunakan untuk berkomunikasi antara client (biasanya disebut browser) dan web server. **Harap diperhatikan Apache yang digunakan pada tutorial ini masih versi 1**.

Mengapa Apache? Terdapat beberapa pilihan mengapa menggunakan Apache, diantaranya

- **Populer**. Apache saat ini merupakan web server yang paling populer, survei Netcraft pada bulan Agustus 1999, menunjukkan bahwa Apache dan turunannya mengisi 55% dari semua domain yang ada.
- **Gratis**. Apache merupakan salah satu produk open source, sehingga source code nya dan programnya tersedia untuk umum. Source code ini kemudian dapat digunakan untuk tujuan komersil dan non komersil.
- **Modular**. Kita dapat dengan mudah menambahkan atau mengurangi modul sesuai dengan keinginan kita. Sehingga kita dapat menentukan spesifikasi web server yang kita inginkan.
- **Portable**. Apache tidak hanya berjalan di lingkungan Unix saja, namun juga dapat dijalankan di Win*, BeOS, mainframe, dan platform lainnya.

Beberapa situs besar yang menggunakan Apache sebagai web server diantaranya Amazon.com, mp3.com, dan Yahoo!. Saat ini bahkan Apache Software Foundation juga mengembangkan berbagai aplikasi yang berjalan di sisi server, seperti Tomcat, yaitu untuk menjalankan Java servlets.

5.2. Setting Apache Web Server

Asumsi di sini anda telah melakukan instalasi Apache. Konfigurasi utama Apache terletak pada file /etc/httpd/conf/httpd.conf. Pada dasarnya, agar Apache dapat berjalan anda hanya perlu mengedit satu baris file saja. Namun, tidak ada salahnya disini juga dijelaskan beberapa konfigurasi yang cukup berguna.

5.2.1. /etc/httpd/conf/httpd.conf

File ini, secara garis besar dibagi menjadi tiga bagian, yaitu :

1. Global Environment, berisi konfigurasi secara umum, seperti jenis server dan modul-modul yang dijalankan.
2. Main Server Configuration, disini kita dapat melakukan setting nilai yang terdapat pada web server, beberapa hal yang dikonfigurasikan seperti nomor port, dukungan SSL, nama server, dan letak dokumen web.
3. Virtual Host, bila anda ingin mengatur beberapa domain dalam satu server, maka disinilah tempatnya.

Di bawah ini beberapa konfigurasi pada `/etc/httpd/conf/httpd.conf`, tepatnya pada main server configuration, yang bisa disesuaikan dengan web server yang anda buat.

Masih berkaitan dengan setting BIND sebelumnya, maka konfigurasi di bawah mengikuti domain yang diset sebelumnya. `ServerAdmin` disini maksudnya alamat kontak Administrator dari web server. Biasanya alamat ini muncul ketika muncul pesan kesalahan yang ditampilkan oleh Apache.

```
ServerAdmin root@iyan.org
```

`ServerName`adalah salah satu konfigurasi terpenting, disini diset nama server yang anda kelola. Dalam hal ini sebagai contoh

```
ServerName iyan.org
```

`DocumentRoot`maksudnya direktori yang berisi semua file-file website anda, contohnya halaman `index.html`. Secara default konfigurasinya akan tampak sebagai berikut

```
DocumentRoot  "var/www/html"
```

Konfigurasi diatas merupakan konfigurasi dasar, anda dapat mempelajari howto atau manual Apache untuk pendalaman yang lebih jauh.

5.3. Situs melalui Home Directory

Misalnya anda memiliki beberapa user pada sistem anda. Dan anda ingin agar tiap user memiliki halaman webnya sendiri-sendiri. Maka ada beberapa cara untuk mewujudkan hal ini. Cara pertama anda membuat direktori untuk masing-masing user. Sehingga bila user bernama Adi ingin membuka halamannya maka ia harus mengetikkan www.iyan.org/Adi/ setiap kali Adi atau orang lain ingin mengakses situsnya. Namun cara ini kurang praktis jika misal terdapat 100 user yang ingin membuat situsnya masing-masing maka administrator harus membuat 100 direktori baru.

Alternatif lain, yang relatif aman dan nyaman adalah tiap user membuat satu direktori di dalam *home directory*-nya, sehingga ia bisa langsung mengedit halaman situs dan langsung melihat hasilnya. Misal, masih tentang user Adi diatas maka urutan langkah penyelesaiannya adalah

1. Ubah permission *home directory* Adi, sehingga group dan dan user lain memiliki hak menjalankan(masuk) ke home directory Adi

   ```
   [Adi@IyaNet home]$ chmod 711 Adi
   ```

2. Buat direktori dengan nama `public_html` di dalam *home directory* Adi.

   ```
   [Adi@IyaNet Adi]$ mkdir public_html
   ```

   Nama direktori ini secara default adalah `public_html` jika anda ingin mengganti dengan nama lain anda dapat mengedit file `httpd.conf` section ke 2.

