

LINUX SYSTEM ADMINISTRATOR



Disusun Oleh :
Tim Komisi Teknik
Bina Sarana Informatika

**Akademi Manajemen Informatika dan Komputer Bina Sarana
Informatika
BSI – Jakarta
2010**

KATA PENGANTAR

Alhamdulillah saya ucapkan pertama kali untuk dapat menyatakan rasa syukur saya. Setelah mungkin selama sebulan penuh saya coba mendesain modul Linux II ini akhirnya terselesaikan juga. Walaupun agak terlambat dari waktu yang ditetapkan tetapi saya terus berusaha menyelesaikan walaupun telat.

Saya meminta maaf karena buku ini selesai pada detik-detik terakhir. Dan dalam kesempatan ini saya ingin juga mengucapkan terima kasih kepada :

1. Allah SWT yang telah menganugerahkan akal dan pikiran ini, serta kekuatan fisik saya.
2. Kedua Orang tua dan Adik-adik saya yang telah memberikan dukungannya baik moral dan spiritual.
3. Bpk Ir. Naba Aji Notoseputro, selaku Direktur Bina Sarana Informatika.
4. Bpk Anton, S.Kom, selaku pimpinan saya serta KAJUR Teknik pada AMIK Bina Sarana Informatika.
5. Bpk/Ibu anggota komisi Tehnik yang tidak bisa saya sebutkan satu persatu.
6. Teman-teman Instruktur Lab Teknik baik yang di Salemba dan Fatmawati
7. Seluruh staff dan karyawan Bina Sarana Informatika khususnya cabang Fatmawati
8. Instruktur-instruktur Lab Linux AMIK Bina Sarana Informatika atas subangsihnya pada diri saya.

Besar harapan saya sebagai penulis modul ini adalah mudah-mudahan modul ini dapat sebagai langkah awal untuk mempelajari Linux. Dan mudahan-mudahan buku ini dapat berguna bagi para pembaca dan pengguna sekalian.

Penulis

Tim Komisi Tehnik BSI

DAFTAR ISI

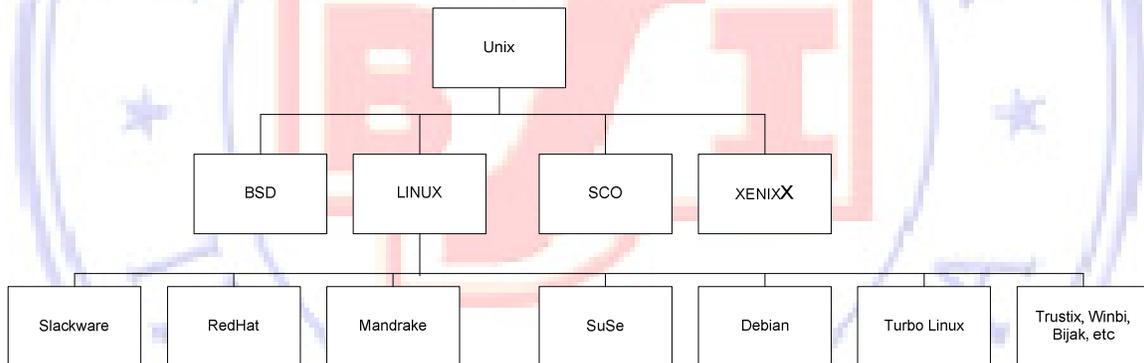
Kata Pengantar.....	2
Daftar Isi.....	3
BAB I : Pengenalan Linux	
1.1 Pengenalan Linux.....	5
1.2 Sejarah Linux.....	6
1.3 Susunan direktori linux.....	8
1.4 Shell Prompt.....	15
BAB II : Editor Vi	
2.1 Pattern matching pencarian string.....	22
2.2 Mengedit teks.....	24
2.3 Mengubah kepemilikan File.....	25
BAB III : IP Address	
3.1 Ip Address Kelas A.....	26
3.2 Ip Address kelas B.....	29
3.3 Ip Address kelas C.....	32
BAB IV : Secure Shell dan Telnet	
4.1 Ip Secure Shell (SSH).....	41
4.2 Telnet.....	42
BAB V : Pemaketan Data	
5.1 Pemaketan Data (TAR).....	44
BAB VI : Secure Copy	
6.1 Secure Copy (SCP).....	46
BAB VII : IPtables (Ip Chains)	
7.1 IP Tables (IP Chains).....	47
BAB VIII : Remote Dekstop dan Genomeeting	
8.1 Remote Dekstop.....	54
8.1 Genomeeting.....	55
BAB IX : sharing Data (samba)	57

PENGENALAN LINUX

1.1. Apakah Linux itu?

Linux merupakan sebuah sistem operasi dengan kemampuan multiuser dan multitasking yang bersifat free. Free disini bukan diartikan sebagai gratis tetapi berarti kebebasan seperti yang tertuang dalam lisensi yang dimiliki oleh Linux yaitu GNU GPL (GNU is Not Unix General Public License). Perangkat lunak yang mempunyai lisensi GNU GPL akan menjamin 3 (tiga) kebebasan bagi para penggunanya, yaitu :

- Kebebasan untuk menjalankan perangkat lunak tersebut dengan tujuan apapun.
- Kebebasan untuk mengubah perangkat lunak tersebut agar sesuai dengan kebutuhan. Hal ini mengharuskan prasyarat tersedianya source code perangkat lunak tersebut.
- Kebebasan untuk mendistribusikan salinannya, baik secara gratis maupun dengan biaya.



Gambar 1.1 Keluarga unix

Sebagai sistem dalam rumpun UNIX, GNU/Linux mewarisi sifat-sifat yang menyerupai UNIX misalnya multiuser, multitasking, line/text command based, secure, ready for network, development tool support. Dalam pengembangannya GNU/Linux sendiri dikembangkan terpisah misalnya kernel Linux dibuat oleh Linus Benedict Tolvalds pada tahun 1991, Donal Knuth membuat standar formatter (Tex), Bob Scheffler mengembangkan X Window dan sebagainya.

Pada dasarnya GNU/Linux adalah sebuah kernel, paket lainnya yang melengkapi misalnya aplikasi, compiler, window manager, redhat paket manager dan sebagainya

disebut sebagai distribusi, di mana saat ini distribusi GNU/Linux banyak sekali. Seperti yang telah disebutkan di atas bahwa Linux pada dasarnya adalah sebuah kernel, dimana kernel menjembatani antara user level hardware dan aplikasi-aplikasi yang menerjemahkan bahasa software sehingga mampu dipahami oleh hardware kemudian hardware memprosesnya sesuai dengan kebutuhan.

1.2. Sejarah Linux

Pembuat sekaligus pemegang hak cipta Linux adalah Linus Benedict Torvalds. Beliau pada saat itu adalah mahasiswa Universitas Helsinki, Finlandia yang terinspirasi oleh sebuah sistem operasi sejenis unix yang dinamakan Minix. Walaupun pada awalnya pembuatan Linux dianggap suatu hobi tetapi setelah dilepas ke internet ternyata mendapat tanggapan yang luar biasa dari programmer di seluruh dunia yang tergabung dalam suatu newsgroup. Selanjutnya Linux dikembangkan oleh Linus di bantu oleh programmer dari seluruh dunia melalui internet.

Nama Linux merupakan kombinasi unik antara nama penciptanya dan nama sistem operasi yang menjadi targetnya (UNIX). Semuanya berawal dari sebuah sistem operasi bernama Minix. Minix dibuat oleh Profesor Andrew Tanenbaum. Minix adalah sistem operasi mirip UNIX yang bekerja pada PC. Torvald adalah salah seorang mahasiswa di Universitas Helsinki yang menggunakan Minix. Walaupun cukup bagus, ia belum menganggap Minix memadai. Kemudian pada tahun 1991 ia membuat sistem operasi yang merupakan clone UNIX, yang diberi nama Linux. Seperti halnya Minix, Linux tidak menggunakan kode apa pun dari vendor UNIX komersial, sehingga Torvalds mendistribusikan linux di internet secara bebas dan gratis.

Pada Oktober 5 1991, Torvalds mengeposkan sistem operasinya di newsgroup comp.os.minix. Ia mengumumkan bahwa source code Linux tersedia dan meminta bantuan programmer-programmer lain untuk ikut mengembangkannya. Ketika itu Linux masih setengah matang, sistem operasi ini hanya bisa menjalankan sedikit perintah UNIX, seperti bash, gcc dan gnu-make. Saat Linux 1.0 diluncurkan pada 1994, sistem operasi ini telah cukup stabil dan memiliki banyak feature, seperti preemptive multitasking (kemampuan untuk membagi sumber daya CPU untuk banyak aplikasi) dan symmetric multiprocessing (kemampuan untuk membagi tugas di antara banyak CPU).

Linux bahkan memiliki maskotnya sendiri yang oleh torvalds dijeaskan sebagai “seekor penguin yang menggemaskan dan ramah, yang kekenyangan setelah makan banyak ikan hering”. Pada 1996, tim pengembangan Linux yang ada diseluruh dunia mulai memberikan hasilnya. Tahun itu mereka telah membuat versi Linux untuk sejumlah versi hardware, dari Atari ST sampai Macintosh. Linux terus berkembang pesat, utamanya karena ada sejumlah distributor (seperti RedHat, Caldera, dsb) yang berkompetisi untuk berebut pangsa pasar. Oleh karena itu dibentuk kelompok bernama Linux Standard Base. Kelompok ini bekerja untuk memastikan bahwa beragam distribusi Linux yang ada tetap bisa menjalankan aplikasi yang sama dan saling berinteroperasi. Saat ini ada tujuh distribusi Linux paling terkenal, yaitu :

1. RedHat Linux, distributor paling populer di AS dan salah satu yang paling mudah digunakan.
2. Mandrake Linux, distributor yang menambahkan update dan patch untuk RedHat Linux.
3. Caldera Open Linux, distibusi Linux dengan instalasi dan lingkungan pengguna berbasis grafis yang bagus.
4. Suse Linux, distribusi Linux paling populer di Eropa yang juga menyediakan perangkat instalasi dan panduan berbahasa Indonesia.
5. Slackware Linux.
6. Debian GNU/Linux.
7. TurboLinux, distribusi Linux paling populer di Asia yang menyediakan dukungan untuk set karakter khusus Asia.

Kemampuan LINUX

- Multiuser
- Multiprocessor
- Multitasking, memungkinkannya program-program berjalan bersamaan (background)
- MultiThreading, dapat menciptakan subproses dengan cara efisien
- Pemblokkan I/O, Pipe dan Filter
- Local dan Network File System
- Security, login dengan password, ownership, group
- X-Window System

1.3. Susunan Direktori Linux

Seperti yang sudah dijelaskan sebelumnya, Di dalam Linux, menggunakan filesystem dimana directory akan tersusun ke dalam hirarkis tunggal. Berbeda dengan struktur di windows dimana susunan directory ditentukan dengan kondisi partisi dan harddisk. Bentuk drive di windows merupakan representasi dari partisi dimana huruf C diberikan untuk partisi pertama di harddisk pertama. Di Linux tidak mengenal konsep drive. Berbagai macam partisi dapat di mount kedalam direktori yang berada di dalam struktur direktori linux. Di Windows, untuk menunjukkan lokasi sebuah file, kita menuliskan seperti ini;

```
|| D:\Folder\tempat\menyimpan\file.txt
```

Di lingkungan Linux, untuk menunjukkan sebuah lokasi file, berupa seperti berikut;

```
|| /Folder/tempat/menyimpan/file.txt
```

Dari kedua perbedaan diatas, kita dapat mudah melihat perbedaan penggunaan slash di Linux versus backslashes di Windows. Danjuga tidak adanya nama drive (C:, D:, E: dll). Pada saat komputer booting, 'partisi root' akan di mount di / dan seluruh file, direktori dan device akan di mount dibawah /. Perlu juga diingat bahwa penamaan file dan direktori di Linux adalah case-sensitive, yaitu huruf besar dan huruf kecil merupakan karakter yang berbeda. /Folder/tempat/file.txt beda dengan /folder/Tempat/file.TXT

Struktur direktori Linux mengikuti standart “Filesystem Hierarchy Structure (FHS)” yang di pegang oleh Free Standart Group walaupun kebanyakan distribusi memodifikasi standart tersebut.

