

# Administrasi Start Up Linux

---



FAKULTAS TEKNOLOGI INFORMASI  
PROGRAM STUDI TEKNIK INFORMATIKA

## Linux: dari POST hingga Login

Dengan mengetahui cara kerja komputer permasalahan yang timbul ketika menggunakan komputer dapat lebih cepat dikenali dan diperbaiki. Salah satu bagian yang perlu diketahui dari cara kerja komputer adalah bagaimana komputer memulai sistem operasi dari mulai komputer dinyalakan hingga pengguna mendapatkan tampilan login dari sistem operasi.

Berikut ini adalah langkah-langkah yang dilakukan mulai dari komputer dinyalakan hingga login sistem operasi:

No	Operasi	Sistem
1	Melakukan Power On-Self Test	BIOS
2	Memeriksa konfigurasi hardware komputer	
3	Mencari perangkat yang memiliki kandidat sistem operasi	
4	Membaca media penyimpanan perangkat dan mencari bootloader	
5	Mengeksekusi bootloader dan memindahkan kendali ke bootloader	
6	Memilih sistem operasi yang akan dimulai	Bootloader
7	Mengeksekusi kernel sistem operasi dan memindahkan kendali kepada kernel	
8	Kernel memulai proses-proses <i>start up</i>	Kernel/OS

Pada sistem operasi turunan UNIX terdapat dua model proses start up, System V dan BSDStyle. Pada model System V (SysV) terdapat 7 level yang dapat dipilih dan dikonfigurasi sesuai kebutuhan sedangkan model BSD Style hanya memiliki satu level. Distribusi Linux yang ada menggunakan model System V.

Berikut ini adalah keterangan mengenai level yang ada pada System V:

Level	Keterangan
0	Halt. Level ini dipanggil ketika komputer akan dimatikan.

1	Single. Level ini biasanya digunakan untuk keperluan <i>recovery</i> . Hanya satu user yang disediakan pada level ini yaitu root.
2	Tidak didefinisikan, user dapat menentukan dengan bebas apa yang akan dilakukan pada level ini.
3	Multi user tanpa GUI dan disertai dengan fungsi-fungsi jaringan.
4	Tidak didefinisikan, user dapat menentukan dengan bebas apa yang akan dilakukan pada level ini.
5	Multi user dengan GUI dan disertai dengan fungsi-fungsi jaringan.
6	Reboot. Level ini dipanggil ketika komputer akan di- <i>restart</i> .

## Init

Proses pertama yang dimulai oleh kernel adalah init. Jika kernel tidak dapat menemukan init maka kernel akan mencoba mengeksekusi shell. Jika shell juga tidak ditemukan maka proses startup gagal.

Proses init akan memilih level yang akan dijalankan (*run level*) berdasarkan pada sebuah file konfigurasi bernama `/etc/inittab`. File `/etc/inittab` tidak hanya berisi konfigurasi level yang akan digunakan tetapi juga menentukan beberapa hal lainnya seperti konfigurasi banyaknya terminal, jenis terminal, bahkan fitur otomatis untuk mematikan komputer ketika UPS hampir kehabisan tenaga.

Di bawah ini adalah contoh dari isi file `/etc/inittab`:

```
# /etc/inittab
# This is the main configuration file of
# /sbin/init, which
# is executed by the kernel on startup.

# The default runlevel
id:5:initdefault:
```

```
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6
```

```
# what to do in single-user mode
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin
```

```
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

```
# getty-programs for the normal runlevels
# :::
# The "id" field MUST be the same as the
last
# characters of the device (after "tty").
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Run level yang dipilih saat startup ditentukan pada baris konfigurasi berikut:

```
id:5:initdefault:
```

Pada baris ini run level yang dipilih adalah 5. Untuk mengganti run level pilihan cukup dengan mengganti angka 5 dengan run level dari 0 s.d. 6.

Fitur lainnya adalah mengkonfigurasi terminal-terminal yang akan ditampilkan untuk pengguna. Format barisnya adalah sebagai berikut:

```
id:runlevels:action:process
```

Id	Nomor terminal
runlevel s	Run level di mana terminal tersebut akan diaktifkan
Action	Menentukan apakah perintah pada kolom proses akan dijalankan kembali (respawn) ketika perintah telah selesai atau cukup sekali saja ( <i>once</i> )
Process	Perintah yang akan dijalankan untuk memulai terminal

Baris:

2:2345:respawn:/sbin/mingetty tty2

berarti terminal nomor 2 akan diaktifkan pada run level 2, 3, 4, dan 5 dengan proses untuk terminal tersebut adalah /sbin/mingetty dan setiap kali proses telah selesai (*logout*) maka perintah /sbin/mingetty akan dijalankan ulang.

## Manajemen Layanan-layanan pada System V

Pengguna sistem operasi Linux dapat menentukan sendiri layanan-layanan yang akan dimulai pada run level tertentu pada saat start up. Contohnya pada run level 5 pengguna akan memulai layanan X server yang berfungsi untuk memberikan tampilan antarmuka grafis untuk *desktop manager* seperti KDE dan GNOME. Pada run level 3 layanan X server tidak dimulai sehingga pengguna tidak mendapatkan tampilan antarmuka grafis. Pada run level 0 layanan tidak dimulai melainkan dimatikan karena sistem akan dimatikan.

*Script* yang digunakan untuk memulai dan mematikan layanan dikumpulkan dalam sebuah direktori beralamat di /etc/rc.d atau /etc/init.d. Layanan-layanan yang akan dimulai atau dimatikan pada run level tertentu ditentukan pada direktori-direktori yang merepresentasikan masing-masing run level. Direktori-direktori tersebut adalah

- /etc/init.d/rc0.d merepresentasikan run level 0
- /etc/init.d/rc1.d merepresentasikan run level 1

- /etc/init.d/rc2.d merepresentasikan run level 2
- /etc/init.d/rc3.d merepresentasikan run level 3
- /etc/init.d/rc4.d merepresentasikan run level 4
- /etc/init.d/rc5.d merepresentasikan run level 5
- /etc/init.d/rc6.d merepresentasikan run level 6

### Memulai Layanan pada Run Level Tertentu

Untuk menambahkan layanan tertentu yang akan dijalankan pada run level tertentu yang perlu dilakukan adalah membuat sebuah link dengan nama berpola khusus di direktori yang merepresentasikan run level tersebut. Polanya adalah sebagai berikut:

<SK><urutan dijalankan><nama\_service>

Contohnya: jika pengguna ingin menjalankan layanan httpd atau lebih dikenal dengan server web pada run level 3 maka pada direktori /etc/init.d/rc3.d buatlah sebuah *softlink* dengan nama S20httpd dengan perintah:

```
#> ln -s /etc/init.d/rc.d/httpd /etc/init.d/rc3.d/S20httpd
```

yang berarti layanan web server akan dimulai pada urutan ke-20 pada run level 3.

### Mematikan Layanan pada Run Level Tertentu

Sebaliknya untuk mematikan layanan ketika run level tertentu dapat dilakukan dengan mengganti huruf S (*start*) dengan K (*kill*) pada direktori run level yang diinginkan.

Contohnya: jika pengguna ingin menghentikan layanan bind atau lebih dikenal dengan server DNS pada run level 6 maka pada direktori /etc/init.d/rc6.d buatlah sebuah *softlink* dengan nama K20bind dengan perintah:

```
#> ln -s /etc/init.d/rc.d/bind /etc/init.d/rc6.d/K20bind
```

yang berarti layanan DNS server akan dihentikan pada urutan ke-20 pada run level 6.



## I.

### PENGENALAN LINUX DAN LINGKUNGAN KERJA

#### I.1 Pengenalan Linux

##### I.1.1 Awal Mula Linux



Linux pada awalnya dibuat oleh seorang mahasiswa Finlandia yang bernama Linus Torvalds. Dulunya Linux merupakan proyek hobi yang diinspirasi dari Minix, yaitu sistem UNIX kecil yang dikembangkan oleh Andrew Tanenbaum. Linux versi 0.01 dikerjakan sekitar bulan Agustus 1991. Kemudian pada tanggal 5 Oktober 1991, Linus mengumumkan versi resmi Linux, yaitu versi 0.02 yang hanya dapat menjalankan shell bash (GNU Bourne Again Shell) dan gcc (GNU C Compiler).

Torvald menggunakan logo penguin sebagai maskot linux. Dari sayembara yang publikasikan di internet, logo racikan Larry Ewing lah yang terpilih sebagai mascot GNU/Linux. Dan logo ini banyak disebut sebagai Tux. Sehingga tidak heran banyak icon ataupun artcover serta perangkat lunak dari linux bercirikan penguin, seperti TUX racer, Xpenguin, pingus, dan lain sebagainya.

##### I.1.2 Manfaat Linux Sebagai Sistem Operasi

Terlepas dari isu kultus antara windows dan linux, ada beberapa hal yang menjadi manfaat untuk menggunakan GNU/Linux:

1. GNU/Linux menggunakan lisensi free software. Sehingga menyebabkan kita bisa menggunakan GNU/Linux untuk keperluan apapun tanpa harus mengeluarkan *cost*. Pertimbangan ini sangat cocok untuk industri kecil yang ingin berkembang. GNU/Linux Ubuntu pada situs resminya menyediakan formulir pemesanan secara gratis CD/DVD installernya. Banyak situs-situs repository yang bisa kita gunakan untuk mengunduh GNU/Linux secara bebas dan legal.
2. GNU/Linux bersifat open source, bagi yang memiliki kemampuan untuk membangun aplikasi bisa melakukan modifikasi terhadap segala sesuatu dalam GNU/Linux (kode program), mengkomersilkan, dengan syarat harus menggunakan lisensi GPL; menyertakan source code, dan mendistribusikan perangkat lunak hasil modifikasi dengan juga menggunakan lisensi GPL



3. Dukungan komunitas internet yang sangat matang. Banyak sekali blog-blog, web forum, aggregator, webzine, social networking site tempat para komunitas GNU/Linux berkumpul dan saling berbagi.
4. Dukungan vendor-vendor besar terhadap GNU/Linux, seperti Sun Microsystems, IBM, Google. Salah satu bentuknya bisa dilihat kontribusi mereka terhadap perkembangan perangkat lunak yang bisa digunakan pada GNU/Linux, seperti Open Office (Sun Microsystems), Picasa (google).
5. Linux memiliki kestabilan yang sangat tinggi. Banyak konfigurasi linux yang tidak membutuhkan proses restart, ketika terdapat perubahan konfigurasi. Tingkat kestabilan Linux sangat tinggi, sangat tahan terhadap sistem crash, jauh dari sifat *lagging*. GNU/Linux juga sangat *immune* terhadap virus-virus yang ada di Windows. Ini semua dikarenakan GNU/Linux merupakan sistem operasi yang mengadopsi konsep UNIX, yang biasanya digunakan sebagai server pada industri *enterprise*. Oleh karena itu GNU/Linux sering disebut sebagai *Unix Like Operating System*.
6. Distro GNU/Linux yang bermacam-macam, membuat kita bisa memilih berdasarkan kebutuhan dan selera. Seperti beberapa distro populer saat ini SuSE, Fedora Linux, Red Hat, Mandriva, memiliki berbagai fitur yang bisa mendukung kebutuhan kita.

### 1.1.3 Distribusi Linux (Linux & GNU)

Saat ini, istilah “Linux” sering diartikan sebagai lingkungan *software* yang dijalankan melalui kernel linux dan dilengkapi dengan berbagai komponen aplikasi di dalamnya. Sehingga banyak orang menyebut sebagai Distro (distribusi) Linux. Pada dasarnya distro linux merupakan gabungan antara kernel linux dan aplikasi-aplikasi GNU yang dipaketkan sedemikian rupa. Maka tidak heran kita sering mendengar distro linux populer seperti; Red Hat linux, Ubuntu, Fedora linux, OpenSuSE, Slackware linux, Debian, Mandriva, Gentoo dan lain-lain.

Tidak usah dibingungkan dengan jenis-jenis distro ini, karena pada umumnya perbedaanya hanya berkisar pada:

#### 1. Metode Instalasi

Tiap distribusi linux memiliki pendekatan yang berbeda dalam proses instalasinya. Kebanyakan distro GNU/Linux menggunakan metode instalasi standar dengan booting dari CD, namun Ubuntu selalu menggunakan live CD.

#### 2. Cara konfigurasi

SuSE linux dilengkapi dengan YAST untuk melakukan segala konfigurasi, Mandriva menggunakan Control Center atau drake (HardDrake,

UserDrake, dll), Red hat menggunakan setup.

3. Manajemen paket

Red Hat menggunakan rpm, sedangkan debian menggunakan dpkg, slackware menggunakan pkgtool, mandriva menggunakan urpmi. Namun saat ini rpm lebih populer dibandingkan dengan manajemen paket distro-distro lainnya.

4. GNU Software yang dipaketkan pada CD Installer

Sifat lisensi bebas dari linux maupun GNU software, menyebabkan, tiap-tiap distribusi linux memaketkan software secara berbeda, sesuai dengan visi dan misi serta cita rasa dari tiap-tiap distribusi linux. GNU/Linux mandriva yang mengedepankan multimedia, banyak memaketkan perangkat lunak yang sifatnya menghibur. Debian yang lebih diperuntukan untuk server hanya memaketkan kebutuhan server.

5. Performance

Kondisi yang berbeda-beda dari tiap distribusi menyebabkan performance yang ditampilkan juga berbeda-beda.e

6. Community Support

Tiap-tiap distro memiliki komunitas sendiri-sendiri. Tingkat kematangan dari merekapun sudah banyak teruji, karena sifat terbuka dan mau berbagi.