3. Buatlah sebuah file `index.html` di dalam direktori `public_html`(tepatnya `/home/Adi/public_html/`) sebagai halaman utama user tersebut.

4. Sekarang anda dapat mengakses situs user Adi dengan mengetikkan www.iyan.org/~Adi/ maka akan ditampilkan halaman index yang telah anda buat.

5.4 Subdomain

Selain cara diatas, kita dapat pula membuat subdomain. Hampir semua universitas yang terhubung ke internet menggunakan cara ini. Masih tentang situs www.iyan.org, misal kita ingin membuat subdomain adi di situs tersebut. Sehingga nantinya untuk masuk ke dalam situs yang dibuat oleh adi kita tinggal mengetikkan adi.iyan.org.

Sebelumnya, kita tambahkan pada file `/var/named/iyan`(masih menurut contoh diatas) sebaris konfigurasi berikut

```
adi     A       10.0.0.4
```

Maksud dari baris tersebut adalah kita tambahkan subdomain baru pada domain iyan.org, dalam hal ini subdomain berada pada komputer dengan alamat 10.0.0.4

Kembali kita edit file `/etc/httpd/conf/httpd.conf`, pada bagian ke-3(virtual host). Pada file tersebut sebenarnya sudah terdapat contoh konfigurasi menggunakan virtual host. Dari contoh tersebut kita sesuaikan dengan keinginan kita. Contoh konfigurasi untuk adi.iyan.org

```
NameVirtualHost www.iyan.org

<VirtualHost www.iyan.org>
    ServerAdmin webmaster@adi.iyan.org
    DocumentRoot /var/www/html/
    ServerName www.iyan.org
    ErrorLog logs/iyan.error_log
    CustomLog logs/iyan-access_log common
</VirtualHost>

<VirtualHost adi.iyan.org>
    ServerAdmin webmaster@adi.iyan.org
    DocumentRoot /var/www/adi/
    ServerName adi.iyan.org
    ErrorLog logs/adi.error_log
    CustomLog logs/adi-access_log common
</VirtualHost>
```

Mengapa terdapat konfigurasi dari domain utama anda pada Virtual Host? Karena ketika Virtual Host ini aktif(dengan menghilangkan tanda # di depan NameVirtualHost) maka Main Server Configuration akan diabaikan dan konfigurasi VirtualHost yang teratas menjadi default.

Bab VI
File Sharing

6.1. Pendahuluan

Salah satu tujuan jaringan komputer dibuat adalah kemampuan untuk berbagi pakai resource. Yang dimaksud resource disini dapat berupa file, printer atau media penyimpanan. Dengan adanya kemampuan berbagi pakai ini maka akan terjadi penghematan yang cukup besar dalam pemakaian suatu resource. Di Linux setidaknya terdapat tiga jenis metode untuk berbagi pakai file yaitu NFS, Samba dan FTP.

6.2. Network File System(NFS)

NFS dikembangkan dengan tujuan suatu komputer dapat me-*mount* direktori atau partisi pada komputer lain seolah-olah direktori atau partisi tersebut terletak pada hard disk lokal. Dengan kata lain NFS bertujuan untuk *sharing* file-file antar komputer.

Dari segi keamanan, NFS juga memungkinkan orang untuk melakukan modifikasi pada direktori atau harddisk yang di-*mount* seperti menghapus file, membaca dokumen pribadi yang terdapat pada direktori atau harddisk yang di-*mount*, dan sebagainya, namun hal ini dapat dicegah jika kita mengkonfigurasi NFS secara benar.

NFS juga dibuat untuk berjalan pada lingkungan Unix. Dengan kata lain apaun jenis sistem operasi anda asalkan memenuhi standar POSIX, maka anda dapat berbagi pakai file dengan menggunakan NFS. Sistem operasi tersebut diantaranya Linux, Berkeley Software Distribution Family (BSD, FreeBSD, NetBSD, OpenBSD), Unix Family(Tru64 Unix, HP-UX, AIX, IRIX) dan Sun Family(Solaris,SunOS).

Terdapat beberapa teknologi bagi pakai file selain NFS, sebut saja Samba yang dibuat untuk lingkungan Windows, Andrew File System yang dikembangkan oleh IBM dengan lisensi open source, dan Coda File System yang dikembangkan oleh kalangan Universitas.

6.2.1. Setting Server NFS

Untuk membuat sebuah server NFS, kita hanya perlu mengedit file `/etc/exports`. Setiap baris dalam file ini menunjukkan direktori apa yang di-*share* dan hak akses dari direktori tersebut. Syntax umum dari baris tersebut adalah :

```
direktori workstation1(opsi1) workstation2(opsi2)
```

penjelasannya :

`direktori`

nama direktori yang akan di-share. Aturannya disini, jika anda melakukan *sharing* suatu direktori induk, maka semua direktori di bawahnya juga akan ikut ter-*share*.