/ (root)

Struktur direktori di Linux secara umum diawali dengan root filesystem “/” dan tentu juga merupakan root atau akar dari seluruh direktori global. Partisi dimana di letakkan / (root system) akan menjadi direktori sistem atau partisi pokok.

/boot

Direktori boot tesimpan file-file boot loader diantaranya grub atau lilo. Kernel, initrd dan system.map juga terletak didalam /boot. Jika system yang digunakan menggunakan partisi LVM ataupun partisi dalam jaringan. Maka ada baiknya dibuatkan partisi kecil tersendiri untuk meletakkan /boot di harddisk dengan filesystem konvensional. /boot ini

umumnya sangat jarang sekali berubah isinya, kecuali memang kita sering bermain-main dengan kernel.

/sys

Berisi informasi yang berkaitan dengan kernel, device dan firmware. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'system'.

/sbin

Berisi file-file biner yang esensinya untuk sistem dan mengendalikan sistem. File-file biner atau bisa dianggap aplikasi sistem ini jika dioperasikan secara tidak tepat bisa berpotensi merusak. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'super binary'

/bin

Berisi file-file binari atau aplikasi yang lebih umum dan dapat digunakan oleh semua user. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'binary'

/lib

Berisi file-file library atau pustaka dari semua aplikasi binari yang tersimpan dalam direktori /sbin dan /bin. Di direktori ini juga tersimpan berbagai macamlibrari yang digunakan untuk aplikasi lain. Konsep penggunaan librari bersama ini membuat aplikasi di linux dapat menghemat ukuran. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'library'

/dev

Merupakan pseudo filesystem, atau directory yang isinya sebenarnya bukan benar-benar berisi file. Isi dari /dev ini berkaitan dengan perangkat-perangkat yang terdapat pada system. Misalkan untuk informasi port serial, port printer, dapat di berlakukan seperti membaca file. Misalkan perangkat serial terletak di /dev/tty01, kemudian partisi dalam harddisk di sebut sebagai /dev/sda7. dan lain sebagainya. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'device'

/etc

Direktori /etc berisi file-file konfigurasi sistem. Mayoritas aplikasi dan layanan konfigurasinya tersimpan di direktori /etc termasuk diantaranya /etc/hosts, /etc/resolv.conf dan lain sebagainya. Di direktori /etc/init.d tersimpan konfigurasi

bagaimana sebuah layanan dijalankan. Di direktori `/etc/rc*.d` tersimpan konfigurasi untuk menentukan service yang dijalankan untuk tiap-tiap sesi init.

/home

Semua direktori home dari pengguna tersimpan di direktori ini dengan nama user masing-masing sebagai pengelompokannya. Untuk beberapa sistem linux yang di spesifikasikan untuk server, direktori pengguna masih dikelompokkan lagi kedalam `/home/users`. Di dalam direktori `/home/nama-user` tersimpan konfigurasi-konfigurasi yang spesifik terhadap user tersebut. Oleh karena itu, berbeda user, walaupun berada di sistem yang sama bisa mendapat lingkungan dan tampilan yang sama sekali berbeda. Direktori `/home` merupakan direktori yang paling 'dekat' dengan user. Direktori `/home` ini bisa berisi dari dokumen-dokumen pekerjaan user hingga file-file hiburan seperti mp3 dan film juga termasuk foto-foto yang dimiliki oleh user. Oleh karena itu, untuk membatasi agar file-file di pengguna tidak mendesak file-file system, sangat umum untuk meletakkan direktori `/home` di partisi yang terpisah. Hal ini dapat menahan file-file yang disimpan user hingga total ukuran tertentu tanpa mengganggu ruang gerak system.

/media

Merupakan direktori untuk menyimpan direktori-direktori mount point. CD-ROM, DVD, flash disk, bahkan floppy disk juga akan termount di direktori ini. Pada distro-distro modern, sudah memberikan fasilitas untuk menampilkan device-device yang dimount ke depan Desktop. Sehingga pengguna tidak perlu susah dan repot menuju ke `/media` untuk dapat mengakses flash disk-nya tapi cukup lihat ke desktop-nya dan masuk ke direktori yang terbuat baru di sana. Untuk workstation yang terintegrasi dengan jaringan, pada umumnya untuk melakukan mounting storage network juga diletakkan di `/media`. Dengan dikelompokkan seperti itu maka mudah untuk mengenali bahwa semua yang berada di dalam `/media` merupakan media penyimpanan.

/mnt

Pada Linux yang masih umum menggunakan kernel 2.4.x. Untuk tempat mengumpulkan mount point berada di `/mnt`. Dikarenakan pada sistem berbasis kernel 2.6.x sudah menggunakan `/media`, maka `/mnt` ini umumnya kosong. `/mnt` bisa juga dijadikan mount point pada saat system rescue atau troubleshooting. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'mount'

/opt

Direktori /opt saat ini jarang digunakan. Beberapa paket software terpisah menggunakan direktori untuk menyimpan paket yang menuju ke lokasi manapun. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'optional'

/usr

Sebuah sub-hirarki dari root filesistem di simpan didalam /usr. Didalam /usr tersimpan aplikasi dan utiliti yang spesifik dengan user. Jika kita melihat kedalam direktori /usr maka kita juga akan menemukan direktori yang mirip dengan di /yaitu bin, sbin dan lib. Hanya saja, aplikasi dan librari yang terletak /usr tidak terlalu kritikal untuk sistem. Untuk istilah mudahnya, /usr merupakan tempat dimana user menginstall aplikasi sendiri yang bukan official dari distro. Misalkan menginstall melalui tar-ball, atau paket yang dibuat sendiri. Jika pengguna termasuk orang yang sering menambah-nambah aplikasi sendiri diluar bawaan paket yang disediakan untuk distro itu, maka direktori /usr sudah dipastikan akan cepat sekali membengkak. Ada baiknya untuk sistem yang penggunaanya seperti itu, /usr di berikan partisi sendiri. Untuk aplikasi yang bisa langsung dijalankan, sistem linux akan membaca secara bersamaan yang ada di /bin dan di /usr/bin begitu juga untuk /sbin dan /usr/sbin. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'user'

/usr/share

Merupakan tempat didalam /usr yang digunakan untuk menyimpan data-data yang bisa dibagikan dan tidak terikat dengan platform. Misalnya seperti wallpaper yang bisa dan boleh digunakan oleh semua user akan diletakkan didalam /usr/share. Lalu ada juga fonts, dan sound theme yang berkaitan dengan tampilan.

/usr/doc

Merupakan tempat untuk menyimpan dokumentasi dan catatan yang berkaitan dengan aplikasi. Apabila aplikasi third-party yang digunakan merupakan aplikasi yang dikembangkan dengan baik, maka tentunya juga menyediakan file dokumentasi yang dapat dibaca di dalam /usr/doc

/usr/src

Merupakan tempat untuk menyimpan source code dari aplikasi sistem. Yang paling umum tersimpan disini adalah source code dari kernel linux. Source code ini sangat

bermanfaat untuk melakukan kompilasi ulang atau melakukan optimasi di tingkat kernel dengan dasar kernel sebelumnya.

`/usr/include`

Di direktori `/usr/include` tersimpan file-file header dari compiler C. File header ini mendefinisikan struktur dan konstanta yang dibutuhkan untuk membangun sebuah aplikasi yang standart. Direktori didalam `/usr/include` tersimpan header untuk compiler C++.

`/usr/X11R6`

Menyimpan sistem X-Window dan hal-hal yang berkaitan dengan X-Window. Subdirectories dibawah `/usr/X11R6` tersimpan binari X itu sendiri dan juga dokumentasi, file header, config. icon. sounf, dan sebagainya yang berkaitan dengan grafis.

`/usr/local`

DI disini tersimpan aplikasi yang terinstall dan file yang yang digunakan di local machine. Jika komputer yang digunakan merupakan bagian dari sebuah jaringan besar, terus direktori `/usr` lokasi fisiknya terletak di komputer yang berbeda dan dibagikan kedalam jaringan untuk di mount kedalam `/usr`. Pada jaringan seperti ini, direktori `/usr/local` akan berisi barang-barang yang hendaknya tidak digunakan di banyak mesin dan hanya di gunakan di local machine saja. Karena kebanyakan komputer tidak memanfaatkan bentuk jaringan seperti yang disebutkan tadi, bukan berarti `/usr/local` menjadi tidak berguna. Jika kita menemukan aplikasi yang menarik dan secara official tidak tersedia dan bukan bagian dari distro yang digunakan, hendaknya kita menginstallnya kedalam `/usr/local`. Sebagai contoh, jika aplikasi tambahan yang umum akan tersimpan kedalam `/usr/bin`, maka aplikasi tambahan yang sifatnya lebih custom hendaknya di simpan di `/usr/local/bin`. Dengan cara ini maka dapat menghindari kebingungan dalam jenis aplikasi yang tersedia dan menjaga sistem tetap bersih dan rapi.

`/root`

Merupakan direktori home-nya superuser (root). Harap jangan bingung dengan direktori root (`/`). Walaupun cara menyebutnya sama, tapi sama sekali berfungsi sangat berbeda.

`/var`

Direktori `/var` merupakan direktori yang isinya sangat dinamis. Jika digunakan didalam server. Sangat dianjurkan `/var` ini untuk diletakkan di partisi terpisahdikarenan direktori

/var dapat membengkak dengan sangat cepat. Selain itu juga, dengan direktori /var dibuat partisinya sendiri atau secara fisik ditentukan lokasinya, maka dapat mencegah internal fragmentasi, dan proses pencarian file tidak terlalu jauh, hanya seputar cylinder itu-itu saja. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'variative'

/var/log

Merupakan direktori untuk menyimpan berbagai macam log atau catatan yang berkaitan dengan sistem. Isi dari /var/log ini terus terupdate selama sistem berjalan. Oleh karena itu, /var/log merupakan alasan dan penyebab utama direktori /var dapat membengkak dengan gila-gilaan. Walaupun di linux sendiri tersedia aplikasi logrotate untuk meredam percepatan pembengkakan direktori /var, tapi tetap juga jika tidak disiapkan ruang tersendiri, ukuran log ini dapat menghantam ruang kosong di sistem. Direktori /var/log ini hendaknya menjadi tempat pertama kita lari apabila di sistem terjadi keanehan. System log linux baik kok, keanehan pun akan di tuliskan dengan gambling kedalam log. Misalkan pada webserver, /var/log/httpd/access.log akan mencatat siapa saja yang mengakses web yang terpasang dan menuju ke mana. Apabila banyak sekali yang mengunjungi web itu, maka file /var/log/httpd/access.log pun akan terupdate dengan kecepatan luar biasa yang susah untuk diikuti oleh pandangan manusia biasa. Misalkan juga /var/log/syslog, disana tersimpan log yang berkaitan dengan system. Misalkan kita colokkan flash disk, maka dengan mengamati /var/log/syslog kita bisa mengetahui apakah sistem bisa mengenali flashdisk itu dan melakukan auto mount, atau memberikan pesan yang lain. Dari /var/log/syslog juga misalnya kita bisa melihat apakah rule crontab yang kita buat berjalan sesuai waktunya atau tidak. Semakin banyak kejadian baik yang buruk atau yang baik terjadi di system, maka /var/log pun akan semakin membengkak.