#### 1.1.4 Perbandingan Linux dengan Windows

Windows memang jauh lebih dulu muncul dibandingkan GNU/Linux, namun hal ini tidak menyebabkan GNU/Linux bisa dipandang sebelah mata, terutama dalam hal aplikasi penunjang kehidupan sehari-hari. Momok seperti ini menjadikan orang-orang banyak beropini bahwa GNU/Linux merupakan sistem operasi yang hanya ditujukan kepada manusia2x nerdy.

Dengan banyak kontribusi dari para komunitas open source, semakin lama GNU/Linux tidak jauh berbeda dengan sistem operasi windows. Terutama bagi yang sudah tergantung terhadap aplikasi semacam Microsoft office, winamp, adobe photoshop, dll, bahwa anda tidak perlu khawatir, karena saat ini GNU/Linux bisa dikatakan memiliki padanan aplikasi sedemikian yang sudah cukup matang, dan terutama gratis. Berikut ini merupakan bukti bahwa GNU/Linux saat ini bisa diandalkan, selayaknya bagi kalian yang sangat mengandalkan sistem operasi windows.

Aplikasi	Windows	Linux	Keterangan
Office suite	Microsoft Office	Open Office	Aplikasi menjalankan pekerjaan rutin kantor

Web browser	Internet Explorer	Firefox, Chrome	Aplikasi untuk menjelajahi internet
Instant messaging	Windows messenger	Pidgin, kopete	Aplikasi ngobrol di dunia maya
Mp3 player	Winamp, Windows media player	Xmms, amaroK	Aplikasi untuk mendengarkan audio file
Diagram	Microsoft Visio	Dia, Kivio	Aplikasi untuk membuat diagram
Image editor	Adobe photoshop, 3DMax	GIMP, blender, google picassa	Aplikasi editor grafis
Programming IDE	Microsoft .NET	Mono, Netbeans, Eclipse, Kdevelop,	Aplikasi developer
Archiving	Winzip, winrar	Archiever, gnozip, gzip	Aplikasi kompresor
Video player	Windows media player	kaffeine	Aplikasi pemutar video
Email client	Microsoft outlook	Evolution, thunderbird	Aplikasi mail elektronik

## **I.2 Panduan Praktis Instalasi Linux**

### **I.2.1 Metode Instalasi**

Salah satu kekuatan GNU/Linux lainnya adalah program instalasi bisa dijalankan dengan berbagai cara. Sebagai contoh, jika kita ingin melakukan instalasi pada satu atau dua sistem berbeda sangat mudah dengan menggunakan CD-ROM karena membutuhkan waktu minimal. Tapi coba bandingkan dengan ketika kita harus melakukan instalasi linux pada sepuluh atau seratus sistem yang terkoneksi pada satu segmen jaringan yang sama, jika kita menggunakan cara yang sama, maka akan sangat menyita waktu, oleh karena itu GNU/Linux menyediakan mekanisme untuk menginstall secara bersamaan pada semua system dengan menggunakan mekanisme booting dari PXE.

Satu hal lagi yang perlu dimengerti di sini, bahwa dari sekian cara menginstall linux, kita bisa mengkombinasikan satu sama lain sesuai dengan kebutuhan kita. Secara umum, berikut ini merupakan beberapa metode yang kerap digunakan oleh berbagai distro GNU/Linux pada program instalasinya, yaitu:

- CD-ROM
- Hard drive

- Instalasi dengan network
- Kickstart (pada redhat/ fedora)
- PXE

### **1.2.1.1 Instalasi dengan Media CD-ROM**

Instalasi dengan metode ini sangat umum. Bahkan beberapa distro GNU/Linux menggunakan live CD sebagai media instalasi sekaligus uji coba kompatibilitas, seperti GNU/Linux Ubuntu, knoppix. Untuk menggunakan metode ini, aturlah terlebih dulu urutan boot pada BIOS, dan tentukan CD-ROM/DVD-ROM sebagai media booting pertama kali. Kita bisa lihat pada bab selanjutnya akan di paparkan salah satu mekanisme instalasi GNU/Linux Fedora.

### **1.2.1.2 Instalasi Melalui Jaringan atau Hard Drive**

Dua mekanisme instalasi ini sangat mirip, yaitu menggunakan *image boot.iso* yang tersedia pada CD instalasi. Bedanya untuk metode melalui jaringan, setelah booting CD yang berisi *boot.iso*, sumber file instalasi diarahkan ke jaringan, sedangkan untuk metode melalui hard drive, sumber instalasi diarahkan ke drive partisi yang berisi file-file instalasi GNU/Linux.

### **1.2.1.3 Kickstart**

Kickstart merupakan salah satu mekanisma script konfigurasi yang digunakan untuk proses instalasi GNU/Linux. Metode ini biasanya disediakan khusus pada GNU/Linux Red Hat.

### **1.2.1.4 PXE**

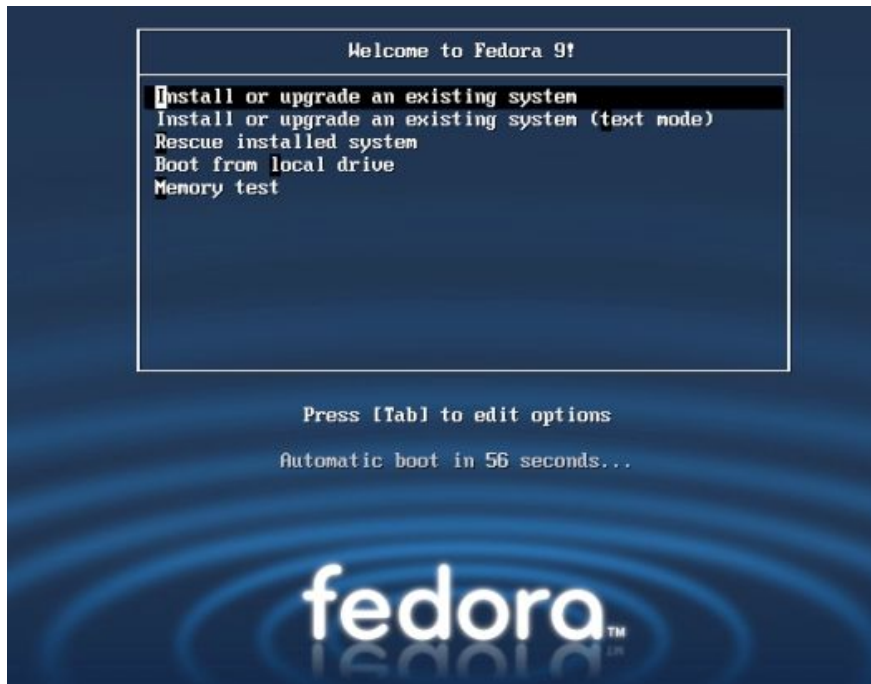
PXE atau Pre-Execution Environment, merupakan metode instalasi yang tersedia pada beberapa jenis kartu jaringan tertentu dan dapat digunakan untuk instalasi melalui jaringan dengan terkoneksi pada sebuah Network File Server. Pada File Server terdapat file-file yang dibutuhkan untuk booting instalasi.

## **1.2.2 Instalasi Fedora 9**

Seperti yang disebutkan sebelumnya, bahwa sifat open source Linux, membuat banyak variasi dari distribusi GNU/Linux. Dengan dasar inilah terkadang tiap-tiap distribusi linux membuat proses dari instalasi dengan cita rasa masing-masing, walaupun esensi dari tiap-tiap proses instalasi tidak jauh berbeda. Untuk itu berikut ini merupakan cuplikan cara instalasi GNU/Linux Fedora, yang merupakan salah satu distribusi populer saat ini.

### **1.2.2.1 Langkah-langkah Instalasi Linux**

1. Ubahlah konfigurasi *first boot* dari BIOS komputer menjadi CD-ROM, sehingga ketika kita menyalakan komputer, CD installer fedora akan terbaca, dan pilihan prosedur instalasi seperti di bawah ini akan tampak.



2. Setelah kita memilih opsi *Install or upgrade an existing system* program instalasi Fedora linux akan melakukan verifikasi CD, apakah ada paket yang rusak atau tidak. Hal ini ditujukan agar tidak ada interupsi apapun ketika proses instalasi berjalan. Jika kita sudah yakin terhadap hal ini, untuk mempercepat proses, pilihlah opsi *Skip*.

Welcome to Fedora for i386

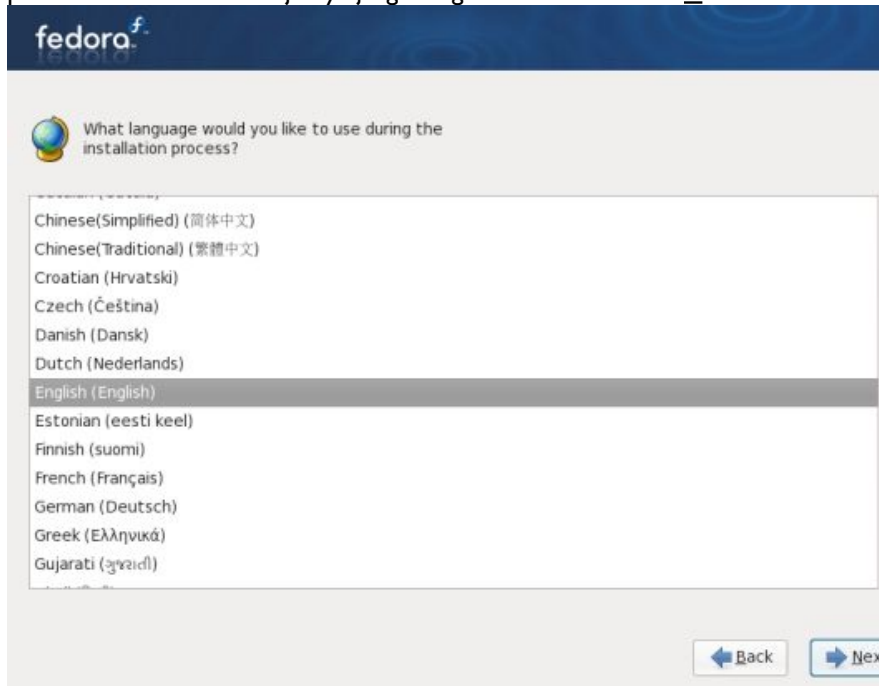


<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

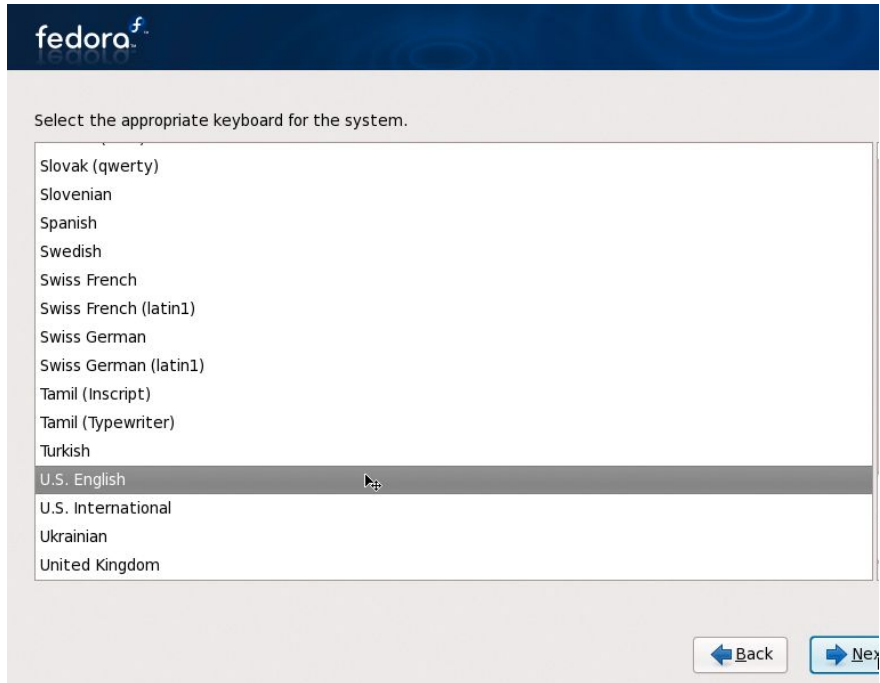
3. *Splash screen* Fedora Linux akan menjadi pembuka dari proses instalasi. Untuk menuju step selanjutnya klik tombol Next.



4. Kita bisa memilih panduan bahasa untuk setiap langkah proses instalasi. Dalam hal ini anda bisa saja memilih bahasa inggris untuk menunjang proses instalasi. Dan selanjutnya jangan ragu untuk klik tombol Next.