`workstation1 dan workstation2`

nama klien yang diperbolehkan mengakses direktori tersebut diatas. Nama disini dapat berupa nama domain atau alamat IP dari workstation tersebut.

`pilihan1 dan pilihan2`

pilihan disini menentukkan hak akses dari workstation terhadap direktori yang di-*share*. Beberapa pilihan yang bisa digunakan diantaranya :

`ro`: *read-only*. Sehingga klien hanya punya hak baca saja. Ini merupakan pilihan default.

`rw` : klien memiliki hak baca dan tulis terhadap direktori yang di-*share*.

`no_root_squash`: root pada komputer klien akan memiliki hak yang sama dengan root pada server NFS.

`no_subtree_check`: biasanya NFS memiliki mekanisme pengecekan apakah file yang akan diminta klien termasuk bagian yang di-*share*. Dengan adanya opsi ini maka akan mempercepat proses transfer, karena pengecekan akan ditiadakan.

`sync`: untuk sinkronisasi antara perubahan yang dilakukan oleh klien dengan direktori sebenarnya di server.

Mari kita lihat beberapa contoh penerapannya pada file `/etc/exports`

```
/home/pingu        192.168.1.1(rw)  192.168.1.2(ro)
/usr/local         *(rw)
```

Pada baris pertama kita men-*share* direktori `/home/pingu` sehingga klien dengan alamat IP 192.168.1.1 dapat memiliki hak baca dan tulis (rw) dan klien dengan alamat IP 192.168.1.2 memiliki hak baca saja.

6.2.2. Setting Client NFS

Untuk setting client pada NFS cukup sederhana. Kita menggunakan perintah mount yang lazim digunakan untuk mounting filesystem. Contohnya sebagai berikut :

```
# mount www.iyan.org:/usr/local /mnt/temp
```

Sehingga isi dari direktori `/mnt/temp` akan sama dengan direktori `/usr/local` yang terletak pada komputer www.iyan.org. Untuk unmounting maka cukup kita ketikkan perintah berikut :

```
# umount /mnt/temp
```

Untuk mendalami lebih jauh anda dapat membaca manual dari NFS atau NFS-howto.

6.3. Samba

6.3.1. Pendahuluan

Samba adalah sebuah paket aplikasi yang berfungsi sebagai perantara antara keluarga Windows(9x, NT, dst) dengan keluarga Unix(Linux, FreeBSD, OpenBSD). Tapi pada awalnya pengembangan Samba adalah pada sistem operasi Linux.

Setidaknya terdapat 4 kemampuan dari Samba yaitu

Berbagi pakai file-file Linux supaya bisa diakses sistem operasi lain atau sebaliknya
Berbagi pakai file antara sesama Linux
Menggunakan printer Linux dari sistem operasi lain dan sebaliknya
Membuat Linux menjadi server WINS(Windows Internet Name Service)

Samba dibuat dengan memanfaatkan protokol SMB (Server Messages Block) yang pada Win family lebih terkenal dengan nama NetBIOS. Protokol ini dikembangkan oleh IBM, Sytee dan Microsoft.

Pada dasarnya Samba terdiri dari 2 daemon(program yang menetap di memori) yaitu *smbd* dan *nmbd*. *smbd* berfungsi mengatur sumber daya yang dapat digunakan secara bersama baik file maupun printer, menyediakan autentikasi dan otorisasi untuk klien Samba. Jadi daemon ini sebagai "bos" yang mengatur aktifitas antara samba server dengan samba client. Sedangkan *nmbd* berfungsi sebagai nameserver, yaitu memberikan informasi soal nama komputer yang terdapat pada jaringan dan menyediakan daftar browse. Untuk mengatur kerja kedua daemon diatas adalah dengan mengedit file `smb.conf` yang biasanya terdapat pada folder `/etc/samba`(pada RedHat 7.3).

6.2.2 Setting Server Samba

Contoh file `smb.conf` yang disederhanakan beserta penjelasannya :
Tiap baris yang diawali oleh `#` merupakan baris komentar. Jangan lupa menjalankan `testparm` untuk test konfigurasi pada `smb.conf`.

```
[global]
# Nama workgroup, disini dapat diisikan nama workgroup tempat amda berada
   workgroup = Network143

# Deskripsi dari komputer
   server string = Iyan on The Nux

# Jika kita ingin user lain dapat membaca dan mengakses file kita maka kita
# set menjadi share, penjelasan yang lebih terperinci terdapat dalam
# /usr/share/doc/samba-2.2.3a/docs/textdocs/security_level.txt
   security = share

#Tempat mendefinisikan direktori yang akan kita bagi
#============================== Share Definition ========================
#Contoh dibawah, direktori Musik yang ingin kita share agar semua user
dapat mengaksesnya namun tidak dapat ditulisi
[Musik]
   comment = Musik
   path = /mnt/win_d/Musik
   read only = yes
   public = yes
#Direktori /sementara di bawah dapat diakses dan juga dapat ditulisi oleh
user lain
[Tampung]
  comment = Penampungan
  path = /sementara
  public = yes
  writable = yes
```