/var/mail

Merupakan direktori untuk menyimpan email masuk dan keluar. Direktori /var/mail ini akan sangat terasa fungsinya jika kita membuat mail server menggunakan postfix, sendmail atau qmail. Sebenarnya tanpa menggunakan mailserver itu, kita juga dapat mengirim email, hanya saja terbatas hanya dalam lingkup mesin kita dan user yang terdapat didalamnya. Apabila mesin yang kita gunakan hanya kita sendiri usernya, tentu saja tidak seru, maka perlu dibuatkan mail server yang sesungguhnya agar aktivitas kirim-mengirim email dapat seheboh dengan mail.yahoo.com ataupun mail.ugm.ac.id.

Direktori `/var/mail` merupakan penyebab kedua yang membuat direktori `/var` dapat membengkak tidak karuan. Aktivitas dan lalu lintas saling kirim email yang tinggi, ditambah lagi dengan attachment yang tidak masuk akal ukurannya, menyebabkan `/var/mail` bertanggung jawab penuh atas membengkaknya `/var`. Dengan di berikannya direktori `/var` sebuah partisi tersendiri, maka apabila direktori `/var` mentok ke total ukuran partisi, 'hanya' menyebabkan mailserver-nya macet. tapi sistem linux itu sendiri masih bisa berjalan dengan nyaman untuk mengatasi 'kekacauan' di `/var`. Ada juga cara agar direktori `/var` tidak segera penuh, dengan menerapkan quota di mail server.

`/var/spool`

Direktori `/var/spool` merupakan direktori untuk menyimpan file-file yang spooling atau yang sedang mengantri untuk diproses. Yang umum berada didalam `/var/spool` ini adalah operasi printing. Apabila linux yang dibangun difungsikan sebagai printer server, atau sharing printer, maka `/var/spool` ini juga perlu diperhatikan apabila yang menggunakan printer tersebut orangnya sangat banyak.

`/var/run`

Di direktori ini tersimpan PID (Process ID) dari layanan-layanan yang berjalan. PID yang tercatat didalam `/var/run` ini nantinya akan digunakan oleh script didalam `/etc/init.d/*` untuk mengendalikan layanan yang tersedia di linux.

`/proc`

Direktori `/proc` juga merupakan pseudo filesystem yang mirip dengan `/dev`. Bedanya, `/proc` ini murni hanya berkaitan dengan sistem dan tidak menyangkut pada device. Jika kita melakukan pengecekan ukuran penggunaan space, jangan terkecoh dengan direktori `/proc` yang tercatat menunjukkan memakan space sangat besar. Segala macam space yang tertulis disitu sama sekali tidak terdapat di dalam media penyimpanan harddisk. Jadi direktori `/proc` sebenarnya tidak ada samasekali. Isi dari `/proc` ini adalah informasi dari sistem, tetapi jika kita mengubah isi informasi dari beberapa file `/proc` juga dapat mengubah sifat jalannya sistem. Jangan takut apabila sistem berubah menjadi aneh tatkala kita mengubah isi `/proc`. Selama perubahan isi hanya menggunakan echo, maka dengan merestart system, maka isi `/proc` pun akan langsung kembali ke semula. Misalkan kita ingin melakukan forwarding network, maka informasi pada `/proc/sys/net/ipv4/ip_forward` yang semula berisi 0 tinggal di ubah menjadi 1. Misalkan

kita melihat informasi processor, maka kita dapat melihat dalam `/proc/cpuinfo`. Tapi jangan harap dengan mengedit `/proc/cpuinfo` maka kita akan mendapat cpu yang bekerja dengan clock cycle yang lebih menakjubkan.

`/tmp`

Merupakan file sistem yang menyimpan file-file sementara. Beberapa distro akan otomatis membersihkan isi dari `/tmp` sewaktu reboot. Direktori `/tmp` memiliki mode yang sangat terbuka sehingga mudah untuk ditulisi oleh siapa saja. Didalam `/var/tmp` juga digunakan sebagai penyimpanan file-file sementara, bedanya `/var/tmp` dengan `/tmp` yaitu `/var/tmp` tidak akan dibersihkan saat system reboot. Untuk memudahkan mengingat, direktori ini dianggap kependekan dari 'temporary'.

`/lost+found`

Di direktori ini linux menyimpan file-file yang berhasil di recover saat sistem crash. Dengan melihat kedalam `/lost+found` mungkin kita sapat menemukan file yang hilang.

1.4. Shell Prompt

Shell berfungsi sebagai *command interpreter*, yaitu menterjemahkan perintah yang diberikan dari prompt dan juga dapat digunakan sebagai bahasa pemrograman, yaitu melaksanakan sekumpulan perintah yang dibaca dari suatu file. Sebagian pengguna Linux masih banyak yang menyukai bekerja di lingkungan shell prompt dikarenakan lebih cepat dibandingkan bekerja di lingkungan grafis (GUI).

A. Shell Bash

Shell default di sistem Linux adalah bash yang merupakan singkatan dari Bourne Again Shell. Prompt untuk shell bash ditandai dengan simbol `$`. Shell bash menggabungkan fitur-fitur dari shell yang ada di sistem Unix yaitu sh, csh dan ksh. Fitur-fitur yang dimiliki oleh shell bash diantaranya adalah :

1. alias

alias biasanya digunakan untuk menyingkat suatu perintah yang panjang. Sintaks :
alias nama_alias=perintah

Contoh :

```
$alias cls=clear  
$alias ls="ls -l"
```

2. history

Anda dapat menampilkan kembali perintah-perintah yang sudah diketikkan dari prompt dengan cara menekan tombol panah atas atau tombol panah bawah.

3. command line editing

Apabila anda salah mengetikkan suatu perintah maka perintah tersebut masih dapat diedit dengan menggunakan tombol panah kiri atau tombol panah kanan.

4. command line completion

Fungsi ini berguna untuk mempermudah pengetikan perintah yang panjang. Caranya dengan menekan tombol **[TAB]** pada saat mengetikkan perintah.

File startup adalah file-file yang akan dieksekusi ketika shell bash dijalankan. File-file itu adalah:

/etc/profile : File ini akan dieksekusi pada urutan pertama. Isinya sebagian besar berisi pendefinisian variabel lingkungan yang berlaku global untuk seluruh sistem.

~/.bash_profile : File ini berada di direktori home user. Fungsinya sama dengan file */etc/profile* tetapi hanya berlaku untuk user yang bersangkutan.

~/.bashrc : File ini berada di direktori home user. Isinya berupa pendefinisian alias dan fungsi.

~/.bash_logout : File ini berada pada direktori home user. Isi file ini akan dieksekusi pada saat keluar (logout) dari shell.

B. Perintah Shell

Sintaks Perintah secara umum yaitu:

perintah [-pilihan...] [argumen...]

Keterangan :

- antara perintah dan pilihan harus ada spasi.
- antara pilihan dan argumen harus ada spasi.
- tanda kurung siku ([]) berarti bahwa pilihan dan argumen tidak harus selalu ada.
- tanda titik tiga (...) berarti bahwa pilihan dan argumen bisa lebih dari satu.
- setiap perintah di Linux membedakan huruf besar dan kecil (*case sensitive*).
- setiap pengetikan perintah selalu diakhiri dengan menekan tombol **[Enter]**.

1. **ls - list**

Fungsi : untuk menampilkan isi sebuah direktori.

Pilihan yang sering digunakan :

- a menampilkan seluruh isi direktori termasuk file yang hidden
- d menampilkan atribut direktori bukan isinya
- l menampilkan isi direktori lengkap dengan atributnya
- R menampilkan isi subdirektori

```
$ ls
$ ls -l
$ ls -la
```

2. **User dan Password**

Fungsi untuk membuat user baru dan memberikan password terhadap user tersebut

```
[root@localhost~]# adduser Nea-Budiarti
[root@localhost~]# passwd Nea-Budiarti
Changing password for Nea-Budiarti
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
passwd: all authentications tokens updated
successfully
```

Catatan : Password yang anda ketikkan tidak akan ditampilkan pada layar.

3. More

Fungsi : sebagai pager, menampilkan output per layar

Navigasi tombol pada perintah more :

[Enter]	scroll down per baris
Space Bar	scroll down per layar
q	keluar dari more

```
$ ls -l /etc | more
```

4. Less

Fungsi : sama dengan more

Navigasi tombol pada perintah less :

Panah Bawah	scroll down per baris
Panah Atas	scroll up per baris
[Enter]	scroll down per baris
Space Bar	scroll down per layar
q	keluar dari less

```
$ ls -l /etc | less
```

5. man - manual page

Fungsi : menampilkan dokumentasi atau manual suatu perintah

```
$ man ls
```

6. pwd - print working directory

Fungsi : menampilkan posisi direktori kerja saat ini

```
$ pwd  
/home/Nea-Budiarti
```

7. cd - change directory

Fungsi : untuk melakukan pindah direktori

Catatan :

Jika perintah cd dijalankan tanpa argumen, maka akan mengembalikan posisi direktori ke direktori home user.

```
$ cd /etc  
$ pwd  
/etc
```

```
$ cd  
$ pwd  
/home/Nea-Budiarti
```

8. **mkdir - make directory**

Fungsi : membuat direktori

```
$ mkdir data
```

9. **cp - copy**

Fungsi : membuat salinan suatu file

```
$ cp latih coba
```

10. **mv - move**

Fungsi :

1. merubah nama file (rename)

```
$ mv coba uji
```

2. memindahkan file ke direktori

Catatan : argumen kedua harus berupa suatu direktori

```
$ mv latih data
```

11. **rm - remove**

Fungsi :

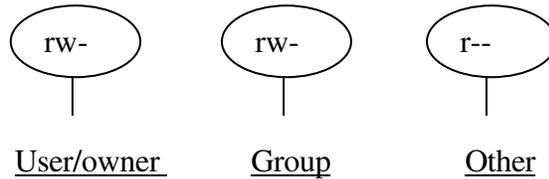
1. menghapus file

```
$ rm uji
```

2. menghapus direktori

```
$ rm -r data
```

12. Permission File / Hak Izin Akses File



Tiga digit pertama digunakan untuk mengatur izin akses bagi owner. Tiga digit kedua digunakan untuk mengatur izin akses bagi user lain yang masih satu group dengan owner sedangkan tiga digit ketiga digunakan untuk mengatur izin akses bagi user lain diluar group owner.

Ijin Akses untuk File

R : read : ijin untuk membaca/melihat isi suatu file
w : write : ijin untuk mengubah isi suatu file
x : execute : ijin untuk mengeksekusi/menjalankan suatu file
dimana jenis file berupa script/program

Ijin Akses untuk Direktori

r : read : ijin untuk melihat isi suatu direktori (contoh: \$ ls -l data)
w : write : ijin untuk membuat/menghapus file dalam direktori
(contoh: \$ rm data/latih)
x : execute : ijin untuk pindah ke dalam direktori (contoh: \$ cd data)

Merubah Permission File - chmod

Perintah chmod memiliki dua format :

1. Format Huruf

Sintaks : **chmod [ugo] [+|=] [rwx] file(s)**

Keterangan :

u : user
g : group
o : other

- + : menambahkan permission
- : menghilangkan permission
- = : set

2. Format Angka

Sintaks : **chmod [kode_oktal] file(s)**

read (r) = 4

write(w) = 2

execute(x) = 1

contoh :

rw-	rw-	r--
420	420	400
6	6	4

File Creation Mask - **umask**

Setiap kali file/direktori dibuat, maka permission-nya sesuai dengan *mask* yang ditentukan. Khusus untuk file, *mask* untuk x tidak ada

Sintaks : **umask [kode_oktal]**

Kode oktal untuk File :

0	=	rw-	000	4	=	-w-	100
1	=	rw-	001	5	=	-w-	101
2	=	r--	010	6	=	---	110
3	=	r--	011	7	=	---	111

Keterangan :

Untuk file *mask* untuk x tidak ada.