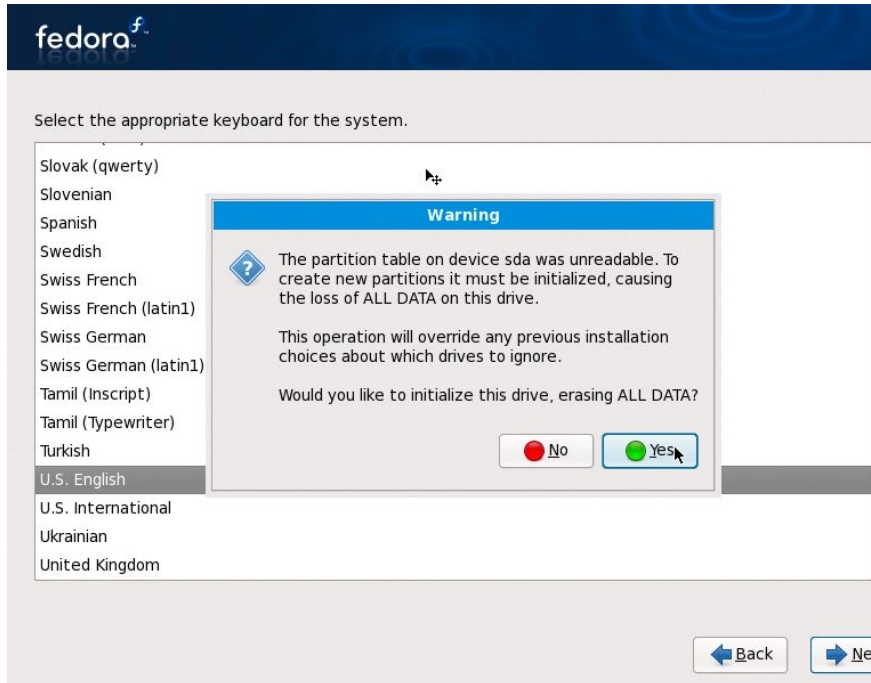


5. Selanjutnya kita harus memilih skema keyboard untuk pengoperasian proses instalasi. Dalam hal ini pilihlah skema *U.S English*. Kemudian langsung saja klik tombol Next.



6. Asumsi yang ditunjukkan pada panduan instalasi di sini adalah dalam hardisk komputer belum terinstal system operasi apapun. Installer akan tahu hal ini dan mengajukan usulan pemformatan hardisk. Sehingga pada langkah ini kita bisa mengklik opsi Yes.





7. Pada step ini, kita bisa melakukan konfigurasi ip address untuk konektifitas network dan nama hostname dari fedora linux box nantinya. Jika lingkungan kita memang mengharuskan menggunakan ip statis, maka tinggal kita ubah menggunakan opsi *Edit* pada bagian *Network Devices*.

fedora

9

Network Devices

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix
<input checked="" type="checkbox"/>	eth0	DHCP	Auto
<div></div>			

Edit

Hostname

Set the hostname:

☐ automatically via DHCP

☒ manually  (e.g., host.domain.com)

Miscellaneous Settings

Gateway:

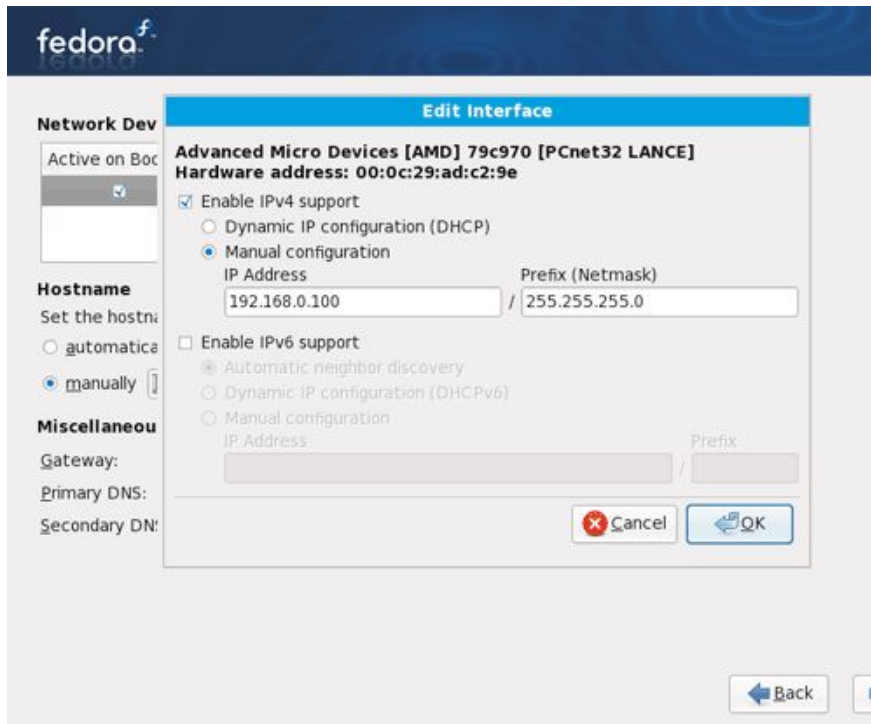
Primary DNS:

Secondary DNS:

Back

Next

8. Fedora Linux 9 sudah mendukung konfigurasi IPv6, namun pada step ini kita menggunakan konfigurasi IPv4 dengan tipe konfigurasi manual. Terlihat pada *screenshot* di bawah, kita tinggal mengisi IP address dan netmask. Langkah ini harus disesuaikan dengan kondisi environment sebenarnya. Jika sudah yakin dengan konfigurasi ini, klik tombol Ok.



9. Langkah selanjutnya kita tinggal mengisi konfigurasi gateway dan DNS. Jika sudah yakin dengan langkah ini, klik tombol Ok untuk menuju langkah selanjutnya.

**fedora**<sup>f</sup>

**Network Devices**

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix
<input checked="" type="checkbox"/>	eth0	192.168.0.100/24	Disabled

**Hostname**

Set the hostname:

☐ automatically via DHCP

☒ manually  (e.g., host.domain.com)

**Miscellaneous Settings**

Gateway:

Primary DNS:

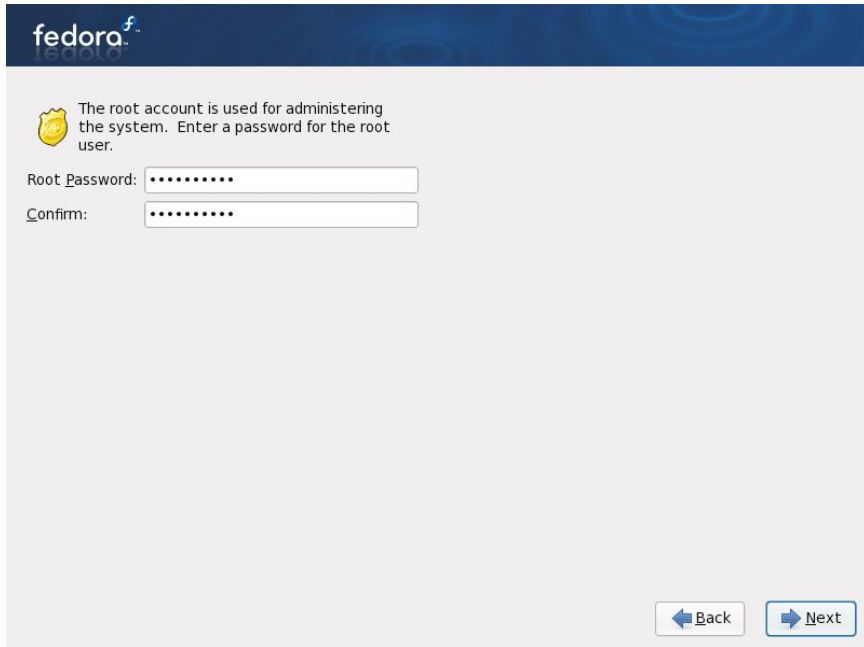
Secondary DNS:

[< Back](#) [Next >](#)

10. Pada langkah ini kita mengkonfigurasi time zone dari Fedora Linux. Installer akan menyajikan dalam bentuk peta dengan titik-titik yang bisa dipilih sesuai dengan wilayah. Klik titik yang bersesuaian dengan lokasi di mana kita berada dan kemudian menuju ke langkah selanjutnya dengan mengklik tombol Next.




11. Root merupakan super user/ user admin yang memiliki hak penuh untuk mengubah konfigurasi Fedora Linux pada nantinya. Proses instalasi mewajibkan kita untuk mendefinisikan password untuk kita penggunaan selanjutnya. Password dari user root ini harus dijaga, mengingat sangat kritikal terhadap perubahan pada Fedora Linux. Klik Next untuk melanjutkan langkah selanjutnya.



The image shows a screenshot of the Fedora Linux installer's root password setup screen. At the top, there is a dark blue header with the Fedora logo. Below the header, a yellow shield icon with a '16' inside is positioned to the left of the text: "The root account is used for administering the system. Enter a password for the root user." Below this text, there are two input fields. The first is labeled "Root Password:" and the second is labeled "Confirm:". Both fields contain a series of dots representing masked characters. At the bottom right of the screen, there are two buttons: "Back" with a left-pointing arrow and "Next" with a right-pointing arrow.

fedora<sup>f</sup>


 The root account is used for administering the system. Enter a password for the root user.

Root Password:

Confirm:

[< Back](#) [Next >](#)

12. Pada langkah ini kita harus menentukan konfigurasi manajemen disk yang akan digunakan untuk menginstal Fedora Linux. Biasanya installer akan mendeteksi sedemikian rupa dan mengaluskasikan secara otomatis. Untuk kenyamanan system penggunaan Fedora Linux, ada baiknya kita lakukan secara manual dengan mengklik tombol opsi *Advanced storage configuration*



Installation requires partitioning of your hard drive. By default, a partitioning layout is chosen which is reasonable for most users. You can either choose to use this or create your own.

Remove Linux partitions on selected drives and create default layout

▼

☐ Encrypt system

Select the drive(s) to use for this installation.

☒ sda 10237 MB VMware, VMware Virtual S

⊞ Advanced storage configuration

What drive would you like to boot this installation from?

sda 10237 MB VMware, VMware Virtual S

▼

☐ Review and modify partitioning layout

⬅ Back

➡ Next

13. Selanjutnya kita akan dihadapkan untuk mengkonfigurasi drive yang akan dialokasikan untuk Fedora Linux nantinya. Pada step ini kita harus berhati-hati, terutama jika berhadapan dengan konfigurasi disk dengan beberapa partisi.

Drive /dev/sda (10237 MB) (Model: VMware, VMware Virtual S)

Free  
10240 MB

New

Edit

Delete

Reset

RAID

LVM

Device	Mount Point/ RAID/Volume	Type	Format	Size (MB)	Start	End
▼ Hard Drives						
▼ /dev/sda						
Free		Free space		10240	1	1306

☐ Hide RAID device/LVM Volume Group members

← Back

Next →

### Add Partition

Mount Point:

File System Type:

Allowable Drives: ☐ sda 10237 MB VMware, VMware Virtual S

Size (MB):

Additional Size Options

☒ Fixed size

☐ Fill all space up to (MB):

☐ Fill to maximum allowable size

☐ Force to be a primary partition

☐ Encrypt

Cancel

OK

LVM

☐ Hide RAID device

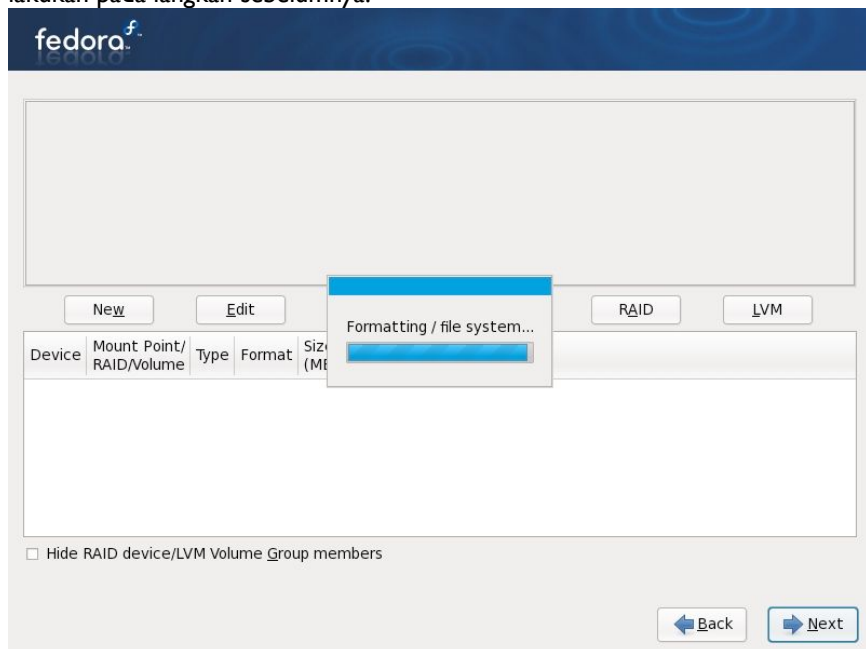
← Back

Next →



Pada langkah ini, kita harus mempartisi disk menjadi dua drive. Drive yang pertama digunakan untuk system file instalasi Fedora Linux. Drive ini diformat dengan *filesystem* ext3. Drive yang kedua digunakan untuk partisi swap memory. Biasanya swap memory berukuran 2x memori fisik. Yang dicontohkan pada gambar di bawah ini adalah drive pertama. Sesuaikan dengan kebutuhan kita untuk melakukan manajemen partisi disk. Jika sudah yakin dengan konfigurasi, klik tombol Next.

14. Program installer fedora akan memformat partisi disk yang sudah kita lakukan pada langkah sebelumnya.



15. Hasil pemformatan akan ditampilkan pada step ini. Kita bisa melakukan perubahan dengan menekan tombol Edit. Pada contoh ini, kita hanya memformat satu drive partisi. Jika anda sudah yakin dengan perubahan yang sudah ada, klik tombol Next.

fedora<sup>f</sup>  
leddo

☒ Install boot loader on /dev/sda. [Change device](#)

☐ Use a boot loader password [Change password](#)


**Boot loader operating system list**

Default	Label	Device
<input checked="" type="checkbox"/>	Fedora	/dev/sda1

[Add](#)  
[Edit](#)  
[Delete](#)

[Back](#) [Next](#)

16. Langkah selanjutnya kita akan dihadapkan dengan pilihan paket yang akan melengkapi Fedora Linux. Secara umum semua paket ada pada program installer akan terbagi menjadi lima; paket produktifitas, paket untuk *developer*, paket server web, paket tambahan, dan Fedora Linux itu sendiri. Untuk melihat lebih detail paket yang tersedia, pilih Costumize now dan klik Next.