Utiliti Samba yang cukup berguna diantaranya adalah `nmblookup` untuk mengetahui nama komputer host lain, contoh penggunaannya :

```
$nmblookup -A 192.168.1.22
A3B4CH            <00> -          B <ACTIVE>
NETWORK143        <00> - <GROUP> B <ACTIVE>
A3B4CH            <03> -          B <ACTIVE>
A3B4CH            <20> -          B <ACTIVE>
NETWORK143        <1e> - <GROUP> B <ACTIVE>
ARDIANSYAH        <03> -          B <ACTIVE>
```

Untuk browsing sharing komputer yang terdapat pada komputer lain kita bisa menggunakan program LinNeigborhood atau Gnomba.

Bab VII
Mail Server di Linux

7.1. Pendahuluan

Mail server adalah sebuah program yang bertugas menangani transportasi e-mail(selanjutnya akan disebut sebagai mail) dari satu komputer ke komputer lain. Program ini akan terus hidup dan menetap dalam memori selama komputer masih hidup. Dan akan aktif apabila ada mail yang masuk ataupun keluar. Contoh program mail server di Linux adalah Sendmail, qmail, dan postfix.

7.2. Cara Kerja Mail Server

Semua mail yang dikirimkan melalui Internet pada dasarnya terdiri atas dua bagian; mail header dan mail body, yang dipisahkan oleh baris kosong. Mail header berisi alamat asal dan tujuan dari e-mail, judul subjek, tanggal pengiriman, berbagai informasi lainnya. Sedangkan mail body berisi pesan yang dikirimkan. Lalu bagaimana e-mail tersebut dikirimkan kita lihat contoh sebagai berikut.

Misalkan iyan memiliki alamat e-mail di `iyan@iyan.org` ingin mengirim email kepada doyo dengan alamat e-mail `doyo@doyo.net`. Maka mail header dan mail body -nya kurang lebih sebagai berikut :

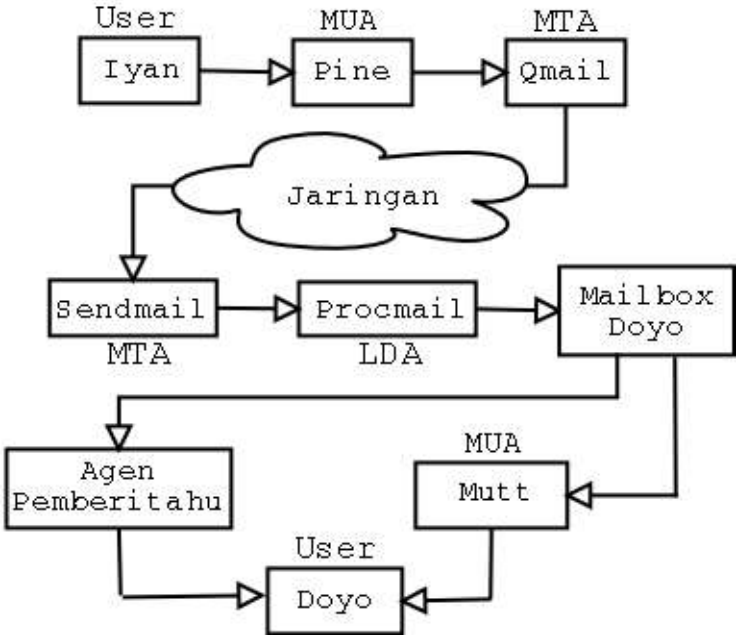
```
From: "Iyan" <iyan@iyan.org>
Message-Id: <20031182812.MAA38998@iyan.org>
Subject: Kapan kumpul modul nich?
To: doyo@doyo.net (Doyo)
Date: Mon, 10 Feb 2003  12:06:22  +0700  (EST)
Content-Type: text

Doyo! Kita sudah dikejar deadline. Mari kita segera selesaikan.

Wassalam
-Iyan-
```

Mengenai susunan dan arti dari Internet mail header dapat dilihat pada RFC 822.
Sedangkan proses pengiriman mail dari iyan ke doyo akan dijelaskan dengan diagram di bawah ini.

7.3. Proses Pengiriman E-mail



Gb 7.1. Proses Pengiriman E-Mail

Penjelasan diagram diatas adalah sebagai berikut :
Misal seorang user bernama iyan akan membuat email. Pertama ia akan menjalankan sebuah program MUA (Mail User Agent), dalam hal ini adalah pine. Pada Linux terdapat banyak program MUA selain pine diantaranya mutt, elm, mush dan mh. Pada Windows contoh MUA adalah Outlook. Fungsi MUA adalah sebagai tempat untuk mengetikkan pesan email baik melalui editor yang terdapat dalam MUA itu sendiri maupun yang dipanggil oleh MUA. Selain itu MUA juga berfungsi untuk membaca email yang ditujukan kepada kita.