Kode oktal untuk Direktori :

0	=	rwX	000	4	=	-wX	100
1	=	rw-	001	5	=	-w-	101
2	=	r-x	010	6	=	--x	110
3	=	r--	011	7	=	---	111

```
$ umask 077
$ touch latih
$ ls -l latih
-rw----- 1 misstux          nea    24343    Dec  03  09:44
$ mkdir data
$ ls -ld data
drw----- 1 misstux          nea    24343    Dec  03  09:44
```

13. Backup Data

backup=menduplikate. Sama halnya dengan perintah “cp”. Namun tidak semua file system di LINUX dapat menggunakan perintah “cp”.

File backupan memiliki link (hubungan). Jadi, jika file master diubah maka file backupan juga ikut berubah. Berbeda dengan kopi. Jika kita mengubah file master, maka file kopian tidak akan ikut berubah.

Backup ada 2 macam,

a. Hardlink

file dapat di backup ke direktori manapun

jika file master dihapus, maka file backupan masih dapat dibuka dan dibaca.

```
ln nama_file alamat_dir_tujuan
```

b. Softlink

file hanya dapat di backup di tempat (di direktori yang sama)

jika file master dihapus, maka file backupan tidak dapat dibaca.

```
ln -s nama_file alamat_dir_tujuan
```

14. Perintah lanjutan

Perintah echo sama dengan perintah cat. Bedanya jika memakai perintah echo, file akan tersimpan secara otomatis tanpa harus (CTRL+C) / menyimpan

```
echo > nama_file "isi file"
echo >> nama_file "isi file"
```

Untuk membuat file

Untuk menambah file

menampilkan baris awal suatu file

```
head -n nama_file
```

menampilkan baris akhir suatu file

```
tail -n nama_file
```

mengompres file=perintah untuk memaketkan file. Mengompres ukuran memori file agar menjadi lebih kecil.

```
gzip nama_file
```

untuk membaca file kompresan menggunakan

```
zcat nama_file
```

mengextraks file= perintah untuk menormalkan perintah gzip

```
gunzip nama_file
```

membuat perintah baru

```
alias perintah_baru='perintah_lama'
```

menormalkan kembali perintah

```
unalias perintah_baru
```

menghitung jumlah baris, kata dan karakter suatu file wc(words character)

```
wc nama_file
```

Editor Teks vi

1. Pattern Matching dalam Pencarian String

Pola	Arti	contoh	keterangan
.	match 1 karakter	/a./	mencari karakter huruf p
^	match awal baris	/^aku/	mencari kata aku diawal baris
\$	match akhir baris	/aku\$/	mencari kara aku diakhir baris
\<	Match awal kata	/\<i/	mencari kata yang diawali "i"
\>	Match akhir kata	/u\>/	Mencari kata yang diakhiri"u".
[...]	Match pilihan	/[0-9]	Mencari numeric
*	Match setiap karakter (termasuk karakter kosong)	/ada*	Mencari kata yang ada string "ad" (jadwal, ada, pada, adalah, adapun)

2. Mengedit Teks dengan Menggunakan Perintah Baris Akhir

Untuk melakukan perintah baris akhir anda harus ada dalam modus instruksi dan selalu diawali dengan mengetikkan karakter titik dua (:). Contoh-contoh penggunaannya antara lain :

<code>:s/linux/LINUX/</code>	Mengganti sebuah kata linux dengan LINUX pada posisi kursor.
<code>:s/linux/LINUX/g</code>	Mengganti semua kata linux dengan LINUX pada posisi kursor .
<code>:1,\$ s/linux/LINUX/g</code>	Mengganti semua kata linux dengan LINUX di seluruh dokumen
<code>:s/[Ll]nux/LINUX/</code>	Mengganti sebuah kata Linux atau linux dengan LINUX pada posisi kursor.
<code>:1,3w test.txt</code>	Menyimpan baris 1 sampai 3 ke file test.txt
<code>:2 r test.txt</code>	Menyisipkan isi file test.txt di bawah baris ke 2
<code>:1,\$co \$</code>	Meng-copy semua isi file ke baris terakhir
<code>:4m 1</code>	Memindahkan baris ke 4 di bawah baris pertama.
<code>:/^\$/d</code>	Menghapus baris kosong
<code>:g/LINUX/d</code>	Menghapus seluruh baris yang mengandung kata LINUX
<code>:1,\$ s/linux//g</code>	Menghapus seluruh kata linux
<code>:1,\$ s/[^0-9]//g</code>	Menghapus seluruh alphabet

:1,\$ s/[A-Z]/\l&/g	Mengganti semua huruf besar menjadi huruf kecil
:1,\$ s/[a-z]/\u&/g	Mengganti semua huruf kecil menjadi huruf besar
:1,\$ s/^/>>>/g	Menyisipkan >> pada setiap awal baris
:1,\$ s/\$/<</g	Menyisipkan << pada setiap akhir baris

Mengubah kepemilikan File (Change Owner)

Perintah `chown` digunakan untuk mengganti pemilik sebuah file, perintah ini hanya dapat digunakan oleh user `root`. Perintah ini hanya dapat digunakan oleh user `root`. Perintahnya adalah sebagai berikut:

```
# chown namauser.namagrup namafile
# chown user.user coba.txt
```

perintah `chown` juga dapat digunakan dengan menggunakan parameter `R`, contohnya adalah sebagai berikut:

```
# chown -R user.group path_asal_direktori
```

Perintah-perintah yang digunakan:

Perintah Dasar	Maksud dari Perintah
<code>-g</code>	Masuk menjadi satu group di dalam user account
<code>chmod</code>	Mengubah izin akses suatu file atau direktori
<code>ls -l</code>	Menampilkan file-file yang terdapat pada direktori beserta izin akses suatu filenya
<code>Vi</code>	Membuat atau mengubah isi sebuah file dalam konsol/terminal
<code>I</code>	Untuk (insert) melakukan penginputan di dalam editor vi
<code>Esc</code>	Untuk menghentikan (insert) atau pengeditan file di dalam editor vi
<code>:wq</code>	Untuk menyimpan dan keluar file di dalam editor vi
<code>Chown</code>	Mengubah kepemilikan sebuah file atau direktori

IP ADDRESS

TCP/IP bukanlah sebuah protocol tunggal tetapi satu kesatuan protocol dan utility. Protocol ini dikembangkan oleh ARPA (Advance Research Project Agency) untuk departemen pertahanan Amerika pada tahun 1969. Sebuah alamat TCP/IP adalah biner berukuran 32 bit yang diberikan kesetiap host dalam sebuah jaringan. Nilai ini digunakan untuk mengenali jaringan dimana host tersebut mengenali nomor host bersangkutan di jaringan tertentu. Nilai tersebut terbagi menjadi empat bagian nomor delapan bit yang disebut octet. Setiap alamat terbagi atas dua komponen :

IP Address kelas A

Network ID

Network ID adalah bagian dari alamat IP yang mewakili jaringan fisik dari host. Setiap komputer dalam segment jaringan tertentu akan memiliki ID jaringan yang sama. Network ID mempunyai ketentuan-ketentuan, yaitu :

1. Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 00000000 atau bernilai 0 (nol).
2. Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 11111111 atau bernilai 255.
3. Sebuah byte Net ID range byte pertamanya itu adalah : 1 sampai 223 kecuali angka 127.
4. Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 01111111 atau bernilai 127. Karena IP ini di jadikan sebagai loopback dari sebuah sistem protocol TCP/IP.

Host ID

Host ID adalah bagian yang mewakili bagian individu dari alamat, bila komputer di segment jaringan anda memiliki alamat, maka jaringan perlu tahu milik siapakah suatu paket data tersebut. Berikut ini adalah tabel kelas-kelas IP dan host yang didukungnya :

Tabel II.4 Internet Protocol Class

Class	Jumlah Host	Jangkauan Octet Pertama
-------	-------------	-------------------------

A	16.777.216	1 – 126
---	------------	---------

Host ID mempunyai ketentuan-ketentuan, yaitu :

1. Semua byte Host ID itu tidak boleh semua bytenya : 192.168.2.00000000. atau bernilai 0 (nol).
2. Semua byte Host ID itu tidak boleh semua bytenya : 11111111 atau bernilai 255.

Class A	Network	Host		
Octet	W	X	Y	Z

Seperti yang disebutkan diatas bahwa IP adalah 32 bit. Nilai tersebut terbagi menjadi empat bagian nomor delapan bit yang disebut octet. Contoh alamat IP adalah 202.149.240.66. Dengan menggunakan contoh diatas katakanlah bahwa administrator mensetup jaringan dengan semua komputer memiliki bagian nilai yang sama : 202.149.240.xxx ini adalah network ID, sedangkan nomor pada xxx adalah node Host ID nya.

Alasan yang mendasari pembagiannya atau pengelompokkan *IP Address* ini adalah untuk mempermudah pendistribusian pendaftaran *IP Address*. Adapun pembagian kelas tersebut adalah sebagai berikut :

IP Address kelas A diberikan untuk jaringan dengan jumlah *Host* yang sangat besar. Bit pertama dari *IP Address* kelas A selalu di set 0 (nol) sehingga Byte pertama terdepan dari *IP Address* kelas A selalu bernilai antara angka 0 – 127.

Pada kelas ini, *Network ID* adalah 8-bit pertama sedangkan untuk *Hostnya* adalah 24-bit berikutnya. Sebagai contoh jika *IP Address* kelas A adalah 11.147.5.5 maka dapat dikatakan *Network ID* tersebut adalah 11 dan *Host ID* dari *IP address* tersebut adalah 147.5.5. *IP address* kelas A ini dapat menampung lebih kurang 16 juta Host. Berikut adalah karakteristik dar *IP address* kales A.

Karakteristik *IP address* kelas A

Format	: 0nnnnnnnn hhhhhhhh hhhhhhhh hhhhhhhh
Bit Pertama	: 0
Panjang NetID	: 8 bit
Panjang HostID	: 24 bit
Byte Pertama	: 1 – 126
Jumlah	: 126 kelas A
Range IP	: 1.xxx.xxx.xxx sampai 126.xxx.xxx.xxx
Jumlah IP	: 16.777.214 <i>IP Address</i> pada setiap kelas A

Setiap alamat TCP/IP jatuh pada kelas satu alamat. Kelas mewakili sebuah group alamat yang dapat segera dikenali komponen software sebagai bagian dari sebuah jaringan fisik. Misalnya alamat TCP/IP berikut dan nilai binernya :10.149.240.66 alamat binernya 00001010.10010101.11110000.10000010

Penggunaan subnet mask sangat ditentukan dengan banyaknya komputer atau host yang digunakan pada sebuah jaringan. Setiap komputer disebuah jaringan biasanya ingin mengirim data langsung ke komputer lainnya. Komputer pengirim harus memastikan bahwa si penerima berada pada jaringan yang sama atau diluar itu. Subnet mask digunakan oleh protocol stack TCP/IP untuk menentukan bahwa host yang akan dicoba dikomunikasikan berada di jaringan yang sama atau berada di diluar jaringan..