The default installation of Fedora includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?

- ☒ Office and Productivity
- ☐ Software Development
- ☐ Web server



Please select any additional repositories that you want to use for software installation.

- ☐ Additional Fedora Software
- ☒ Fedora

 Add additional software repositories    Modify repository

You can further customize the software selection now, or after install via the software management application.

☐ Customize later   ☒ Customize now

 Back    Next

17. Dan kemudian kita akan disajikan detail dari paket2x yang ada pada Fedora Linux 9. Paket-paket standar akan secara otomatis sudah terpilih. Pilihlah paket2x tambahan sesuai dengan kebutuhan kita. Jika sudah yakin, jangan ragu untuk mengklik tombol Next.

## Desktop Environments

**Applications**  
**Development**  
**Servers**  
**Base System**  
**Languages**

## ☒ GNOME Desktop Environment

☐ KDE (K Desktop Environment)  
☐ Window Managers  
☐ XFCE

GNOME is a powerful graphical user interface which includes a panel, desktop, system icons, and a graphical file manager.

47 of 50 optional packages selected

Optional packages

Back

Next

## Desktop Environments

**Applications**  
**Development**

## ☒ GNOME Desktop Environment

☐ KDE (K Desktop Environment)  
☐ Window Managers

### Packages in GNOME Desktop Environment

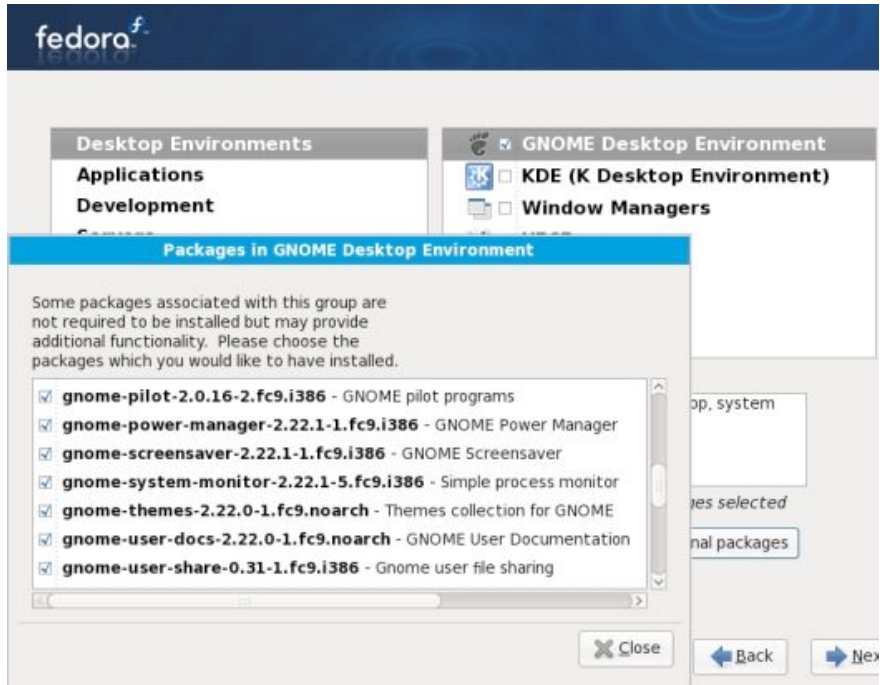
Some packages associated with this group are not required to be installed but may provide additional functionality. Please choose the packages which you would like to have installed.

- ☒ **gnome-pilot-2.0.16-2.fc9.i386** - GNOME pilot programs
- ☒ **gnome-power-manager-2.22.1-1.fc9.i386** - GNOME Power Manager
- ☒ **gnome-screensaver-2.22.1-1.fc9.i386** - GNOME Screensaver
- ☒ **gnome-system-monitor-2.22.1-5.fc9.i386** - Simple process monitor
- ☒ **gnome-themes-2.22.0-1.fc9.noarch** - Themes collection for GNOME
- ☒ **gnome-user-docs-2.22.0-1.fc9.noarch** - GNOME User Documentation
- ☒ **gnome-user-share-0.31-1.fc9.i386** - Gnome user file sharing

Close

Back

Next



18. Setelah mengklik tombol Next, program installer Fedora Linux akan melakukan pengecekan dependensi dari semua paket yang sudah kita pilih. Proses ini akan meresolve paket-paket apa saja yang dibutuhkan oleh paket-paket yang sudah kita pilih. Kemudian proses instalasi Fedora Linux akan dimulai. Biasanya proses ini memakan waktu 30 menit sampai 1 jam. Tergantung jumlah paket-paket yang kita pilih untuk diinstal.

### Desktop Environments

**Applications**  
**Development**  
**Servers**  
**Base System**  
**Languages**



☒ **GNOME Desktop Environment**



☐ **KDE (K Desktop Environment)**



☐ **Window Managers**



☐ **XFCE**

Checking dependencies in  
packages selected for installation...



GNOME is a powerful graphical user interface, desktop, system  
icons, and a graphical file manager.

47 of 50 optional packages selected

Optional packages

← Back

Next →

fedora™



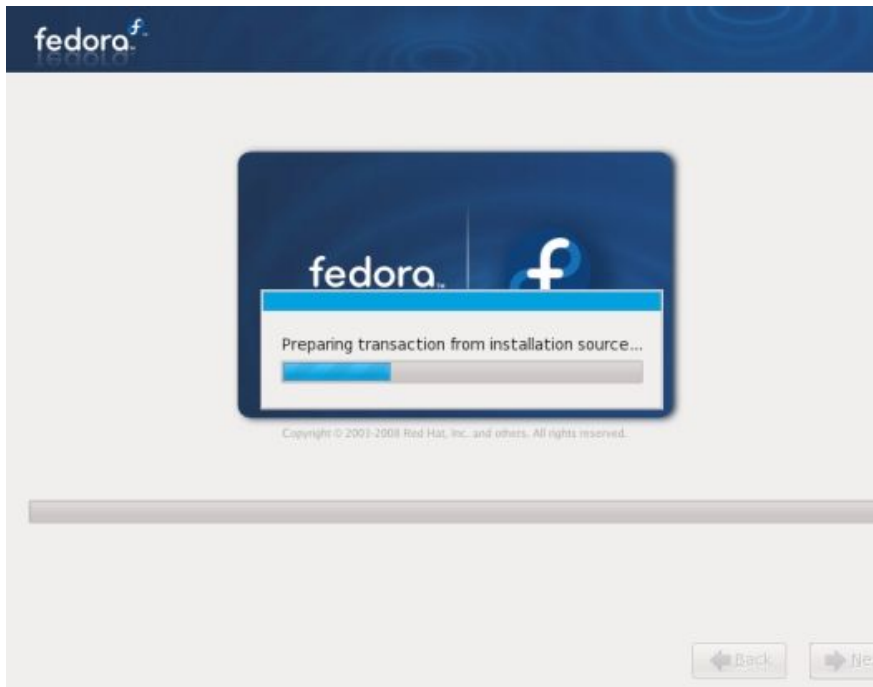
Starting install process. This may take several minutes...



Copyright © 2003-2008 Red Hat, Inc. and others. All rights reserved.

← Back

Next →



ketika proses instalasi dimulai, kita bisa memperhatikan detail proses seperti gambar di bawah ini.





Copyright © 2003-2008 Red Hat, Inc. and others. All rights reserved.

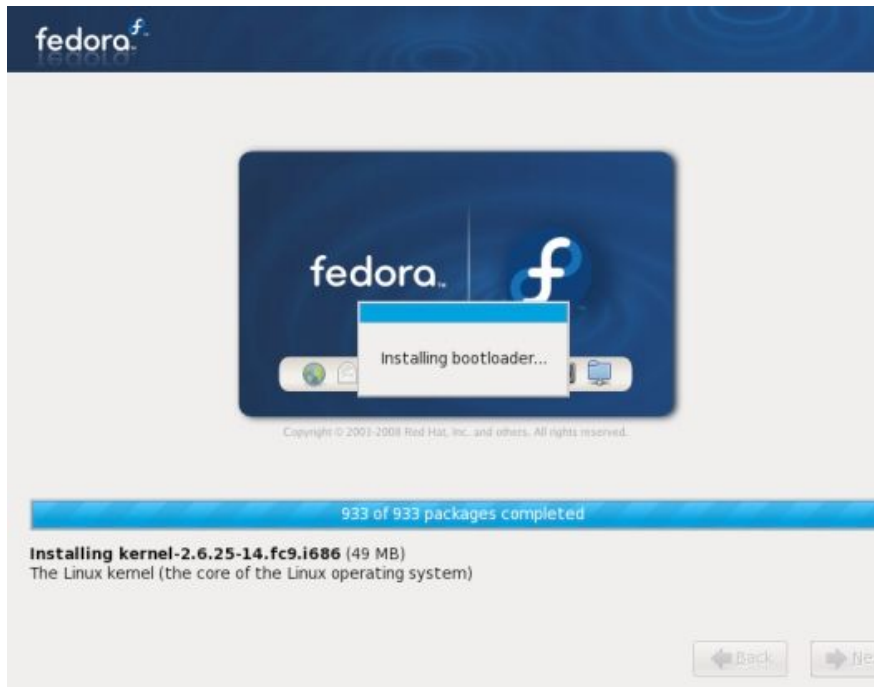
449 of 933 packages completed

**Installing python-2.5.1-25.fc9.i386** (15 MB)

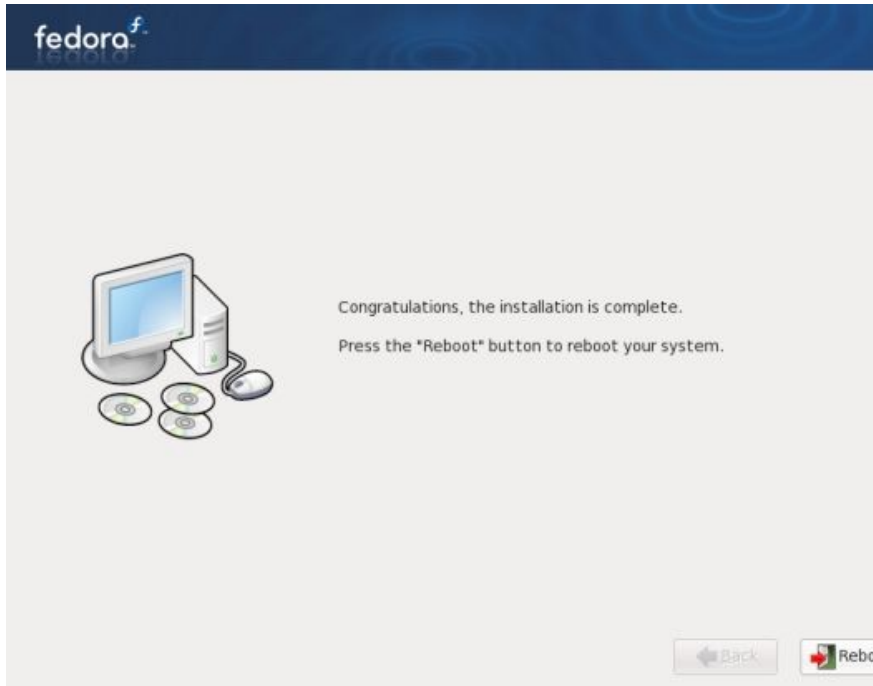
An interpreted, interactive, object-oriented programming language.

← Back

Next →



19. ketika proses instalasi telah selesai, kita harus melakukan proses reboot dengan menekan tombol *Reboot*. Namun sebelumnya jangan lupa keluarkan CD instalasi Fedora Linux



### **I.2.2.2 Konfigurasi Pasca Instalasi**

Panduan ini merupakan tahapan setelah kita menyelesaikan setup Fedora Linux dan pasca restart komputer. Pada intinya, step-step di bawah ini hanya akan berjalan ketika kita memulai Fedora Linux pada saat pertama kali.

1. Pada saat komputer booting untuk pertama kalinya-setelah proses restart instalasi. Grub akan ditampilkan pertama kali. Grub merupakan boot loader yang berada di awal partisi dari hardisk. Grub mengatur menu booting pemilihan system operasi dan sangat berguna terutama ketika kita menginstall banyak sistem operasi. Secara default, grub akan memandu kita untuk memulai Fedora Linux. Namun jika kita membutuhkan pilihan detail untuk booting, tekanlah tombol enter pada keyboard.

Press any key to enter the menu

Booting Fedora (2.6.25-14.fc9.i686) in 2 seconds... █

fedora™

2. Selanjutnya kita dihadapkan dengan layar yang menunjukkan proses booting Fedora Linux. Pada tahap ini Fedora Linux akan melakukan deteksi perangkat keras, dan meload semua modul-modul yang dimilikinya. Jika berhasil, kita akan sukses untuk mengoperasikan fedora secara optimal.



3. Ketika semua berhasil dikenali oleh Fedora, layar *welcome* akan menyapa kita untuk memulai konfigurasi user, waktu sistem, dan *profiling* perangkat keras. Kemudian kliklah tombol *Forward* untuk menuju tahap selanjutnya.



## Welcome

There are a few more steps to take before your system is ready to use. The Setup Agent will now guide you through some basic configuration. Please click the "Forward" button in the lower right corner to continue.