Selanjutnya mail tersebut dari MUA diberikan kepada MTA(Mail Transport Agent) dalam hal ini yaitu Qmail, selain Qmail terdapat program lain diantaranya sendmail, smail, dan exim. MTA inilah yang disebut-sebut sebagai mail server. Fungsinya untuk mengirimkan mail ke MTA yang terdapat pada komputer Doyo. MTA mengetahui alamat tujuan dengan membaca header To dan melihat alamat di belakang tanda @(dalam hal ini doyo.net). Dari alamat IP tersebut lalu dibuka sebuah koneksi ke doyo.net. Dalam hal ini MTA menghubungi alamat tersebut melalui port 25 menggunakan protokol SMTP(Simple Mail Transfer Protocol).

Kemudian sesampainya di komputer Doyo, mail tersebut akan diterima oleh MTA yang terdapat pada komputer Doyo, dalam hal ini sendmail. Sendmail merupakan server mail yang terdapat di hampir semua distribusi Linux.

Dari MTA, mail kemudian diberikan kepada LDA(Local Delivery Agent). Pada Linux LDA ini biasanya berupa program dengan nama procmail, walaupun terdapat program-program

LDA lainnya. Tugas LDA adalah mengirimkan mail tersebut ke mailbox Doyo. Tugas ini sengaja tidak dilakukan oleh MTA, untuk menjaga kesederhanaan program dan agar MTA hanya berfokus pada penanganan koneksi dengan mail server lain.

Mailbox yang terdapat dalam komputer Doyo biasanya berupa sebuah file yang bernama / usr/spool/mail/doyo atau var/mail/doyo. Ketika Doyo akan membaca mail yang masuk maka ia akan menjalankan MUA dalam hal ini mutt. Mutt kemudian membaca file mailbox dan menerjemahkannya sehingga Doyo dapat membacanya.

Dalam proses pengiriman surat terdapat satu program yang tak kalah pentingnya, walau program tersebut tidak berperan dalam transportasi mail. Program ini disebut agen pemberitahu, yaitu sebuah program yang mengawasi bila ada mail yang masuk dan kemudian memberitahukannya kepada user.

Apa yang terjadi apabila komputer Doyo tidak dapat dihubungi oleh komputer Iyan? Maka MTA pada komputer Iyan akan mencoba mengirimkan mail tersebut dalam interval waktu tertentu dan dalam batas waktu tertentu. Pada sendmail interval waktunya adalah 15 menit dan batas waktunya adalah 4 hari. Lebih dari itu mail akan dianggap gagal dan dikembailkan ke Iyan.

7.4. Remote mail dan protokolnya

Kebanyakan pengguna komputer saat ini terkoneksi ke Internet melalui ISP(Internet Service Provider) seperti TelkomNet, MegaNet, M-Web dst. Mail-mail user biasanya dikirimkan melalui mailbox yang terdapat pada ISP. Karena jarang sekali user terkoneksi secara full 24 jam ke ISP(kecuali mampu :)).

Namun demikian user biasanaya ingin agar mail-mail yang terdapat mailbox ISP dapat diambil dan kemudian koneksi dapat diputuskan(untuk menghemat biaya dan efisiensi bandwith). Untuk itu digunakan protokol remote-mail, sehingga mail yang terdapat pada suatu server dapat diambil untuk kemudian dibaca pada komputer klien. Terdapat dua jenis protokol remote mail yaitu POP3 (Post Office Protocol versi 3 - didefinisikan oleh RFC 1939) dan IMAP(Internet Message Acces Protocol - didefinisikan oleh RFC 2060). Hampir semua ISP mendukung POP3 dan baru beberapa yang mendukung IMAP.

Untuk menggunakan protokol tersebut digunakan program remote-mail client program. Beberapa MUA seperti mutt misalnya, telah mendukung kedua protokol diatas.

7.5. Setting Mail Server dengan Postfix

7.5.1. Latar Belakang

Postfix adalah sebuah mail server yang dikembangkan oleh Wietse Venema. Postfix dibuat dengan tujuan menjadi sebuah mail server yang cepat, mudah dikonfigurasikan dan aman.

Mengapa Postfix? Terdapat beberapa alasan mengapa pilihan jatuh terhadap Postfix seperti :

- **Keamanan**. Postfix dikenal aman, terlebih jika dibandingkan dengan Sendmail yang sering menjadi bulan-bulanan para hacker. Namun demikian Sendmail sering menjadi pilihan default beberapa distro.