Berikut ini adalah table klasifikasi dari subnet mask :

Tabel II.5 Subnet Mask Class

Class	Subnet Mask
A	255.0.0.0

Subnet mask digunakan untuk membedakan antara network ID dan host ID dan untuk menunjukkan suatu host apakah berada pada jaringan local atau non local. Untuk jaringan non local berarti harus mentransmisi data melalui sebuah router.

Penggunaan subnet mask sangat ditentukan oleh banyaknya komputer atau host pada sebuah jaringan. Salah satu standar dalam protokol jaringan yang di kembangkan oleh ISO (*International Standard Organization*) adalah model referensi OSI (*Open System Interconnection*) merupakan suatu sistem yang terbuka untuk berkomunikasi dengan sistem-sistem yang lain.

IP Address kelas B

Network ID

Network ID adalah bagian dari alamat IP yang mewakili jaringan fisik dari host. Setiap komputer dalam segment jaringan tertentu akan memiliki ID jaringan yang sama. Network ID mempunyai ketentuan-ketentuan, yaitu :

1. Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 00000000 atau bernilai 0 (nol).
2. Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 11111111 atau bernilai 255.
3. Sebuah byte Net ID range byte pertamanya itu adalah : 1 sampai 223 kecuali angka 127.

Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 01111111 atau bernilai 127. Karena IP ini di jadikan sebagai loopback dari sebuah sistem protocol TCP/IP

Host ID

Host ID adalah bagian yang mewakili bagian individu dari alamat, bila komputer di segment jaringan anda memiliki alamat, maka jaringan perlu tahu milik siapakah suatu paket data tersebut. Berikut ini adalah tabel kelas-kelas IP dan host yang didukungnya :

Tabel II.4 Internet Protocol Class

Class	Jumlah Host	Jangkauan Octet Pertama
B	16.536	128 –191

Host ID mempunyai ketentuan-ketentuan, yaitu :

1. Semua byte Host ID itu tidak boleh semua bytenya : 192.168.2.00000000. atau bernilai 0 (nol).

2. Semua byte Host ID itu tidak boleh semua bytenya : 11111111 atau bernilai 255.

Class B	Network		Host	
Octet	W	X	Y	Z

Seperti yang disebutkan diatas bahwa IP adalah 32 bit. Nilai tersebut terbagi menjadi empat bagian nomor delapan bit yang disebut octet. Contoh alamat IP adalah 172.1168.240.66. Dengan menggunakan contoh diatas katakanlah bahwa administrator mensetup jaringan dengan semua komputer memiliki bagian nilai yang sama : 172.168.xxx.xxx ini adalah network ID, sedangkan nomor pada xxx.xxx.240.66 adalah host ID nya.

IP Address kelas B digunakan untuk jaringan yang berukuran sedang dan besar. *IP Address* kelas B ini 2 (bit) bit pertama dari IP selalu di set dengan 1 0 (satu nol) sehingga bit terdepan dari *IP Address* selalu bernilai 128 sampai 191.

IP Address kelas B dapat menampung lebih kurang 65000 *Host*. Berikut adalah karakteristik dari *IP Address* kelas B.

Karakteristik *IP Address* kelas B

Format	: 10nnnnnnnn nnnnnnnn hhhhhhhh hhhhhhhh
Bit Pertama	: 10
Panjang NetID	: 16
Panjang HostID	: 16 bit
Byte Pertama	: 128 – 191
Jumlah	: 16.384 kelas B
Range IP	: 128.0.xxx.xxx sampai 191.255.xxx.xxx
Jumlah IP	: 65.532 <i>IP Address</i> pada setiap kelas B

Setiap jaringan selalu memerlukan subnet yang biasa disebut subnet mask. Subnet merupakan bagian dari jaringan yang dimaksudkan untuk memecah network ID menjadi beberapa bagian kecil. Tujuannya agar beban kerja jaringan tidak terlalu berat. Subnet ini

terdiri dari angka 32 bit misalnya :: 11111111.11111111.11111111.00000000 atau 255.255.255.0. Pada subnet mask seluruh bit yang berhubungan dengan host ID diset dengan angka 0.

Jadi gambaran kurang lebih seperti ini :

255		255		0		0	
1 1 1 1 1 1 1 1		1 1 1 1 1 1 1 1		0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0	
128 64 32 16 8 4 2 1		128 64 32 16 8 4 2 1		0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0	

Subnet diatas menandakan 16 bit. Jadi setiap segment IP Address atau 1 byte IP Address itu terdiri dari 8 bit, subnet diatas menunjukkan bahwa ia menggunakan 2 segment IPAddress atau 2 byte.

Untuk penghitungan konversinya dari desimal ke biner adalah sebagai berikut :

Misal : 172. konevrsikan ke biner

$$\frac{172}{2} = \frac{86}{2} = \frac{43}{2} = \frac{21}{2} = \frac{10}{2} = \frac{5}{2} = \frac{2}{2} = 1$$

$$0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0$$

Jadi dibalik hasil pembagian diatas, menghasilkan : 127 = 10101100

Pembuktiannya :

127							
1 0 1 0 1 1 0 0							
128 64 32 16 8 4 2 1							

$$(128 \times 1) + (64 \times 0) + (32 \times 0) + (16 \times 0) + (8 \times 1) + (4 \times 1) + (2 \times 0) + (1 \times 0) = 127$$

Penggunaan subnet mask sangat ditentukan dengan banyaknya komputer atau host yang digunakan pada sebuah jaringan. Setiap komputer disebuah jaringan biasanya ingin mengirim data langsung ke komputer lainnya. Komputer pengirim harus memastikan bahwa si penerima berada pada jaringan yang sama atau diluar itu. Subnet mask digunakan oleh protocol stack TCP/IP untuk menentukan bahwa host yang akan dicoba dikomunikasikan berada di jaringan yang sama atau berada di diluar jaringan..

Berikut ini adalah table klasifikasi dari subnet mask :

Tabel II.5 Subnet Mask Class

Class	Subnet Mask
-------	-------------

B	255.255.0.0
---	-------------

Subnet mask digunakan untuk membedakan antara network ID dan host ID dan untuk menunjukkan suatu host apakah berada pada jaringan local atau non local. Untuk jaringan non local berarti harus mentransmisi data melalui sebuah router.

Penggunaan subnet mask sangat ditentukan oleh banyaknya komputer atau host pada sebuah jaringan. Salah satu standar dalam protokol jaringan yang di kembangkan oleh ISO (*International Standard Organization*) adalah model referensi OSI (*Open System Interconnection*) merupakan suatu sistem yang terbuka untuk berkomunikasi dengan sistem-sistem yang lain.

IP Address kelas C

Network ID

Network ID adalah bagian dari alamat IP yang mewakili jaringan fisik dari host. Setiap komputer dalam segment jaringan tertentu akan memiliki ID jaringan yang sama. Network ID mempunyai ketentuan-ketentuan, yaitu :

1. Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 00000000 atau bernilai 0 (nol).
2. Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 11111111 atau bernilai 255.
3. Sebuah byte Net ID range byte pertamanya itu adalah : 1 sampai 223 kecuali angka 127. Sebuah byte Net ID itu tidak boleh byte pertamanya itu berupa : 01111111 atau bernilai 127. Karena IP ini di jadikan sebagai loopback dari sebuah sistem protocol TCP/IP

Host ID

Host ID adalah bagian yang mewakili bagian individu dari alamat, bila komputer di segment jaringan anda memiliki alamat, maka jaringan perlu tahu milik siapakah suatu paket data tersebut. Berikut ini adalah tabel kelas-kelas IP dan host yang didukungnya :

Tabel II.4 Internet Protocol Class

Class	Jumlah Host	Jangkauan Octet Pertama
C	256	192 – 223

Host ID mempunyai ketentuan-ketentuan, yaitu :

1. Semua byte Host ID itu tidak boleh semua bytenya : 192.168.2.00000000. atau bernilai 0 (nol).
2. Semua byte Host ID itu tidak boleh semua bytenya : 11111111 atau bernilai 255.

Class C	Network			Host
Octet	W	X	Y	Z

Seperti yang disebutkan diatas bahwa IP adalah 32 bit. Nilai tersebut terbagi menjadi empat bagian nomor delapan bit yang disebut octet. Contoh alamat IP adalah 202.149.240.66. Dengan menggunakan contoh diatas katakanlah bahwa administrator mensetup jaringan dengan semua komputer memiliki bagian nilai yang sama : 202.149.240.xxx ini adalah network ID, sedangkan nomor pada xxx adalah node ID nya.

IP Address kelas C digunakan untuk jaringan yang lebih kecil seperti LAN. Pada *IP Address* kelas C ini 3 (tiga) bit pertamanya selalu berisi 110 (satu satu nol). Bersama 21-bit dan 8-bit terakhir untuk *Host ID*. Jika *IP Address* kelas C adalah 192.168.1.1 maka dapat dikatakan *Network ID* dari IP tersebut adalah 1.

IP Address kelas C dapat menampung lebih kurang 2 juta *Network* dengan masing-masing *Network* memiliki 256 *IP Address*. Berikut adalah karakteristik dari *IP Address* kelas C.

Karakteristik *IP Address* kelas C

Format : 110nnnnn nnnnnnnn hhhhhhhh hhhhhhhh
 Bit Pertama : 10
 Panjang NetID : 24 bit
 Panjang HostID : 8 bit

Byte Pertama : 192 – 233
Jumlah : 2.097.152 kelas C
Range IP : 192.0.0.xxx samapi 233.255.255.xxx
Jumlah IP : 254 *IP Address* pada setiap kelas C

Setiap alamat TCP/IP jatuh pada kelas satu alamat. Kelas mewakili sebuah group alamat yang dapat segera dikenali komponen software sebagai bagian dari sebuah jaringan fisik. Misalnya alamat TCP/IP berikut dan nilai binernya :10.149.240.66 alamat binernya 00001010.10010101.11110000.10000010

Setiap jaringan selalu memerlukan subnet yang biasa disebut subnet mask. Subnet merupakan bagian dari jaringan yang dimaksudkan untuk memecah network ID menjadi beberapabagian kecil. Tujuannya agar beban kerja jaringan tidak terlalu berat. Subnet ini terdiri dari angka 32 bit misalnya :: 11111111.11111111.11111111.00000000 atau 255.255.255.0. Pada subnet mask seluruh bit yang berhubungan dengan host ID diset dengan angka 0.

Penggunaan subnet mask sangat ditentukan dengan banyaknya komputer atau host yang digunakan pada sebuah jaringan. Setiap komputer disebuah jaringan biasanya ingin mengirim data langsung ke komputer lainnya. Komputer pengirim harus memastikan bahwa si penerima berada pada jaringan yang sama atau diluar itu. Subnet mask digunakan oleh protocol stack TCP/IP untukmenentukan bahwa host yang akan dicoba dikomunikasikan berada dijaringan yang sama atau berada di diluar jaringan..

Berikut ini adalah table klasifikasi dari subnet mask :

Tabel II.5 Subnet Mask Class

Class	Subnet Mask
C	255.255.255.0

Subnet mask digunakan untuk membedakan antara network ID dan host ID dan untuk menunjukkan suatu host apakah berada pada jaringan local atau non local. Untuk jaringan non local berarti harus mentransmisi data melalui sebuah router.