← Back

→ Forward

4. Tahap ini berisi lisensi Fedora linux, yaitu GPL (GNU Public License). Tahap ini hanya menjelaskan bahwa Fedora linux menggunakan lisensi open source GPL. Kliklah tombol *Forward* untuk menuju layar konfigurasi selanjutnya.



5. Fedora linux merupakan sistem operasi yang penggunaan multiuser pada saat bersamaan. Biasanya ini sangat berguna jika memang ditujukan sebagai server. Pada tahap ini kita melakukan konfigurasi user untuk pertama kalinya. User inilah nantinya akan digunakan untuk mengoperasikan Fedora linux. Seperti yang tampak pada layar di bawah, sesuaikan dengan kondisi yang anda mau untuk pada tiap-tiap field. Baik *username*, *fullname*, dan *password*. Seperti biasa, klik tombol Next untuk melanjutkan ke tahap selanjutnya.


Welcome

License Information

► Create User

Date and Time

Hardware Profile



## Create User

It is recommended that you create a 'username' for regular (non-administrative) use of your system. To create a system 'username,' please provide the information requested below.

Username:

Full Name:

Password:

Confirm Password:

If you need to use network authentication, such as Kerberos or NIS, please click the Use Network Login button.

6. Pada bagian ini kita diwajibkan untuk melakukan konfigurasi terhadap waktu system yang akan digunakan pada Fedora linux nantinya. Terdapat tiga bagian terhadap konfigurasi bagian ini; *date&time, network time protocol, time zone*





# Date and Time

Please set the date and time for the system.

Date & Time

Network Time Protocol

Time Zone

Date

< May > < 2008 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Time

Current Time : 08:42:59

Hour : 8

Minute : 29

Second : 30

Back

Forward

pada bagian ini kita bisa mengkonfigurasi linux Fedora agar menyesuaikan date & time dengan time server. Biarkan saja jika memang tidak terkoneksi dengan time server.



# Date and Time

Please set the date and time for the system.

Date & Time

Network Time Protocol

Time Zone

Your computer can synchronize its clock with a remote time server using the Network Time Protocol

☐ Enable Network Time Protocol

NTP Servers

0.fedora.pool.ntp.org

1.fedora.pool.ntp.org

2.fedora.pool.ntp.org

+ Add

Edit

Delete

► Show advanced options

← Back

Forward →

pada bagian ini kita diwajibkan menyesuaikan dengan kondisi daerah waktu sebenarnya.



# Date and Time

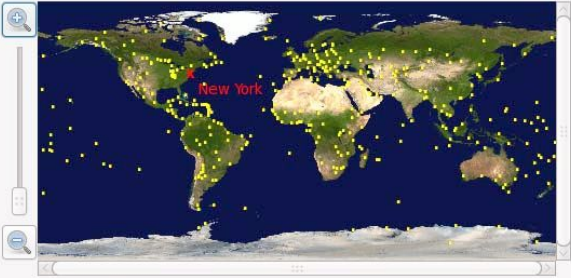
Please set the date and time for the system.

Date & Time

Network Time Protocol

Time Zone

Please select the nearest city in your time zone:



Selected city: New York, America (Eastern Time)

New York	Eastern Time
Nipigon	Eastern Time - Ontario & Quebec - places that
Nome	Alaska Time - west Alaska

☒ System clock uses UTC

Back

Forward

7. Fedora linux sudah dilengkapi dengan berbagai macam *driver* perangkat keras. Kalaupun kita menggunakan perangkat jenis baru dan belum ada drivernya, Fedora akan mencoba *resolve* dengan *driver* tipe generic. Pada bagian ini semua konfigurasi perangkat keras yang dikenali oleh Fedora akan di tampilkan. Fedora linux melakukan *profiling* engine yang disebut dengan *smolt*. Jika memang linux kita terkoneksi dengan internet, kita bisa memilih opsi send profile ke situs Fedora project, sehingga jika terdapat update driver baru, kita akan mendapat manfaat update secara langsung. Sebenarnya bisa kita abaikan. Klik Finish untuk mensudahi proses instalasi.

Welcome


License


Information

Create User

Date and Time

➤ Hardware Profile





## Hardware Profile

Smolt is a hardware profiler for The Fedora Project. Submitting your profile is a great way to give back to the community as this information is used to help focus our efforts on popular hardware and platforms. Submissions are anonymous. Sending your profile will enable a monthly update.

UUID: fd1cf605-3bed-4894-ba51-be749fd40b33

OS: Fedora release 9 (Sulphur)

Default run level: 5

Language: en\_US.UTF-8

Platform: i686

BogoMIPS: 4401.27

CPU Vendor: GenuineIntel

CPU Model: Intel(R) Core(TM)2 CPU 6400 @ 2.13GHz

Number of CPUs: 2

CPU Speed: 2174

System Memory: 1010

System Swap: 2047

Vendor: VMware, Inc.

System: VMware Virtual Platform None

Form factor: unknown

Kernel: 2.6.25-14.fc9.i686

SELinux Enabled: True

SELinux Policy: targeted

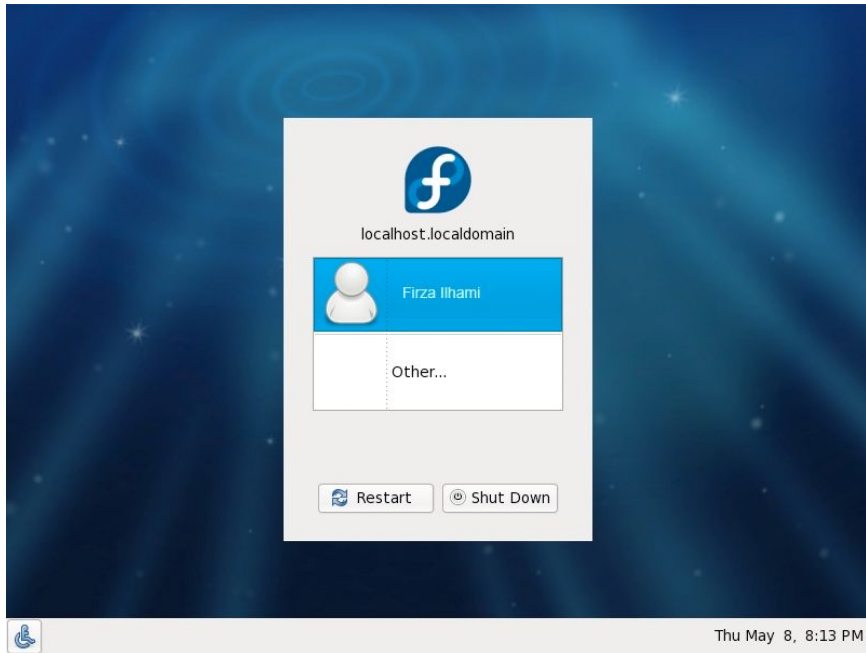
☐ Send Profile

☒ Do not send profile

Back

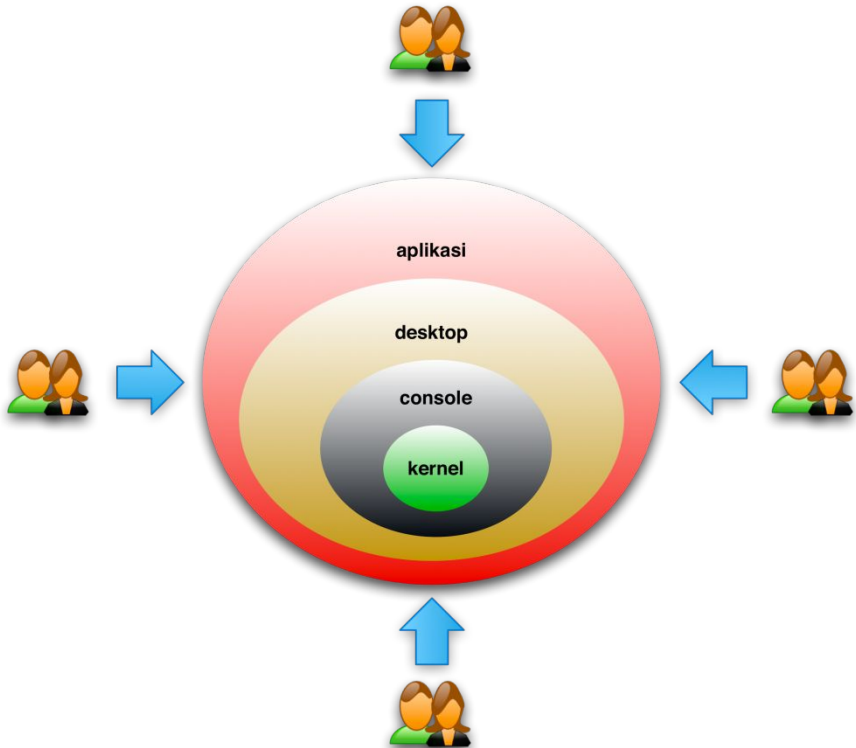
Finish

8. *login manager* (GDM – Gnome Desktop Manager) akan ditampilkan. User yang sudah kita buat tadi secara otomatis akan muncul pada form login manager. Fedora linux secara default menggunakan GNOME sebagai windows manager gui.



### I.3 Komponen Linux

Arsitektur sistem linux sangat menyerupai unix. Secara umum arsitektur Linux bisa dilihat pada gambar di bawah ini.



### 1.3.1 Kernel

Kernel merupakan jantung dari GNU/Linux. Semua fasilitas pemrosesan berpusat pada kernel. Berpusat di sini bahwa kernel mengatur sumber daya yang dialokasikan untuk menyimpan data pada memori, melakukan print, mengatur peletakkan file, mengatur proses yang sedang berjalan, mengatur bagaimana sebuah perangkat keras yang baru terpasang bisa dikenali dengan baik, dll.

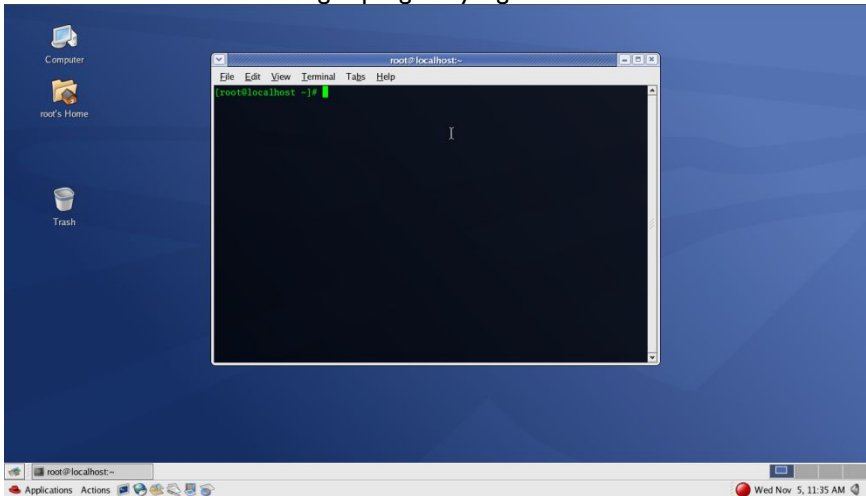
Kernel menentukan siapa yang harus menggunakan suatu *resource*, untuk berapa lama, dan kapan. Kernel yang menjalankan xmms sehingga kita bisa mendengarkan file audio mp3. Kernel mengeksekusi semua file biner dari program. Kernel bertindak sebagai mediator antara perangkat keras komputer dengan berbagai macam program/aplikasi/shell. Secara garis besar, ketika kernel linux sudah *terload* dalam residen memori, ia akan melakukan fungsi-fungsi sebagai berikut:

- Manajemen I/O
- Manajemen proses

- Manajemen perangkat keras
- Manajemen file
- Manajemen memori

### 1.3.2 Console (Shell)

*Console* atau sering disebut dengan *shell* merupakan *command-line interpreter*. Untuk melakukan segala sesuatu harus dengan mengetikkan baris perintah. Pada sistem operasi windows shell di sini mirip dengan *command prompt*. Kita bisa membuka shell linux dengan program yang bernama *terminal*.



Dalam perkembangannya, shell sendiri banyak sekali jenisnya, namun dilihat dari segi kuantitas pemakai, berikut ini merupakan jenis shell yang populer pada linux:

- bourne again shell (bash)  
Bash dikembangkan oleh Brian Fox dan Chet Ramey, dan merupakan pengembangan dari bourne shell.
- TCSH shell (tcsh)  
TCSH merupakan c shell yang dilengkapi dengan fitur *file name completion* dan *command line editing*.
- korn shell (ksh)  
Dikembangkan pertama kali oleh David Korn di AT & T Bell Labs.
- c shell (csh)  
Dikembangkan pertama kali oleh Bill Joy. C shell menggunakan sintaks yang sangat sama dengan pemrograman bahasa C

Gunakanlah perintah di bawah ini untuk mengetahui tipe shell apa yang bisa kita gunakan pada linux:

```
$ cat /etc/shells
```

Sedangkan untuk mengetahui tipe shell yang sedang kita gunakan, gunakanlah perintah berikut ini:

```
$ echo $SHELL
```

### **1.3.3 Linux Desktop**

Pada perkembangannya desktop Linux memang berpatokan terhadap Windows. Namun ini tidak berarti inovasi yang ada pada linux hanya itu-itunya saja. Linux menyediakan banyak alternatif *engine* desktop yang dapat kita gunakan, seperti GNOME, KDE, Xfce, blackbox, Enlightenment, Equinox, dll. Namun dari sekian desktop tersebut, yang paling populer sampai saat ini adalah GNOME dan KDE (K Desktop Environment). Perbedaan dari dua desktop manager ini lebih kepada usability, program pendukung, dan *user interface*. KDE lebih banyak menyajikan opsi-opsi konfigurasi. Sedangkan GNOME (GNU Network Object Model Environment) lebih fokus pada kesederhanaan.