- **Kemudahan**. Setting Mail Server dengan postfix hanya membutuhkan ±10 menit saja. Jika dibandingkan dengan Qmail yang memang terkenal akan keamananya namun cukup sulit dalam instalasinya. Terlebih dengan banyaknya script tambahan.
- **Performa**. Postfix mampu menangani jutaan pesan yang berbeda dalam satu harinya. Dan klaim dari pembuatnya, Postfix lebih cepat tiga kali dibandingkan kompetitor terdekatnya.

7.5.2. Instalasi dan Konfigurasi

Sebelum melakukan instalasi, postfix memerlukan paket `db*-devel` dimana `*` berupa bilangan, pastikan terlebih dahulu paket tersebut sudah terinstall. Paket tersebut dapat ditemukan pada CD 2 RedHat 7.3.

Untuk instalasi kita menggunakan paket dalam bentuk tar.gz. Dalam modul ini Postfix yang digunakan adalah versi 2.0.8. Segera saja kita ikuti langkah-langkah instalasinya :

1. Kopikan file `postfix-2.0.8.tar.gz` ke direktori `/usr/src`, lalu masuklah ke direktori tersebut.
2. Kemudian ekstrak file tersebut dengan perintah
   ```
   # tar xzf postfix-2.0.8.tar.gz
   ```
3. Masuk ke direktori yang baru terbentuk lalu lakukan perintah sebagai berikut, untuk mengkompile *source code* dari postfix
   ```
   # make
   ```
4. Kemudian buat sebuah user baru dengan nama postfix dan group dengan nama postdrop
   ```
   # useradd postfix
   # groupadd postdrop
   ```
5. Barulah kita melakukan instalasi dengan perintah
   ```
   # make install
   ```
6. Kemudian kita edit file konfigurasi utama postfix yang terletak pada `/etc/postfix/main.cf`. Disini kita harus mensetting beberapa baris. Contoh konfigurasinya adalah sebagai berikut.
   ```
   myhostname = www.iyan.org
   mydomain = iyan.org
   myorigin = $mydomain
   inet_interfaces = all
   mydestination = $mydomain
   ```
7. Kemudian kita buat(jika tidak ada) file `/etc/aliases`, tambahkan baris berikut
   ```
   poftfix: root
   ```
   Kemudian jalankan perintah
   ```
   # newaliases
   ```
8. Barulah kita menjalankan daemon postfix dengan perintah
   ```
   # postfix start
   ```
9. Agar Postfix selalu berjalan tiap kali startup Linux maka tambahkan baris berikut pada file `/etc/rc.d/rc.local`, perintah tersebut akan menjalankan postfix ketika startup.
   ```
   /usr/sbin/postfix start
   ```

BAB VIII
Firewall di Linux

8.1. Pendahuluan

Mengapa diberi nama firewall? Mungkin itu pertanyaan yang pertama kali muncul ketika membaca artikel tentang keamanan, baik di Win, Linux maupun sistem operasi apa saja. Menurut istilah konstruksi bangunan, firewall adalah sebuah struktur yang dibuat untuk mencegah penyebaran api. Hampir senada dengan istilah awalnya, Firewall bertujuan menjaga LAN dari "api" akses yang tidak diinginkan dari Internet. Disamping agar pengguna LAN tidak sembarangan mengeluarkan "api" aksesnya ke Internet. Dengan kata lain firewall dibuat untuk membatasi antara dua dunia(LAN dan Internet).

Firewall yang pertama kali dibuat adalah sebuah mesin Unix yang menjadi perantara antara sebuah LAN dengan Internet. Jika seorang user ingin mengakses Internet maka ia harus login terlebih dahulu ke mesin Unix untuk kemudian mengakses dari situ. Begitu pula ketika seorang user ingin mendownload data dari Internet maka ia harus download terlebih dahulu pada mesin Unix, baru dipindah pada workstation user.

Firewall sendiri terbagi menjadi dua jenis

1. Filtering Firewalls - yang akan memblok dan melewatkan paket-paket tertentu
2. Proxy Servers - berfung sebagai perantara koneksi

8.2. Packet Filtering Firewalls

Packet filter adalah sebuah software yang memeriksa header dari paket ketika paket tersebut lewat, dan memutuskan tindakan apa yang dilakukan terhadap paket tersebut. Apakah paket tersebut di-DROP (misal dengan menghapus paket tersebut), ACCEPT(misal, paket tersebut diteruskan ke tujuannya), atau hal lain yang lebih kompleks.

Pada Linux, packet filtering ditanamkan pada kernel(sebagai modul kernel, atau digabungkan ke dalam kernel). Penerapan packet filtering sudah cukup lama sejak kernel 1.1. Versi pertamanya, masih banyak mencontoh cara kerja `ipfw` milik BSD(Sistem Operasi buatan University California at Berkeley), dibuat oleh Alan Cox pada akhir 1994. Berkembang menjadi `ipfwadm` pada kernel 2.0, `ipchains` pada kernel 2.2 dan terakhir `iptables` sejak kernel 2.4.