Ada suatu penulisan tentang IP Address dengan subnetnya, jika anda sering membaca sebuah buku tentang Jaringan Komputer IP Address biasanya itu dituliskan sebagai berikut :

contoh : 192.168.1.155/24

Ini berarti : sebuah IP address 192.168.1.155 mempunyai subnet 24, lalu subnet 24 itu dari mana ? Jawabanya adalah 24 itu adalah hasil kumulatif atau penambahan dari tiap segment IP Address. Kita telah ketahui bahwa dalam 1 segment IP Address itu terdiri dari 4 byte, dan 1 byte IP Address terdiri dari 8 bit. Jadi jikalau 8 bit itu berjumlah 1 semua maka penkonversian ke dalam bilangan desimalnya berjumlah 255.

Jadi jika sebuah IP Address dengan subnet 24 maka jumlah subnet itu adalah :

$$24 \text{ ----- } 8 \text{ bit } + 8 \text{ bit } + 8 \text{ bit} \\ 11111111.11111111.11111111.0 = 255.255.255.0$$

Penggunaan subnet mask sangat ditentukan oleh banyaknya komputer atau host pada sebuah jaringan. Salah satu standar dalam protokol jaringan yang di kembangkan oleh ISO (*International Standard Organization*) adalah model referensi OSI (*Open System Interconnection*) merupakan suatu sistem yang terbuka untuk berkomunikasi dengan sistem-sistem yang lain.

FILE INTERFACE ETHERNET

File ini terdapat dalam direktori `/etc/etc/sysconfig/network-scripts`. Di dalam direktori ini terdapat beberapa file yang mendukung jalanya sebuah Ethernet card. Berikut ini salah satu yang terdapat dalam direktori tersebut adalah `ifcfg-eth0` yang berisi:

- **Pengaturan IP Address secara tetap**
DEVICE = eth0
BOOTPROTO = none
ONBOOT = yes
NETWORK = 192.168.0.0
NETMASK = 255.255.255.0

IPADDR = 192.168.0.254

USERCTL = no

- **Pengaturan IP Address secara DHCP**

DEVICE = eth0

BOOTPROTO = dhcp

ONBOOT = yes

```
[tehnik@bonazen network-scripts]$ cat ifcfg-eth0
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
METRIC=10
[tehnik@bonazen network-scripts]$ cat ifcfg-eth1
DEVICE=eth1
BOOTPROTO=static
IPADDR=192.168.4.6
NETMASK=255.255.255.0
NETWORK=192.168.4.0
BROADCAST=192.168.4.255
ONBOOT=yes
METRIC=10
MII_NOT_SUPPORTED=no
USERCTL=yes
IPV6INIT=no
IPV6T04INIT=no
PEERDNS=yes
NETMASK=255.255.255.0
IPADDR=192.168.4.6
```

Berikut keterangan dari parameter-parameter diatas :

1. BOOTPROTO = <protocol>, dimana <protocol> adalah salah satu nilai di bawah ini :
 - a. **none** : Tidak ada protocol boot-time yang digunakan
 - b. **bootp** : Menggunakan protocol boot-time BOOTP
 - c. **dhcp** : Menggunakan protocol boot-time DHCP
2. BROADCAST = <address>, di mana <address> merupakan alamat broadcast
3. DEVICE= <name>, dimana <name> adalah nama peraltan fisik (bukan peralatan dinamis).
4. DNS (1,2)=<address>, dimana <address> adalah name server yang di letakkan di dalam file /etc/resolv.conf jika direktif PEERDNS di set menjadi yes.
5. IPADDR=<address>, dimana <address> adalah IP Address yang di gunakan oleh Ethernet
6. NETMASK=<mask>, dimana <mask> adalah nilai subnet mask
7. NETWORK=<address>, dimana <address> adalah IP address jaringan
8. ONBOOT=<answer>, dimana <answer> adalah salah satu nilai di bawah ini :
 - a. **yes** : peralatan ini diaktifkan secara boot
 - b. **no** : peralatan ini tidak diaktifkan pada saat boot
9. PEERDNS=<answer>, dimana <answer> salah satu nilai di bawah ini :
 - a. **Yes** : memodifikasi /etc/resolv.conf jika direktif DNS di set. Jika anda menggunakan dhcp, maka yes merupakan default.
 - b. **No** : tidak memodifikasi /etc/resolv.conf

10. SRCADDR=<address>, dimana <address> adalah source IP Address yang telah ditentukan oleh paket-paket keluar.
11. USERCTL=<answer>, dimana <answer> adalah salah satu nilai berikut :
 - a. yes : user non-root diijinkan mengatur peralatan ini
 - b. no : user non-root tidak diijinkan mengatur peralatan ini.

Scripts pengatur kendali interface

Scripts-scripts ini menonaktifkan dan mengaktifkan interface system. Terdapat dua scripts kendali interface dalam system : **/sbin/ifdown** dan **/sbin/ifup** yang memanggil yang memanggil scripts kendali lain di dalam direktori **/etc/sysconfig/network-scripts**.

Berikut scripts-scripts kendali interface yang digunakan :

1. **ifup-aliases**. Mengkonfigurasi ip alias dari file konfigurasi interface pada saat lebih dari satu IP address diasosiasikan dengan satu interface.
2. **ifdown-cipcb** dan **ifup-cipcb** Dibuat untuk memutuskan atau menyambungkan koneksi *Cripto IP Encapsulation (CIPE)*
3. **ifdown-ipv6** dan **ifup-ipv6**. Mengandung fungsi yang berhubungan dengan ipv6 yang menggunakan variable lingkungan untuk membuat file konfigurasi interface dan **/etc/sysconfig/network-scripts**.
4. **ifup-ipx**. Digunakan untuk menghubungkan suatu interface IPX.
5. **ifup-plip**. Digunakan untuk menghubungkan suatu interface PLIP
6. **ifup-plusb**. Digunakan untuk menghubungkan suatu interface USB.
7. **ifdown-post** dan **ifup-post**. Mengandung perintah-perintah yang dieksekusi setelah suatu interface dikoneksi atau diputuskan.
8. **ifdown-ppp** dan **ifup-ppp**. Digunakan untuk menghubungkan taua memutuskan suatu interface PPP.
9. **ifup-routes**. Menambah route-route statis ke peralatan pada saat interface aktif
10. **ifdown-sit** dan **ifup-sit**. Mengandung fungsi-fungsi yang mengaktifkan atau mematikan tunnel IPv6 dalam suatu koneksi IPv4
11. **ifdown-sl** dan **ifup-sl**. Digunakan untuk mengaktifkan atau mematikan interface SLIP.

Perlu diwaspadai bahwa penghapusan atau perubahan suatu scripts di dalam direktori **/etc/sysconfig/network-scripts** dapat menyebabkan interface bekerja secara aneh atau gagal sama sekali. Anda juga dapat menggunakan scripts **/etc/init.d/network** untuk mengaktifkan dan mematikan interface jaringan yang di konfigurasi. Contohnya adalah sebagai berikut :

- Aktifkan jaringan
/etc/init.d/network start
- Melihat konfigurasi komputer
ifconfig
- Mensetting IP address
ifconfig eth0 ip netmask subnet
- Membuat gateway
route add default gw ip
- Mengetes IP
ping ip
- Stop jaringannya
/etc/init.d/network stop
- Restart Jaringan
/etc/init.d/network restart
- Melihat gateway
route

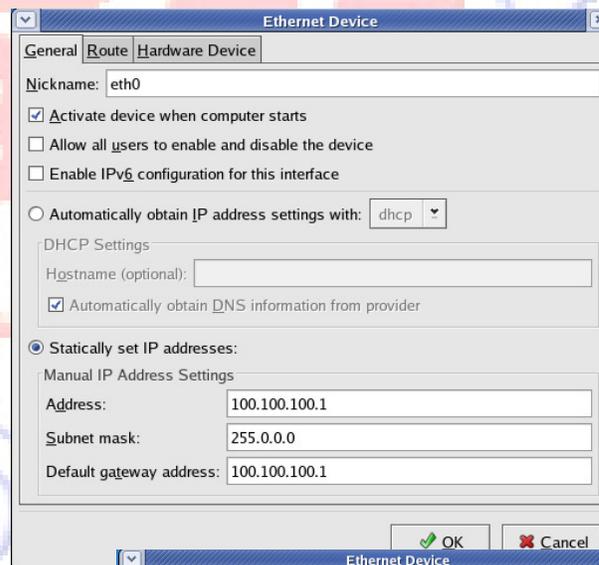
Maksud dari perintah-perintah yang digunakan:

Perintah Dasar	Maksud dari Perintah
ifconfig	Untuk mengetahui IP Address dalam Lan Card atau bias juga dapat di pakai sebagai penginputan IP Address dalam Lan Card
route	Untuk dapat mengetahui IP Adress yang di pakai sebagai

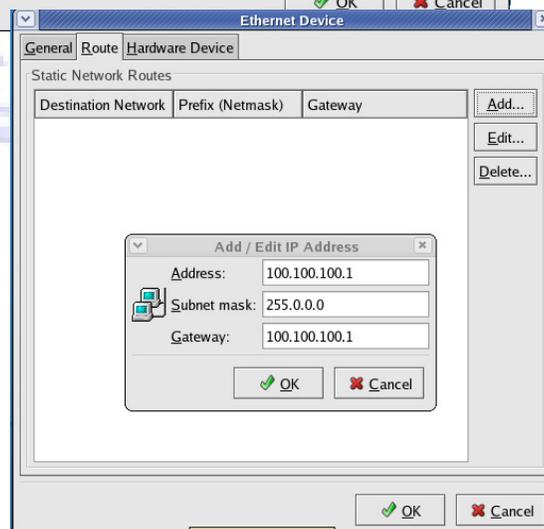
	gateway atau bias juga dipakai sebagai penginputan IP Address yang digunakan sebagai gateway.
Arp	Menampilkan komputer-komputer yang terkoneksi dengan PC anda
Ping	Untuk mengetahui computer anda sudah terkoneksi secara baik atau belum
/etc/init.d	Menjalankan, mematikan atau melihat sebuah service
cat	Memebaca sebuah file
Cd	Masuk ke dalam sebuah direktori
Vi	Membuat atau mengedit sebuah file

Pensettingan IP melalui GUI :

1. Klik application – system setting – network
2. Klik new kemudian isi



3. Klik tab route - klik add



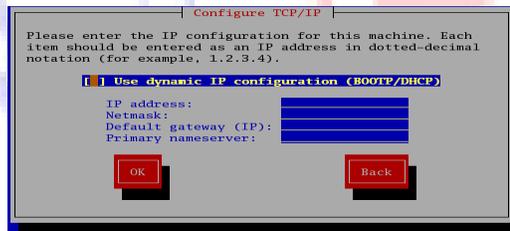
Klik OK 2x kemudian klik activate – klik Yes – OK

Mensetting IP dengan GUI melalui konsol :

1. Ketikkan **# setup** pada konsol

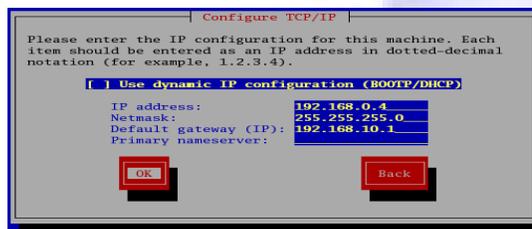


2. Untuk menginput IP Adres pilih **Network Configuration – Run Tool** maka akan tampil konfirmasi untuk penyettingan IP Address, disini kita pilih **Yes**.



3. Input IP address sesuai dengan aturan mainnya dan kolom-kolom yang diinput adalah

Misalnya saya menggunakan IP Address kelas C.



pilih Ok lalu restartlah interface jaringan anda.