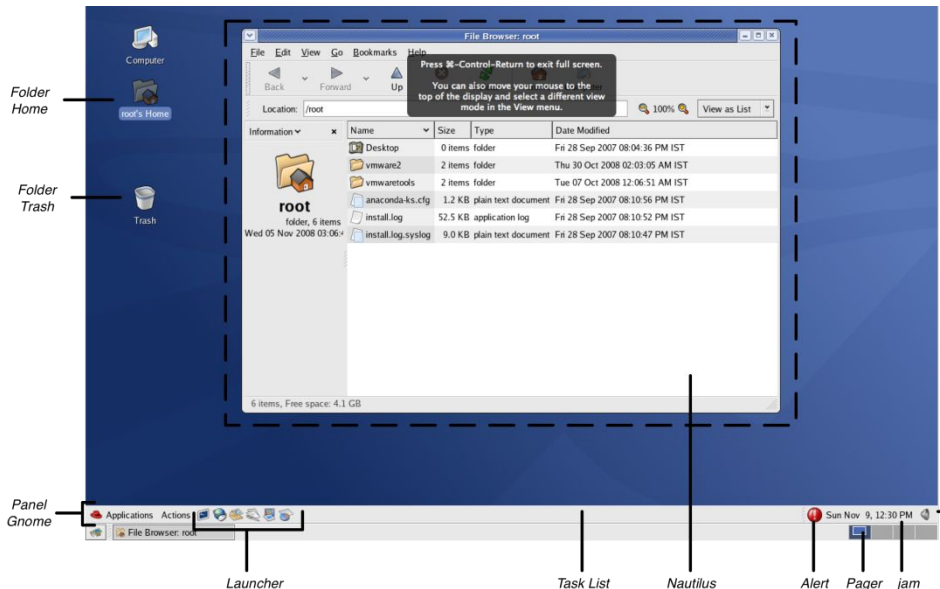
#### **1.3.3.1 Pengenalan Desktop Environment**

Pada awalnya memang GNU/Linux hanya berjalan dengan mode text. Seiring dengan banyaknya aplikasi GNU berbasis grafis yang dipaketkan, maka akhirnya GNU/Linux dilengkapi dengan *windows manager*. Windows manager inilah yang mengatur

Windows manager berjalan di atas service yang bernama X Server. X Serverlah yang mengatur penggunaan resource mouse, keyboard, kartu grafis.

#### **1.3.3.2 GNOME**



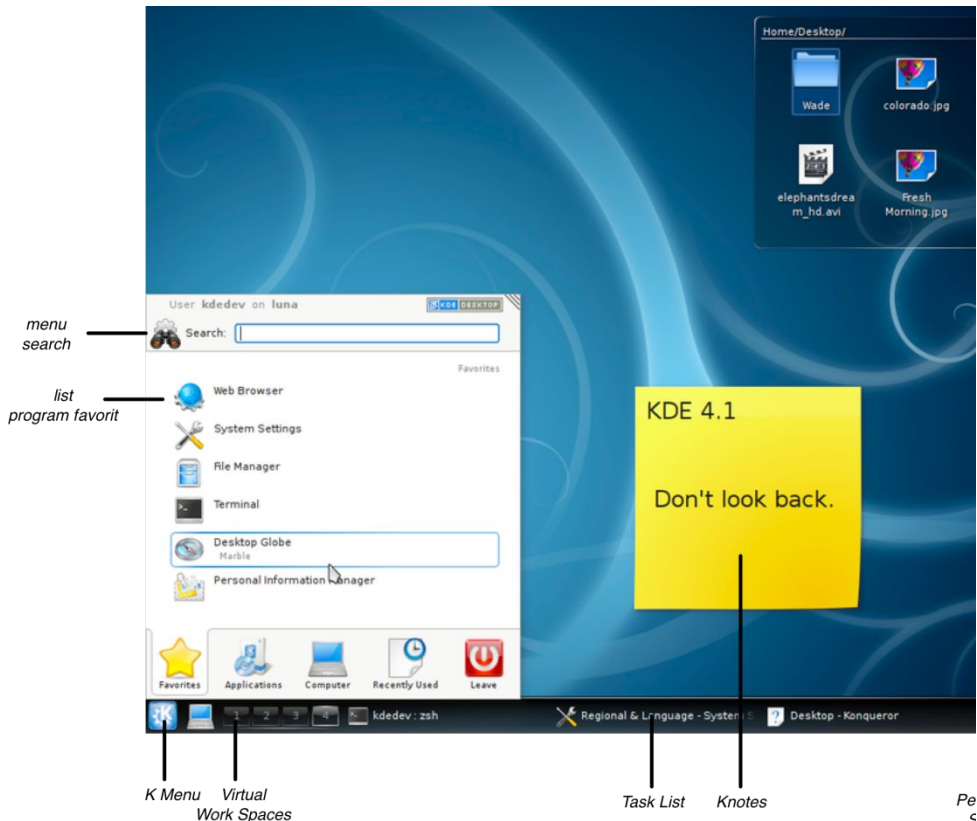


GNOME merupakan desktop alternatif setelah KDE. GNOME muncul setelah banyak respon terhadap KDE yang menggunakan Qt, sebuah *widget toolkit* third party yang dipertimbangkan sebagai non *free software*. Project GNOME dimulai oleh Miguel de Icaza dan Federico Mena. Sampai saat ini versi terkini dari GNOME adalah 2.24 yang *release* pada bulan September 2008 lalu. Seperti yang terlihat pada gambar di atas, bahwa desktop GNOME tidak begitu berbeda dengan GUI dari windows. GNOME menggunakan nautilus sebagai *file manager*. Tampilannya yang sederhana menyebabkan pengoperasiannya sangat intuitif, bahkan jika dibandingkan dengan KDE.

Perbedaan yang mencolok antara desktop windows dan GNOME terutama pada fasilitas *pager*. Pager merupakan desktop virtual yang membuat seolah-olah kita memiliki 4 desktop berbeda. Ini sangat berguna untuk dimanfaatkan ketika kita membuka banyak aplikasi. Selebihnya GNOME juga memiliki folder, trash yang serupa dengan recycle bin, nautilus yang serupa dengan windows explorer, task list yang serupa dengan taskbar pada windows, dll.

### 1.3.3.3 KDE

KDE diperkenalkan pertama kali oleh Matthias Ettirich pada tahun 1996. Pada awalnya KDE merupakan desktop yang ditujukan sebagai alternatif sebuah desktop unix yang bernama CDE, Common Desktop Environment, yang menggunakan X11 sebagai dasar *engine* desktopnya.



Pada umumnya, pengoperasian lingkungan desktop KDE tidak begitu berbeda dengan GNOME. Hampir kebanyakan fitur KDE terdapat pada GNOME, dengan sedikit perspektif yang berbeda. Jika GNOME menggunakan nautilus, KDE menggunakan konqueror sebagai file manager.

Perbedaan paling mencolok yang membedakan antara GNOME dan KDE adalah bahwa KDE dilengkapi dengan lebih banyak aplikasi dibandingkan dengan KDE. Berikut ini merupakan list aplikasi bawaan dari KDE:

- Konqueror – web browser, file manager
- Amarok – audio player
- Dolphin – file manager
- K3b – aplikasi CD/DVD burning
- KOffice – padananan Microsoft Office pada KDE
- Kopete – aplikasi chatting pada KDE
- Konsole – terminal pada KDE

- Kate – text editor

## **I.4 Latihan**

## II. NAVIGASI FILESYSTEM PADA LINUX

### II.1 Filesystem

Filesystem merupakan suatu format internal yang mengatur dan mengorganisasikan data pada suatu partisi harddisk. GNU/Linux mensupport berbagai jenis filesystem. Ada beberapa hal yang paling membedakan dengan sistem operasi windows, yaitu sifat filesystem GNU/Linux *case sensitive*, dimana huruf besar dan kecil sebagai karakter yang berbeda dan menggunakan system hierarki dalam struktur filesystemnya.

#### II.1.1 Journaling Filesystem

Filesystem journaling memiliki mekanisme untuk mencatat semua hal yang dilakukan pada filesystem pada log. Sehingga sebelum perubahan dilakukan pada filesystem, akan ada sebuah penanda pada entri journal sebagai awal akan adanya perubahan. Setelah perubahan selesai dilakukan, maka filesystem journal akan kembali menandai sebagai sesuatu aktifitas yang berhasil. Jika gagal, maka status entri journal tersebut akan dikembalikan seperti sebelum adanya aktifitas perubahan. Mekanisme ini meniru konsep transaction pada SQL.

##### II.1.1.1 EXT2

Filesystem ext2 merupakan populer generasi awal yang digunakan oleh Linux. EXT2 didesain pada awalnya untuk memperbaiki kekurangan dari filesystem EXT. filesystem ini merupakan hasil karya dari Theodore Ts'o, Remy Card, Stephen Tweedie.

Ada beberapa fitur yang menjadi andalan filesystem ini:

- mendukung berbagai tipe file standar UNIX.
- Dengan modifikasi pada layer VFS, filesystem ini mendukung ukuran sampai dengan 4T
- Mendukung nama file sampai dengan 1012 karakter
- Dapat mengalokasikan 5% dari total ukuran untuk bisa digunakan hanya oleh super user sebagai bantuan ketika ada masalah.
- Ukuran blok yang bersifat variabel
- File dapat dihapus dengan aman
- Mendukung *flag immutable* pada file, sebagai penanda tidak dapat diubah ataupun dihapus oleh siapapun
- Memiliki *hook* yang dapat ditambahkan banyak fitur, seperti *online compression*, *undelete*, dan lain sebagainya. Dengan adanya hook ini, kita bisa mengextend fitur dari filesystem sesuai dengan kebutuhan.

Setiap partisi system EXT2 terbagi atas *group* yang hanya dapat diatur pada saat pemformatan. Setiap *group* terbagi atas *superblock*, setiap *superblock* terbagi atas *group descriptor*, setiap *group descriptor* terbagi atas *block bitmap*, setiap *block bitmap* terbagi atas *inode bitmap*, setiap *inode bitmap* terbagi atas *data block*.

Untuk proses membaca dan penulisan data, EXT2 mengalokasikan beberapa blok secara berurutan, sehingga lebih cepat. EXT2 juga dapat diresize dengan aman secara offline ataupun on the fly. Untuk offline menggunakan utility *ext2resize*, sedangkan untuk on the fly menggunakan *ext2online*.

Dari semua kelebihan yang ada EXT2 juga memiliki kelemahan, terutama berkaitan dengan penggunaan system blok. Satu blok pada EXT2 berukuran 1024 byte, maka jika terdapat file yang berukuran 1026 byte maka akan disimpan pada dua blok. Padahal blok kedua hanya akan berisi 2 byte, dan tidak bisa ditambah lagi. Jika ada data baru, maka akan disimpan pada blok selanjutnya. Ini akan menyebabkan banyak space yang terbuang. Pada umumnya kelemahan ini dapat diakali dengan mengubah ukuran blok, namun tetap saja efisiensi tidak dapat tuntas secara mutlak.

### II.1.1.2 EXT3

EXT3 merupakan penyempurnaan dari generasi filesystem EXT2 oleh DR. Stephen Tweedie. Penyempurnaan di sini memang karena EXT3 adalah EXT2 yang ditambahkan fitur *journaling* dan beberapa fitur lainnya.

EXT3 memiliki sifat kompatibilitas ke bawah dengan EXT2, dan pengguna EXT2 bisa melakukan konversi ke EXT3 dengan memanfaatkan program *tune2fs* (dengan opsi *-j*) dan setelah itu dapat di remount sebagai EXT3. Namun tentu saja, sebelum melakukan hal ini modul EXT3 pada kernel harus diaktifkan terlebih dulu. EXT3 didukung sejak versi kernel 2.4.16.

Berikut ini merupakan kelebihan filesystem EXT3:

- *Backward* dan *forward* kompatibel dengan EXT2. EXT3 bahkan bisa dimount sebagai EXT2 tanpa harus menghapus fitur *journaling*.
- Sifat open dan tidak tergantung pada vendor tertentu
- Dilengkapi dengan fitur *journaling*. Terdapat dua mode *journaling*: pertama, *journaling* untuk semua metadata dan data, kedua, mode *journaling* yang hanya untuk metadata
- Dilengkapi dengan utility yang sudah sangat matang, seperti *e2fsck*.

### II.1.1.3 ReiserFS

Reiserfs diperkenalkan pertama kali oleh Hans Reiser (*namesys*) dan merupakan filesystem *journaling* pertama pertama yang dimasukkan ke kernel

linux sejak kernel 2.4.13. Reiserfs juga merupakan filesystem default dari GNU/Linux SuSE. Sejak reiserfs versi 4.x, pengembangannya disponsori langsung oleh DARPA dan memiliki standar keamanan militer amerika serikat. Reiserfs 3.x memanfaatkan *balanced tree*, dimana reiserfs akan melakukan kunjungan tree lengkap untuk setiap 4K blok yang ditulis, dan kemudian memasukkan satu pointer ke tree tersebut. Dengan kata lain, setiap penulisan sebanyak 4K, akan selalu ada mekanisme *balancing* pada tree (pemindahan data). *Balanced tree* selalu menyimpan metadata dan data, sehingga ini menyebabkan reiserfs kalah cepat dengan XFS dalam menangani file-file besar.