8.3. Packet Filtering Firewalls dengan Iptables

Iptables merupakan paket program yang disertakan secara default oleh banyak distro bersama dengan kernel versi 2.4. Pada iptables nantinya kita akan banyak berhubungan dengan aturan-aturan(rules) yang menentukan tindakan apa yang akan dilakukan terhadap sebuah paket. Aturan-aturan ini dimasukan dan dihapus pada tabel packet filtering yang terdapat pada kernel. Sekedar mengingatkan kernel adalah "jantung" sistem operasi yang terus berada pada memori sejak komputer booting hingga komputer dimatikan. Sehingga aturan apapun yang kita tentukan akan hilang pada saat terjadi rebooting, namun demikian terdapat beberapa cara

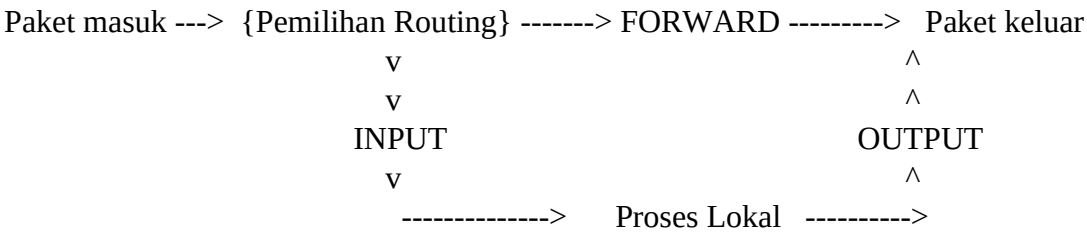
agar aturan-aturan yang telah kita buat dapat di kembalikan pada saat Linux booting, yaitu :

1. Menggunakan perintah iptables-save, untuk menyimpan aturan-aturan yang telah ditentukan dalam sebuah file, dan iptables-restore, untuk memanggil file aturan yang telah dibuat.
2. Meletakannya pada sebuah skrip yang akan berjalan pada saat inisialisasi Linux.

8.3.1. Perjalanan Paket Melintasi Filter

Terdapat tiga daftar aturan pada tabel filter. Daftar-daftar ini disebut "firewall chains" atau "chains". Ketiga chains tersebut adalah INPUT, OUTPUT dan FORWARD.

Chains tersebut tersusun kurang lebih sebagai berikut :



Ketika paket melewati salah satu chains (INPUT, OUTPUT atau FORWARD), maka chain akan dilihat untuk menentukan "takdir" dari paket tersebut. Jika menurut chain paket tersebut harus di DROP maka paket akan dihapus, begitu juga sebaliknya jika menurut chain paket tersebut di-ACCEPT maka paket tersebut akan melanjutkan perjalanannya.

Jadi chain sebenarnya merupakan daftar aturan-aturan. Tiap aturan mengatur **tindakan apa** yang akan dilakukan terhadap sebuah paket berdasarkan **header** dari paketnya. Jika aturan pertama tidak cocok dengan header paket maka akan dilanjutkan dengan aturan berikutnya, begiru seterusnya. Hingga apabila tidak ada lagi aturan yang sesuai dengan header paket maka kernel akan melihat pada chain policy, yang berisi aturan umum tentang tindakan terhadap suatu paket. Pada kebanyakan sistem, chain policy biasanya akan men-DROP paket tersebut.

8.3.2. Menggunakan iptables

Seperti sudah disebutkan sebelumnya, terdapat tiga chain dasar yaitu:

INPUT
OUTPUT
FORWARD

Untuk memanipulasi chain terdapat beberapa option sebagai berikut :

1. Membuat chain baru (-N)
2. Menghapus chain yang kososng (-X)
3. Mengganti policy untuk chain built-in (-P)
4. Melihat aturan-aturan dari sebuah chain (-L)
5. Menghapus seluruh aturan dari sebuah chain (-F)
6. Mengosongkan paket dan mengeset nol semua aturan pada sebuah chain (-Z)

Terdapat beberapa hal yang dapat kita lakukan sehubungan dengan aturan(rules), yaitu :
1. Menambah aturan baru pada sebuah chain (-A)
2. Memasukkan aturan baru pada posisi tertentu dalam sebuah chain (-I)
3. Mengganti sebuah aturan pada posisi tertentu dalam sebuah chain (-R)
4. Menghapus sebuah aturan pada posisi tertentu, atau aturan pertama yang cocok (-D)

Juga terdapat beberapa parameter berikut, beserta contoh penggunaannya :

Memilih tindakan (-j atau --jump)

Secara garis besar terdapat 3 tindakan terhada suatu paket yaitu, ACCEPT (paket boleh lewat), DROP (paket berhenti / diblok) dan REJECT (mirip dengan DROP bedanya akan diberitahukan ke alamat IP asal/sumber paket bahwa paket ditolak). Contoh :

```
# iptables -A INPUT -j DROP
```

Perintah diatas akan menambahkan aturan ke dalam chain INPUT, dimana semua paket yang ditujukan ke IyaNet akan di-DROP. Untuk menghapus aturan diatas digunakan perintah

```
# iptables -D INPUT -j DROP
```

Hasil yang sama juga akan didapatkan dengan perintah

```
# iptables -D INPUT 1
```

jika aturan yang dimaksud merupakan aturan pertama.