SECURE SHELL DAN TELNET

Di dalam operasi sistem linux kita dapat meremot PC atau computer lain melalui jendela konsol atau terminal. Di dalam peremotan ini kita hanya bisa masuk ke dalam system operasinya saja tidak bisa masuk ke dalam tampilan desktopnya. Pada peremotan PC ini kita dapat lakukan melalui instruksi-instruksi yang kita kerjakan tetapi PC yang kita remot tersebut yang akan menjalankannya.

Tetapi syarat utama yang harus kita lakukan untuk melakukan peremotan ini kita harus tahu user account yang ada pada PC tersebut.dan passwordnya. Dan lebih di enak lagi jikalau kita mengetahui password rootnya.

SSH

Sebagai Informasi di dalam peremotan ini kita menggunakan sebuah service yang bernama **sshd** yang di miliki oleh operasi system linux yang bernama **ssh (Secure shell)**, service ini berjalan pada operasi system Linux menggunakan port 22. Dan pengaturan filenya terdapat dalam direktori **/etc/ssh**.

Agar dapat mengakses ssh anda harus menginstal paket **openssh-client** dan **openssh** pada mesin yang digunakan untuk client Untuk memastikan apakah ssh berjalan dengan baik anda dapat mengakses melalui client pada system local dengan menjalankan perintah :

```
# etc/init.d/sshd/status
```

Client SSH akan login ke system local di mana server SSH berjalan menggunakan account aktif, account yang digunakan untuk log in ke mesin local linux dan akan meminta password. Jika anda memasukkan password yang tepat, anda akan mendapatkan prompt shell, meandakan bahwa client dan server berjalan dengan baik. Ketik **exit** dan tekan enter untuk keluar dari ssh.

Jika anda mengakses ssh dari mesin lain, anda harus menentukan hostname atau ip address server tersebut. Jika anda ingin login menggunakan account lain, bukan account yang sedang digunakan, jalankan perintah berikut :

\$ ssh username@server

Dimana server adalah hostname atau ip address server SSH dan username adalah nama account yang ingin digunakan.

Perintah-perintah yang digunakan:

1. Restartlah interface jaringannya

/etc/init.d/network start

2. Setting IP address kelas A / B / C

ifconfig eth0 ip_address_user netmask kelas_dari_IP

Cth : ifconfig eth0 50.50.50.1 netmask 255.0.0.0

3. Setting ip gateway

route add default gw ip_address_user

4. Koneksikan jaringan

ping ip_address_user_tujuan

5. Aktifkan ssh

/etc/init.d/sshd start

6. Kemudian ketik **ssh ip_address_tujuan**

7. Masukkan password root atau password user account yang ada di dalam PC

8. Kirim pesan ke satu user **write nmuser/ip address**

9. Jika tidak bisa, maka ketikkan **mesg y**

10. Kirim pesan ke semua user **wall ketik_pesan**

TELNET

Adalah protokol yang digunakan untuk melakukan remote access, yaitu mengakses suatu "machine" dari jarak jauh. Dan "machine" ini memakai OS berbasis *NIX (linux, BSD maupun Unix). Akses telnet di internet dikenali pada port 23.

Sedangkan sintak penulisan pemanggilannya adalah :

\$ telnet port

Perintah-perintah yang digunakan:

Perintah Dasar	Maksud dari Perintah
ssh	Meremot PC lain
uname -a	Menampilkan Informasi secara keseluruhan tentang nama kernel, hostname, paket instalasi kernel, versi kernelnya, nama hardware mesinnya, nama prosesor yang digunakan, platform dari hardwarenya, operasi system yang di pakai.
uname -n	Menampilkan Informasi hostname
/etc/init.d	Menjalankan atau mematikan sebuah service
Wall	Mengirim pesan ke komputer yang di remot
telnet	Meremot PC

Perintah-perintah yang digunakan:

Setting IP address kelas A / B / C dan buat gatewaynya

2. Koneksikan jaringan
3. Aktifkan service telnetnya yang ada di terminal

Klik application – security level kemudian telnet – OK

4. Aktifkan telnetnya

/etc/init.d/xinetd start

5. Ketik **telnet**

telnet > open

(to) ip_address_tujuan

password : <<input password root>

PEMAKETAN DATA (TAR)

Dalam pemaketatan sebuah file atau direktori di dalam system operasi linux dikenal sebagai tar. Format penulisan command atau perintah tar adalah :

tar <perintah> <option>

Sedangkan paket-paket perintah di dalamnya adalah :

A menambahkan files ke tarball

-- catenate

-- concatenate

c membuat tarball baru
-- create
d menemukan perbedaan atau membandingkan antar isi tarball dan
-- diff = files yang ada di system
-- compare
-- delete = Menghapus file ke akhir dari tarball
r Menambah file ke akhir dari dari tarball
-- append
t menampilkan daftar isi tarball
-- list
u mengganti file di dalam tarball dengan file yang lebih baru
-- update
x mengeluarkan file dari tarball
-- extract
-- get
Sedangkan paket-paket option di dalamnya adalah
- C adalah direktori
- f adalah file
- j adalah bzip2
- p adalah permission

Atau ada cara lain yang menginkannya adalah dengan cara memaketkannya folder tersebut terlebih dahulu. Dalam pemaketan folder tersebut dengan menggunakan perintah **tar**. Lebih jelasnya adalah :

\$ tar -cf rename_file path_file

Penjelasan dari instruksi diatas adalah awalnya anda ketikkan tar selanjutnya untuk mengcreat file atau folder menjadi paket ketikkan -cf lalu dilanjutkan dengan rename_file atau rename_folder maksudnya adalah nama baru dari file atau folder yang anda paketkan dengan instruksi tar. Terakhir anda berikan file atau alamat folder tersebut.

Dalam pemaketan ini baru folder dapat anda kirimkan ke computer lain lewat perintah scp. Sama perintahnya yang dilakukan dalam pengiriman folder yang sudah terpaketkan dengan pengiriman file. Untuk mengekstrak folder yang kita paketkan tersebut diatas anda dapat menggunakan perintah tar. Perintahnya adalah :

\$ tar -xf folder_paketnya

Mengompres dan Mengekstrak File atau direktori yang telah terpaketkan

Sebuah file dapat dikompres atau diekstrak di dalam Linux menggunakan command atau perintah :

- Gzip = Mengompres sebuah file yang telah terpakatkan
- Ungzip = Mengekstrak sebuah file atau folder
- Xcat = Membaca sebuah file yang telah terkompres

Perintah-perintah yang digunakan:

Perintah Dasar	Maksud dari Perintah
Scp	Transfer file
scp -r	Mentransfer folder
Tar	Memaketkan data menjadi satu paket
-xf	Mengeluarkan file dari pemaketan
-cf	Memaketkan file
Gzip	Mengompres file yang telah terpakatkan
gunzip	Mengekstrak file yang telah terkompres dan berekstention .gz
Zcat	Membaca sebuah file yang sedang terekstrak dan file tersebut berekstention .gz

SECURE COPY (SCP)

Salah satu instruksi yang ada di Linux yang berfungsi untuk mentransfer data atau file adalah **scp**. Sedangkan dalam penulisan sintaks ataupun instruksi yang harus dilakukan adalah :

```
$ scp file username@server:path_tujuan
```

Instruksi diatas adalah pengiriman hanya dapat dilakukan dengan satu file, sekarang bagaimana jikalau anda menginginkan transfer beberapa file dalam satu folder. Maka instruksi yang harus anda lakukan adalah menambahkan `-r` lalu dilanjutkan dengan menuliskan nama foldernya selanjutnya menuliskan ip address atau hostname computer tujuan terakhir adalah menuliskan path tujuan folder yang anda kirim ingin di letakkan di mana dalam computer tujuan, maka instruksinya adalah :

```
$ scp -r folder username@server:path_tujuan
```

Kirim File / Direktori di dalam / luar telnet :

- `scp -r / path yang mau dikirim IP yg dituju : / path yang mau diletakkan`
- `scp nm_file yg akan dikirim IP yg dituju : / path yang mau diletakkan`

Catatan :

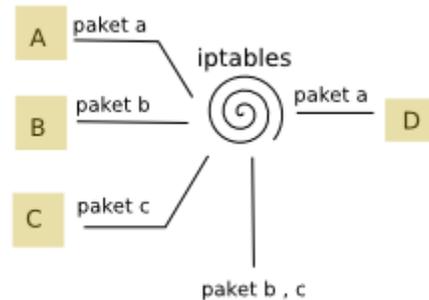
Untuk pengiriman tergantung dari letak posisi user dan juga penggunaan remote.

IPTABLES (IPCHAINS)

Iptables merupakan aplikasi untuk administrasi filtering paket dan Network Address Translation (NAT) pada IPv4. Gambaran umum, iptables digunakan untuk konfigurasi, merawat dan memeriksa rules tables (tabel aturan) tentang filter paket IP yang terdapat di kernel linux. Tiap-tiap tables memiliki beberapa built-in (bawaan) chains kernel linux dan chains buatan user sendiri. Setiap chains memiliki list / daftar aturan untuk mencocokkan

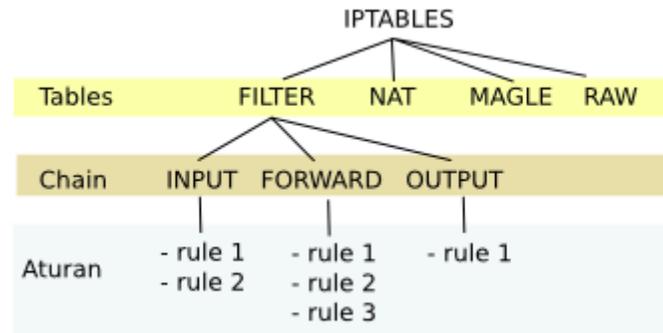
suatu paket yang datang. Setiap aturan tersebut berfungsi memberikan keputusan eksekusi apa yang akan dilakukan bila paket yang datang cocok dengan aturan yang telah dibuat.

Berikut gambaran dan contoh kegunaan iptables :



Terlihat bahwa komputer A, B dan C mengirim paket data kepada komputer D yang sudah dilengkapi iptables. Pada konfigurasi iptables dibuat pernyataan bahwa hanya paket yang memiliki huruf C yang boleh diterima komputer D, selain itu ditolak semua. Dan hasilnya, hanya komputer C saja yang bisa mengirimkan paket dan diterima komputer D. Tentunya dengan gambaran sederhana ini kita bisa membayangkan kegunaan iptables lebih detail. Misalkan bila ada komputer asing yang mengirimkan paket data untuk keperluan scanning atau hacking maka paket tersebut langsung ditolak oleh komputer kita.

Iptables memiliki 4 tabel aturan yaitu filter, nat, mangle dan raw. Yang akan saya bahas adalah aturan "filter" karena ini yang seringkali digunakan dalam pengaturan jaringan. Kita sudah tahu bahwa pada table terdapat chains (rantai) yang berisi rules / aturan. Namun perlu diketahui tiap table memiliki chains yang berbeda-beda. Chains pada tables "filter" dari 3 fungsi yaitu INPUT, FORWARD dan OUTPUT. INPUT untuk paket yang disiapkan untuk socket lokal atau komputer kita sendiri. FORWARD untuk paket yang diarahkan / routing ke box dan OUTPUT untuk paket yang di generate / dibuat sendiri. Cara gampang nya untuk memahami, chains INPUT berguna untuk mengatasi paket data yang masuk. FORWARD berguna untuk mengalihkan paket yang datang dan OUTPUT berguna untuk menghasilkan paket data yang akan diteruskan nantinya.