Sejak reiserfs versi 4.x, *balanced tree* sudah tidak digunakan lagi, dan algoritma penggantinya adalah *dancing tree*.

Reiserfs versi 4.x memiliki kelebihan fitur sebagai berikut:

- Filesystem tercepat (menurut namesys)
  - Atomic: ya atau tidak dalam satu kesatuan. Tidak ada *performance lost* karena menemukan algoritma baru tanpa mengopikan data dua kali
  - Menggunakan *dancing tree*, bukan *balanced tree*. Sehingga menjadi lebih efisien, karena memadatkan file-file kecil secara bersama-sama, daripada membuang space.
  - Dating dengan dukungan plugin. Kita bisa menambahkan plugin ke filesistem, seperti menambahkan plugin ke dalam GIMP. tentunya, plugin akan bekerja secara low level dan sangat berpengaruh terhadap data itu sendiri.
  - Didesain untuk keamanan level militer
  - Mendukung *multiple-stream* atribut dan *live query (LQ)* milik BFS (BeOS). LQ merupakan query yang mengirim otomatis perubahan pada saat perubahan dilakukan.
  - Mendukung inheritance untuk isi file dan data statistik
- Berikut ini merupakan kekurangan dari Reiserfs:
- Bekerja lambat ketika mengolah file besar

#### II.1.2.4 Vfat

Filesystem ini merupakan filesystem terlama, dan masih sangat populer mengingat interoperabilitasnya terhadap system operasi Windows. Dukungan Vfat sudah ada sejak linux kernel 1.2x. di Linux, dukungan Vfat sangat baik dan kompatibel dengan dukungan LFN (*Long FileName*) pada windows 9x dan NT.

Berikut ini sedikit catatan tentang filesystem ini:

- Ukuran maksimal file adalah 2G.
- Beberapa masalah yang mungkin terjadi berkaitan dengan filesystem ini:

- Pada linux kernel 2.4.x, lost cluster mungkin terjadi ketika ukuran filesystem melebihi 100G. pada kernel 2.6, hal ini lebih jarang terjadi
- Butuh RAM besar untuk disk besar pada saat fsck karena dosfsck mengkopi FAT ke RAM
- Jangan lupa tambahkan opsi -F 32 pada saat pemformatan, karena default tanpa opsi ini akan dianggap FAT 12 atau 16

### II.1.2.5 XFS

XFS merupakan filesystem hebat dari SGI. Filesystem ini merupakan adaptasi dari implementasinya di Irix, karena ketika diterapkan pada linux, tim pembuat XFS harus mendesain dan menulis ulang dari awal. Untuk menggunakan XFS kita harus menggunakan linux kernel 2.4.x.

Berikut ini beberapa fitur yang dimiliki oleh XFS:

- Direct IO akan membuat pekerjaan dengan file yang berukuran besar tetap cepat
- Fast recovery
- Memiliki fitur *Extents based space management*. Untuk menyiasati fragmentasi, XFS akan melakukan alokasi nomor blok ke blok setiap penulisan
- Mendukung journaling
- Mendukung *extended attribute file*.
- Mendukung *Access Control List*
- Filesystem sudah menggunakan mode 64 bit.
- Fast transaction
- Mendukung *disk quota*.

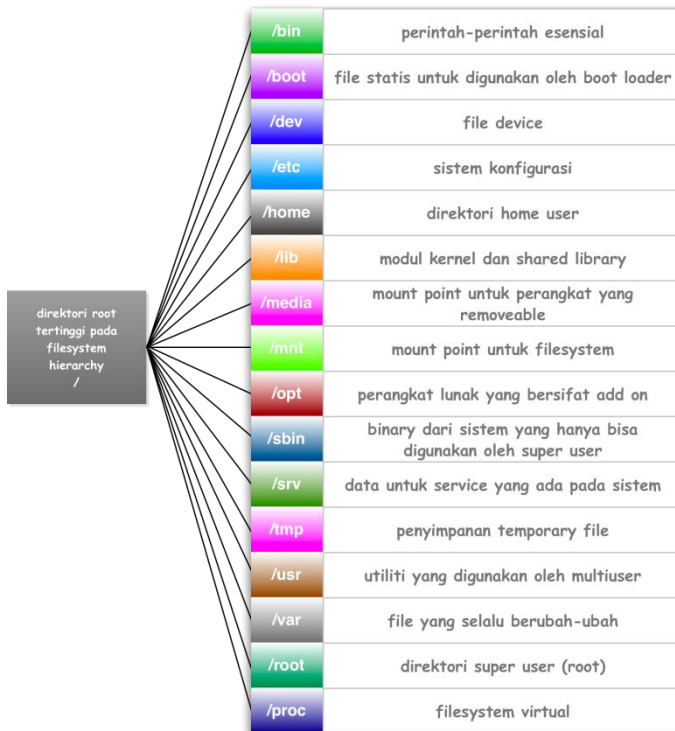
Berikut ini kekurangan dari filesystem ini:

- Tidak mendukung undelete
- LILO tidak kompatibel dengan filesystem ini, dikarenakan XFS menggunakan blok 0, yang juga digunakan LILO

### II.1.3 Perbandingan dengan Filesystem Windows

Segala sesuatu yang ada pada linux diperlakukan sebagai file. Bahkan perangkat keras pun diperlakukan sebagai file, sebagai contoh jika kita memplug sebuah hardisk baru dengan linux box kita, maka untuk menggunakan hardisk tersebut kita hanya perlu melakukan proses mounting ke dalam sebuah folder. Pada windows, perangkat keras dipisahkan terhadap filesystem, untuk mengaturnya windows menyediakan utilitas *device manager*.

Jika windows menggunakan struktur drive untuk tiap partisi *harddisk*, maka linux menggunakan system hierarki. Berikut ini struktur hierarki standar dari filesystem linux.



## II.2 Perintah-Perintah dasar Untuk Navigasi

Berikut ini merupakan perintah-perintah dasar yang digunakan untuk melakukan navigasi pada shell linux. Untuk melihat lebih jauh penggunaan dari masing-masing perintah, gunakan cara di bawah ini:

```
$ man <nama_perintah>
```

### II.2.1 *pwd*

Perintah *pwd* digunakan untuk melihat status direktori kerja saat ini, atau mudahnya untuk melihat di mana posisi kursor kita sekarang. Berikut ini merupakan sintaks perintah *pwd*

```
$ pwd
```

```
1 $ pwd          #menunjukkan direktori kerja saat ini pada direktori
   /home/wongacid
   /home/wongacid
```

### II.2.2 *cd*



Perintah `cd` digunakan untuk berpindah dari satu direktori ke direktori lain. Sintaksnya sebagai berikut:

```
$ cd <nama_direktori>
```

Ada satu hal yang menarik dalam penggunaan perintah `cd`, ketika kita ingin berpindah ke dalam direktori yang memiliki nama panjang dan kita tidak ingin mengetik terlalu lama, karena ada kecenderungan kesalahan ketik, maka cukup ketik satu atau beberapa huruf dari nama direktori yang panjang tersebut dan tekan tombol tab. Fitur ini disebut dengan *auto completion*.

```
1 $ cd wong      #kemudian tekan tab
2 $ cd wongacid  #bash shell akan melist direktori dengan nama awal wong.
    Jika hanya          #satu maka nama direktori langsung ditampilkan
```

Linux menggunakan `.` (dot) untuk mengacu kepada direktori aktif shell, dan menggunakan `..` (double dot) untuk mengacu kepada direktori *parent* dari direktori aktif shell kita. Perhatikan contoh di bawah ini.

```
1 $ pwd
   /home/wongacid
2 $ cd ..
3 $ pwd
   /home
```

### II.2.3 `ls`

Perintah `ls` digunakan untuk melihat isi dari suatu direktori. Mirip dengan perintah `dir` pada DOS. Berikut ini merupakan sintaks dari perintah `ls`.

```
$ ls [option] [namadirektori]
```

```
1 macjnun:~ macjnun$ ls #tanpa parameter akan melisting current direktori
Applications      Documents          Library
                  Music
Public            Sites              ngoprek
                  treasury-jobreport.txt
Desktop           Downloads          Movies
                  Pictures Send Registration  Stufflt          temp
2 macjnun:~ macjnun$ ls Documents/ # akan melist semua file pada direktori
Documents
About Stacks.pdf   Office Projects    Virtual
Machine Library eclipse-workspace  workspace
```

### II.2.4 `set`

Perintah ini digunakan untuk melihat dan menulis variabel pada lingkungan GNU/Linux. Berikut ini sintaks penggunaan perintah ini.

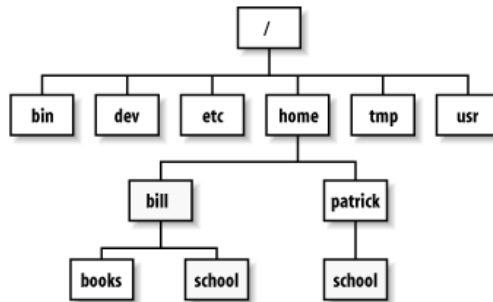
```
$ set
```

### II.2.5 export

Perintah *export* digunakan untuk menginisialisasikan suatu variabel global pada shell. Global di sini agar bisa dikenali oleh *child process* dalam lingkungan linux. Berikut ini merupakan sintaks dari penggunaan perintah *export*.

```
$ export <nama_variable>
```

### II.3 Penggunaan Path Absolut dan Path Relatif



Coba perhatikan dua direktori yang bernama *school* di atas. Direktori *school* yang pertama merupakan sub direktori dari direktori *bill*, sedangkan yang satunya lagi merupakan subdirektori dari direktori *patrick*. Untuk menghindari kebingungan terhadap dua direktori yang bernama sama ini, linux mengenal sesuatu yang disebut dengan nama *path*. Nama path mengacu suatu file dan menunjukan tempat berada suatu file, baik direktori ataupun file biasa.

Ada dua jenis path: abosolut dan relatif. Path absolut mengacu nama sebuah direktori dari *root directory* linux (/). Jadi jika kita ingin mengacu nama path dari direktori *school* yang merupakan subdirektori *bill*, akan seperti ini */home/bill/school*, sedangkan yang satunya lagi akan seperti ini */home/patrick/school*. Path relatif mengacu suatu file/atau direktori berdasarkan direktori aktif tempat kita berada pada shell. Untuk mengacu nama path dari suatu file, kita hanya cukup menggunakan referensi dari lokasi direktori aktif shell kita. Perhatikan contoh di bawah ini:

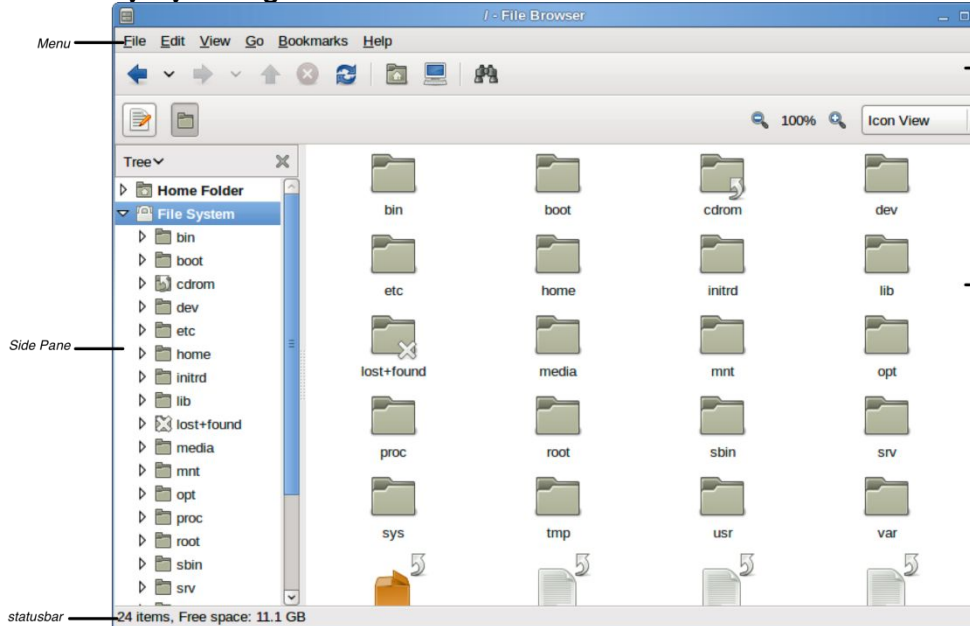
- 1 `$ pwd`  
`/home/patrick`
- 2 `$ cd school` # cara seperti ini disebut dengan path relatif
- 3 `$ cd /home/Patrick/school` # cara seperti ini disebut dengan path absolut

### II.4 Navigasi Filesystem pada Desktop Linux

Linux menyediakan berbagai cara mudah agar kita bisa melakukan manajemen file. Selain dengan perintah-perintah di atas pada terminal, GNU/Linux juga menyediakan alternatif aplikasi GUI seperti *Nautilus* pada GNOME, *Konqueror* dan *dolphin* pada KDE. Aplikasi sejenis ini kerap kali disebut dengan *File*

*Manager*. Pada Windows, kita bisa membandingkan aplikasi sejenis ini dengan windows explorer. Secara fungsional tidak jauh berbeda.

#### 11.4.1 Menjelajah dengan Nautilus



Nautilus merupakan aplikasi file manager standar pada lingkungan desktop GNOME. Nautilus memiliki dua panel utama, panel kiri dan panel kanan. Panel kiri menunjukkan posisi folder yang sedang terbuka maupun folder hirarki pada filesystem. Panel kanan menunjukkan isi dari folder yang sedang terseleksi pada panel kiri. Selain digunakan sebagai file manager, nautilus juga bisa berfungsi sebagai web browser.