Memilih peralatan input (-i atau --in-interface) dan memilih peralatan output (-o atau
--out-interface)

Parameter ini sangat berguna bila kita memiliki beberapa peralatan masukan dan/atau keluaran, misal kita memiliki 2 kartu jaringan (eth0 dan eth1). Perlu diingat bahwa chain INPUT hanya memiliki peralatan input. Sehingga jika kita menambah parameter -o pada chain INPUT, tidak akan match!. Begitu juga sebaliknya untuk chain OUTPUT. Hanya chain FORWARD saja yang dapat menggunakan kedua parameter.

Contoh :

```
# iptables -A INPUT -i eth0 -j DROP
```

Perintah diatas akan men-DROP semua paket yang berasal dari eth0. Contoh penerapan pada chain FORWARD adalah sebagai berikut :

```
# iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

Perintah tersebut akan meng-ACCEPT semua paket yang masuk dari eth0 dan akan keluar menuju eth1.

Memilih jenis protokol (-p atau --proto)
Jenis protokol yang sering digunakan diantaranya adalah TCP, UDP dan ICMP.

```
# iptables -A INPUT -p icmp -j DROP
```

Dengan perintah diatas maka akan menambahkan aturan ke dalam chain INPUT dimana untuk setiap paket yang datang dan berjenis protokol icmp (contoh : ping), maka paket tersebut akan di-DROP

Memilih sumber/asal paket (-s atau --source) dan memilih tujuan paket (-d atau --destination)
Ada empat cara pendeklarasian sumber atau tujuan paket yaitu :

1. dengan nama domain, misal www.yahoo.com
2. dengan alamat IP, misal 192.168.1.23
3. dengan range alamat berdasarkan most significant bit, misal 212.99.221.0/24
4. dengan range alamat berdasarkan subnet mask, misal 212.99.221.0/255.255.255.0, dimana hasilnya akan sama dengan no.3.

Contoh :

```
# iptables -A INPUT -s www.hackers.net -j DROP
```

Perintah diatas akan menambahkan sebuah aturan ke dalam chain input, dimana semua paket yang berasal dari www.hackers.net akan di-DROP.

Kebalikan dari (!)
Parameter inversi ini dapat digabungkan dengan parameter sebelumnya, sebagai contoh :

```
# iptables -A INPUT -s ! www.hackers.net -j ACCEPT
```

Perintah diatas akan memberikan hasil yang sama dengan contoh perintah sebelumnya, dimana semua nama domain selain www.hackers.net akan di-ACCEPT.

Dibawah ini beberapa contoh penggunaan gabungan parameter tersebut diatas

Contoh 1

```
# iptables -A INPUT -s 178.191.122.45 -p tcp --dport telnet -j DROP
```

Maksud perintah diatas adalah menambah aturan ke dalam chain INPUT, dimana paket yang berasal dari alamat IP 178.191.122.45 dan berjenis protokol tcp serta memiliki port tujuan (destination port) telnet akan di-DROP. Penggunaan -dport dan -sport(source port atau port tujuan) merupakan perluasan dari penggunaan parameter -p tcp.

Contoh 2

```
# iptables -A OUTPUT -p tcp --dport telnet -i eth1 -j DROP
```

Perintah diatas akan menambah aturan ke dalam chain OUTPUT, dimana semua paket yang menggunakan protokol tcp, memiliki port tujuan telnet dan keluar melalui peralatan eth1 akan di-DROP

Contoh 3

```
# iptables -A INPUT -i eth0 -p tcp --dport ! 80 -j DROP
```

Aturan diatas berarti semua paket yang datang melalui eth0, menggunakan protokol tcp, dengan alamat tujuan selain port 80 akan di-DROP.

Untuk mengetahui nama-nama port beserta nomornya anda dapat melihat file `/etc/services`.

Daftar Pustaka

1. Cisco System, *Cisco Internetworking Technologies Handbook*, www.cisco.com.
2. Forouzan, Behrouz A. , *Local Area Network*, Mc Graw Hill, 2003.
3. Stevens, W. Richard, *TCP/IP Illustrated vol.1 : The Protocols*, Addisson-Wesley, 1994.
4. Tanenbaum, Andrew S. , *Jaringan Komputer jilid 1*, Prenhallindo, Jakarta, 2000.