Setelah kita mengetahui ketiga fungsi chains tersebut maka kita perlu mengetahui command / perintah dalam konfigurasi iptables. Perlu diketahui bahwa dalam satu chain terdiri dari beberapa rule / aturan. Tiap-tiap aturan tersebut memiliki urutan prioritas tersendiri. Bila ada paket datang maka akan disesuaikan dengan chains. Setelah dikelompokkan maka paket tersebut diseleksi oleh rule yang terdapat pada chain. Bila ada paket datang yang tidak cocok dengan aturan pertama maka akan diteruskan menuju seleksi aturan kedua dan selanjutnya hingga aturan terakhir. Berikut command-command yang umum digunakan :

-A

yaitu append. Memiliki struktur `-A [chain] [aturan]`. Berfungsi untuk menetapkan aturan ke dalam chains. Contoh : `iptables -A INPUT -s 192.168.0.1`

-D

yaitu delete aturan. Memiliki struktur `-D [chain] [aturan]` atau `-D [chain] [nomor urutan aturan]`. Berfungsi untuk menghapus aturan dari chains atau menghapus aturan berdasarkan urutan list didalam chains. Contoh : `iptables -D INPUT 1` (menghapus aturan pertama dalam chain INPUT)

-I

yaitu insert. Memiliki struktur `-I [chain] [nomor urutan aturan] [aturan]`. Berfungsi untuk memasukan aturan baru kedalam chain. Bila nomor urutan aturan adalah 1 berarti aturan tersebut dimasukkan ke prioritas utama dalam chain. Contoh : `iptables -I OUTPUT 2 -s 192.168.0.1`

-R

yaitu replace. Memiliki struktur `-R [chain] [nomor urutan aturan] [aturan baru]`. Berfungsi untuk menimpa / me-replace aturan lama dengan aturan baru dalam chain. Contoh :

iptables -I OUTPUT 2 -s 192.168.0.1 (menimpa rule kedua dengan rule baru -s 192.168.0.1

-L

yaitu list. Memiliki struktur -L [chain]. Berfungsi untuk menampilkan daftar aturan-aturan didalam chain. Bila chain tidak disertakan maka akan muncul aturan dalam semua chain. Contoh : iptables -L INPUT

-F

yaitu flush. Memiliki struktur -F [chain]. Berfungsi untuk menghilangkan semua aturan pada chain. Contoh: iptables -F FORWARD (menghapus semua aturan didalam chain FORWARD)

-N

yaitu new. Memiliki struktur -N [chain]. Berfungsi untuk membuat chain baru. Contoh: iptables -N GET

-X

yaitu delete chain. Memiliki struktur -X [chain]. Berfungsi untuk menghapus chain dan ini berbeda dengan -D yang berguna untuk menghapus rule saja. Untuk menghapus chain, dipastikan terlebih dahulu bahwa tidak ada aturan-aturan didalam chain tersebut. Dapat digunakan flush untuk menghapus aturan-aturan di dalam chains, Contoh: iptables -X GET

-E

yaitu rename chain. Memiliki struktur -E [chain lama] [chain baru]. Berfungsi untuk me-rename / mengganti nama chain yang ada didalam iptables. Contoh: iptables -E GET PUT

Setelah kita mengetahui perintah-perintah pada iptables, yang kira perlu pahami berikutnya adalah parameter. Kita melihat contoh perintah "iptables -I OUTPUT 2 -s 192.168.0.1". Disini, -s merupakan parameter. Kegunaan parameter adalah untuk mengidentifikasi spesifikasi aturan dan digunakan untuk mengikuti perintah umum seperti add, delete, insert, replace dan append.

-p

yaitu menunjukkan protokol. Untuk mengidentifikasi protokol dalam rule seperti tcp, udp, icmp,dst diperlukan parameter ini. Contoh: iptables -A INPUT -p tcp

-m

yaitu match option. Mirip dengan -p tetapi perbedaannya adalah modul yang digunakan. Bila pada -p menggunakan modul yang bersifat spesifik tetapi berbeda dengan -m. Dengan menggunakan parameter ini, kita bebas menentukan nama module yang dipakai dan meng-variasikannya dalam perintah selanjutnya. Contoh : iptables -A INPUT -s 192.168.0.0/16 -m comment --comment "IP yang di-blok" (berarti modul comment berisi perintah --comment "IP yang di-blok")

-s

yaitu source alamat hostname / ip. Contoh : iptables -A INPUT -s 192.168.0.1

-d

yaitu destination / tujuan dari alamat ip. Contoh: iptables -A INPUT -d 192.168.0.2

-j

yaitu jump. Berfungsi untuk memberikan keputusan setelah paket data cocok dengan aturan. Biasanya terdapat di akhir perintah dan diikuti argumen perintah. Contoh : iptables -A INPUT -s 192.168.0.2 -j DROP.

-i

yaitu in-interface alias nama interface yang menerima kiriman paket (terbatas pada chain INPUT, FORWARD dan PREROUTING saja). Contoh: iptables -A INPUT -i eth0 -s 192.168.0.2

-o

yaitu out-interface alias nama interface yang akan mengirim paket keluar (terbatas pada chain FORWARD, OUTPUT dan POSTROUTING). Contoh : iptables -A INPUT -o eth1 -s 192.168.0.2

-c

yaitu counter untuk menghitung paket-paket yang lewat dari sebuah aturan. Penulisan parameter ditulis sebelum command semacam APPEND,INSERT,REPLACE,dst. Contoh : iptables -c -A INPUT -s 192.168.0.2

-n

yaitu numeric. Parameter ini akan menampilkan output numeric seperti hostname,ip, port, nama network,dst. Contoh: iptables -L -n

-v

yaitu verbose yang berarti menampilkan informasi secara keseluruhan alias dalam bahasa indonesia terjemahannya "bertele-tele". Contoh: `iptables -L -n -v`

Paket-paket yang masuk akan di periksa, apakah rusak, salah informasi atau tidak, kemudian di berikan ke chain INPUT. Tergantung pada informasi yang terdapat di dalam *header* paket dan kebijakan dalam *ruleset*, keputusan yang diambil untuk suatu paket dapat berupa :

1. ACCEPT
Menerima paket dan diproses lebih lanjut oleh kernel
2. DROP
Menolak paket tanpa pemberitahuan terlebih dahulu
3. REJECT
Mengembalikan paket ke asalnya dngan pesan kesalahan ICMP
4. LOG
Melakukan log (pencatatan) terhadap paket yang bersesuaian
5. RETURN
Untuk *chain user-defined* akan dikembalikan ke *chain* yang memanggil, sedangkan untuk *chain* INPUT, OUTPUT dan FORWARD akan dijalankan kebijakan default.
6. Mengirim ke chain user-defined

Rule dalam IPTABLES dapat dikenakan terhadap asal paket (-s), tujuan paket (-d), protocol (-p) dan port. Misalnya untuk menolak semua paket yang dating dari mesin dengan IP Address 192.168.0.100 dapat di tulis :

```
IPTABLES -t filter -A INPUT -s 192.168.0.100 -j DROP
```

atau

```
IPTABLES -A INPUT -s 192.168.0.100 -j DROP
```

Baris kedua tidak menyertakan `-t` filter karena sebenarnya table default yang digunakan IPTABLES adalah filter

Perintah-perintah diatas di tulis secara langsung di shell Linux. Sebaiknya anda login sebagai root atau user lain yang diberi wewenang menjalankan IPTABLES

Tanda seru (!) dapat ditempatkan di depan IP address untuk membuat pengecualian terhadap IP tersebut. Perintah di bawah ini akan menolak semua paket yang masuk dari semua host kecuali dari mesin dengan IP address 192.168.0.100

```
IPTABLES -A INPUT -s ! 192.168.0.100 -j DROP
```

Perintah-perintah yang digunakan:

Perintah Dasar	Maksud dari Perintah
iptables	Pembuatan firewall di dalam PC
Man	Melihat description suatu command atau perintah
setup	Memanggil konfigurasi computer melalui GUI

Perintah-perintah yang digunakan:

1. Setting IP address kelas A
ifconfig eth0 50.50.50.1 netmask 255.0.0.0
2. Koneksikan jaringan
ping ip_address_tujuan
3. Mengembalikan semua aturan pada posisi default
iptables -F
4. Melihat semua aturan yang ada
iptables -L
5. Memblok pengiriman data keluar
iptables -A OUTPUT -d ip_user_tujuan -j DROP
6. Koneksikan kembali ke user lain
7. Normalkan kembali semua aturan dan lihat semua aturan yang ada.

8. Buat direktori dengan nama masing-masing di user BSI kemudian buat file dengan nama coba1 isi file dengan : BELAJAR FIREWALL
9. Kirim direktori nama masing-masing yang ada di user BSI ke user lain.

```
scp -r /home/bsi/direktori ip_user_tujuan :/home/bsi
```

10. Buat file dengan nama coba2 di direktori nama masing-masing yang ada di user BSI

11. Menolak kiriman data dari keluar

```
iptables -A INPUT -s ip_user_tujuan -j REJECT
```

12. Kirim kembali file coba2 ke direktori nama masing-masing user BSI ip_user_tujuan

```
scp /home/bsi/direktori/nama_file ip_user_tujuan  
:/home/bsi/direktori
```

REMOTE DESKTOP DAN GENOMEETING

Di dalam remot dekstop ini Linux menggunakan sebuah aplikasi VNC yaitu Virtual Network Computing. Selain kita bisa meremot Linux dengan Linux, ternyata Windows

pun bisa meremot Linux, begitu pula Linux pun dapat meremot Windows. Tetapi dalam system seperti kita perlu software lagi yang harus diinstal di Windows agar kedua operasi system ini dapat berjalan salang meremot.

Software yang digunakan adalah VNC (Virtual Network Computing)

Langkah-langkah Remote Desktop :

1. Klik application – klik preferences – remote desktop

2. pilih



3. Jika security ingin di aktifkan maka di cek list setelah itu masukan passwordnya.

4. Klik Close

Langkah – langkah VNC Viewer :

1. Klik application – klik Accessories – VNC Viewer

2. Masukan IP user lain – Klik OK

Langkah – Langkah Gnomemeeting :

1. Klik application – klik internet- video conferencing

2. Masukkan IP user lain



3. Kirim pesan di

SHARING DATA (SAMBA)

Samba merupakan implementasi dari protokol SMB (Server Message Block) pada sistem UNIX. Protokol ini digu-nakan 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

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
```

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

Perintah-perintah yang digunakan:

Langkah-langkah :

Mengaktifkan samba

/etc/init.d/smb start

Menshare data yang akan di share

vi /etc/samba/smb.conf

tekan insert untuk mengedit, ketik nama file / direktori yang akan di share

```
root@bonazen:~
File Edit View Terminal Tabs Help
; writable = yes
; printable = no

# The following two entries demonstrate how to share a directory so that two
# users can place files there that will be owned by the specific users. In this
# setup, the directory should be writable by both users and should have the
# sticky bit set on it to prevent abuse. Obviously this could be extended to
# as many users as required.
;[myshare]
; comment = Mary's and Fred's stuff
; path = /usr/somewhere/shared
; valid users = mary fred
; public = no
; writable = yes
; printable = no
; create mask = 0765

[bonazen]
  path = /home/bonazen/bonazen
  writable = yes
  public = yes
```

Kemudian

1. Kli

2. Klik smb kemudian klik save

Restart samba

/etc/init.d/smb restart

3. Klik application – network server

4. Klik kanan – windows network – browse folder

5. Masukkan IP user lain