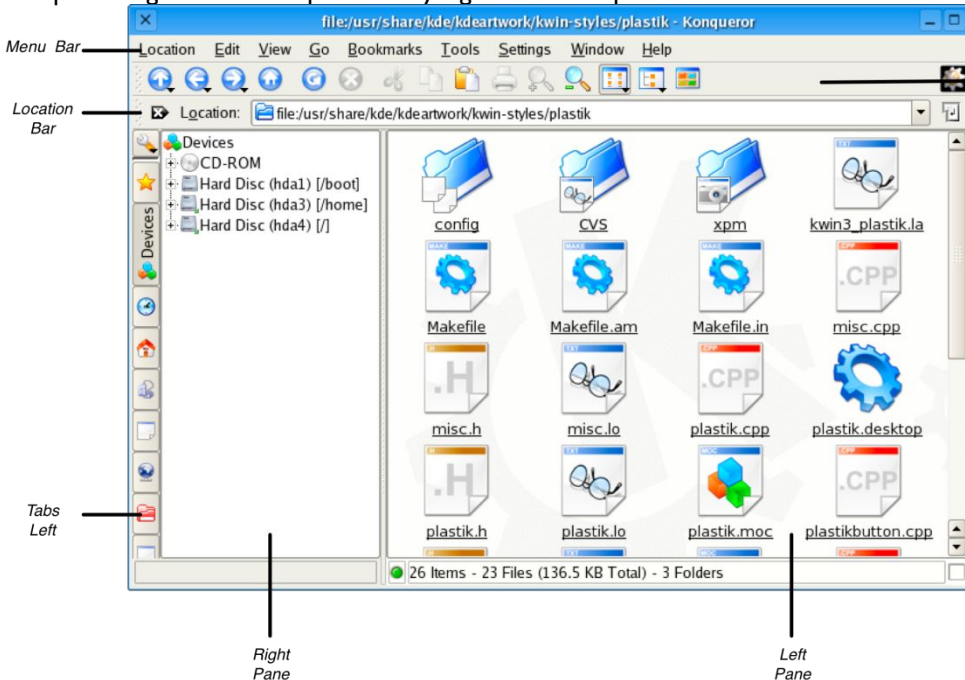
Nautilus menyediakan berbagai cara untuk dioperasikan sebagai file manager, berikut ini beberapa hal yang paling esensial:

- Untuk memindahkan file (*move*), cukup mudah hanya dengan *mendrag* file tersebut menuju lokasi baru
- Untuk menggandakan sebuah file (*copy*), tekan kontrol sambil *mendrag* file yang bersangkutan. Alternatif lain, menggunakan klik kanan dan pilih action *copy* pada menu pop up yang tampil
- Untuk mengganti nama file (*rename*), klik kanan file yang bersangkutan dan pada menu pop up, pilih action *Rename*. Setelah mengganti nama, tekan enter

- Untuk menghapus file (delete), cukup klik kanan pada file yang bersangkutan dan pada menu pop up, pilih action *Move to trash*

## II.4.2 Menjelajah dengan Konqueror

Konqueror merupakan file manager sekaligus merangkap sebagai web browser. Pada lingkungan KDE, jika kita membuka folder manapun, maka aplikasi yang akan digunakan untuk ini adalah konqueror. Alternatif lain, kita dapat membuka konqueror melalui tombol start (pada red hat) atau tombol K pada pojok bawah kiri desktop KDE. Operasi yang dapat dikerjakan oleh konqueror untuk mengatuf file tidak jauh berbeda dengan nautilus pada GNOME, yaitu move, copy, delete, rename. Secara umum, semua operasi ini sudah ada pada menu pop up klik kanan pada konqueror. Pada KDE versi 4, posisi file manager konqueror digantikan oleh aplikasi lain yang bernama dolphin.




Konqueror dilengkapi dengan layout engine KHTML yang kompatibel dengan standar HTML dan mensupport penggunaan javascript, applet java, CSS, SSL, dan standar terbuka lainnya. Konqueror juga dilengkapi dengan beberapa service pencarian yang bisa dikostumasi dan dapat diakses dengan kode singkatan tertentu, seperti gg: untuk google.

Main Page - Wikipedia, the free encyclopedia - Konqueror

Location Edit View Go Bookmarks Tools Settings Window Help

Location: http://en.wikipedia.org/wiki/Main\_Page



[main page](#)
[discussion](#)
[view source](#)
[history](#)
[watch](#)

[.wiki.pedias](#)
[my talk](#)
[my preferences](#)
[my watchlist](#)
[my cont](#)

Welcome to **Wikipedia**,  
the free encyclopedia that anyone can edit.  
2,298,529 articles in English

Arts
History
Biography
Mathematics
Geography
Science

Overview • Editing • Questions • Help

Contents • Categories • Featured c

navigation

- Main Page
- Contents
- Featured content
- Current events
- Random article

interaction


- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

search

Go Search

toolbox

- What links here
- Related changes
- Upload file



**Reese Witherspoon** is an American actress who has established herself as one of the highest-paid Hollywood actresses in recent years. Witherspoon landed her first feature role as the female lead in the movie *The Man in the Moon* in 1991; later that year she made her television acting debut, in the cable movie *Wildflower*. In 1996, Witherspoon's performance in *Freeway* established her as a rising star and led to roles in three major 1998 movies: *Overnight Delivery*, *Pleasantville*, and *Twilight*. The following year, Witherspoon appeared in the critically acclaimed *Election*, which earned her a *Golden Globe* nomination. 2001 marked her career's turning point with the breakout role as Elle Woods in the box office hit *Legally Blonde*, and in 2002 she starred in *Sweet Home Alabama*, which became her biggest commercial film success to date. 2003 saw her return as lead actress and executive producer of *Legally Blonde 2*. In 2005, Witherspoon received worldwide attention and praise for her portrayal of June Carter Cash in *Walk the Line*, which earned her an *Academy Award*, a *Golden Globe* and a

Today's featured article

In the news

- Kuomintang candidate Ma Ying-jeou (picture) is **elected** President of the Republic of China (Taiwan). Two referenda on joining the United Nations **failed** due to low turnout.
- French swimmer **Alain Bernard** sets a new world record for the men's 100 metres freestyle in 47.50 seconds.
- A permanent coalition government agreed in Belgium, ending a nine-month stalemate.
- Leterme** is sworn in as Prime Minister.
- David Paterson** is sworn in as the first black Mayor of New York (and the fourth ever in the U.S.).
- Spitzer** resigns amidst a prostitution scandal.
- The wreckages of HMAS Sydney and the HMAS **are located** off the coast of Western Australia after their mutual destruction in battle.

## BAB 3

# Operasi-Operasi Dasar Pada File dan Direktori Linux

### 3.1 Perintah Dasar Operasi File dan Direktori

Dalam lingkungan Linux terdapat perintah-perintah dasar yang membantu dalam hal Administrasi Sistem dalam hal ini operasi file dan direktori. Selalu ingat bahwa ada perintah man yang membantu kita dalam menjelaskan fungsi – fungsi dari perintah – perintah yang ada. Contoh :

```
1 $ man rm # dengan menjalankan perintah diatas maka linux akan menampilkan
           penjelasan dari perintah rm
```

#### 3.1.1 cat

Perintah cat dapat digunakan untuk menampilkan isi sebuah file.

contoh :

```
1 $ cat index.txt
```

#### 3.1.2 cp

Untuk menyalin file atau direktori kita dapat menggunakan perintah cp, contoh :

```
1 $ cp index.txt index_lain.txt # perintah diatas berarti kita akan menyalin file
                                index.txt dengan index_lain.txt
2 $ cp -R dir1 dir 2            # perintah diatas berarti kita akan menyalin
                                direktori dir1 beserta isinya dengan dir2.
```

#### 3.1.3 rm

Menghapus direktori yang mempunyai isi file atau direktori di dalamnya kita dapat menggunakan option -rf, contoh :

```
1 $ rm index.txt # perintah ini berarti akan menghapus file dengan nama index.txt
2 $ rm -rf tes   # perintah ini berarti kita menghapus direktori tes beserta
                 isinya tanpa konfirmasi
3 $ rm -ri tes   # perintah ini berarti kita menghapus direktori tes beserta
                 isinya dengan konfirmasi
```

#### 3.1.4 ln

Untuk membuat link bagi file yang memungkinkan di akses oleh nama yang berbeda. Links dapat berupa “hard” atau “soft”. Hard link membuat dua nama untuk file yang sama, sedangkan soft atau symbolic link membuat file kedua yang merupakan shortcut dari file pertama. Contoh :

```
1 $ ln -s test.txt /tmp/test2.txt # artinya membuat symbolic link untuk file
                                   test.txt di direktori tmp dengan nama
                                   test2.txt
```

#### 3.1.5 find

Perintah find digunakan untuk mencari file atau direktori. Contoh :

```
1 $ find /tmp -name test -type f # perintah ini berarti mencari semua file yang
                                   bernama test didalam direktori /tmp
```

#### 3.1.6 locate

Mencari di dalam database semua file-file yang sesuai dengan pola yang ditentukan. Contoh :

```
1 $ locate *.htm # artinya akan mencari file-file yang berekstensi .htm di
semua file system.
```

### 3.1.7 whereis

Fungsinya hampir sama dengan perintah locate yaitu untuk mencari akan tetapi whereis lebih spesifik mencari binary, source code dan manual untuk perintah atau file yang ditentukan. Contoh :

```
1 $ whereis rm # artinya akan mencari lokasi perintah rm
```

### 3.1.8 more

Perintah more dapat digunakan untuk menampilkan isi sebuah file per halaman.

contoh :

```
1 $ more index.txt
```

### 3.1.9 less

Kegunaannya seperti halnya perintah more, tapi dapat menggunakan Page Up dan Page Down untuk melihat seluruh file.

Contoh :

```
1 $ less index.txt
```

### 3.1.10 mkdir

Perintah mkdir digunakan untuk membuat direktori baru.

contoh :

```
1 $ mkdir /tmp/dir1 # berarti kita baru saja membuat direktori baru dengan nama
dir1 di dalam direktori /tmp
```

### 3.1.11 mv

Untuk mengubah nama file atau memindahkan file atau direktori kita dapat menggunakan perintah mv

contoh :

```
1 $ mv index.txt index_old.txt # yang berarti kita baru saja mengganti nama file
index.txt dengan nama baru index_old.txt
$ mv index.txt /tmp/tes # perintah diatas berarti kita baru saja memindahkan
file index.txt ke direktori /tmp/tes.
```

### 3.1.12 file

Berfungsi untuk mengklasifikasi file yang bersangkutan menurut data yang disimpannya.

Contoh :

```
1 $ file * | grep roff # perintah diatas berarti kita melihat file-file yang
dianggap input troff/nroff
```

### 3.1.13 which

Perintah which digunakan untuk menunjukkan path lengkap dari perintah shell.

Contoh :

```
1 $ which ls # Artinya akan menunjukkan dimana letak perintah ls dengan path
lengkap
```

### 3.1.14 merge

Perintah merge digunakan untuk menggabungkan file dengan tiga arah, menaruh hasilnya di file urutan pertama.

Contoh :

```
1 $ merge file1.txt file2.txt file3.txt # Yang artinya setelah di merge,
file1.txt mengandung perubahan-
perubahan dari file2.txt ke
file1.txt dan perubahan-perubahan
dari file2.txt ke file3.txt.
Tetapi tidak mengubah isi dari
file2.txt dan file3.txt
```

### 3.1.15 diff

Perintah diff digunakan untuk menampilkan perbedaan antara dua buah file

Contoh :

```
1 $ diff test1 test2      # Yang berarti mencari perbedaan antara file test1 dan
                           test2, apabila ada maka akan ditampilkan. Apabila tidak
                           ada berarti dua file tersebut identik.
```

### 3.1.16 gzip

Perintah ini digunakan untuk mengompresi sebuah file sehingga memiliki ukuran lebih kecil.

Perintah ini juga dipakai untuk mendekompresi sebuah file yang sudah dikompres sebelumnya.

Contoh :

```
1 $ gzip sample.tar      # Pada contoh di atas sebuah file bernama sample.tar
                           dikompres dengan menggunakan perintah gzip. Hasil dari
                           kompresinya dinamai sample.tar.gz.
2 $ gzip -d sample.tar.gz # Contoh di atas memperlihatkan bagaimana
                           mengembalikan file yang sudah dikompres menjadi
                           file aslinya. Ukuran hasil dekompresi pasti menjadi
                           lebih besar daripada sebelumnya
```

### 3.1.17 tar

Perintah ini digunakan untuk menggabungkan beberapa file menjadi sebuah file tunggal. Selain itu perintah ini digunakan juga untuk memecah file tar yang ada menjadi file-file aslinya.

Contoh :

```
1 $ tar -cvf sample.tar latihan.txt latihan2.txt      # Perintah diatas menggabung
                                                         file latihan.txt dan
                                                         latihan2.txt menjadi
                                                         sample.tar
2 $ tar -xvf sample.tar      # Dengan menggunakan perintah di atas, semua isi
                              sample.tar akan diekstrak di dalam direktori saat ini
```

## 3.2 Penggunaan Editor Vi

### 3.2.1 Perkenalan

Vi Editor adalah “screen based” editor yang digunakan oleh sebagian besar pengguna UNIX. Vi Editor mempunyai fitur-fitur yang handal untuk membantu para programmer, tetapi banyak pengguna pemula enggan menggunakan VI karena fitur-fitur yang membingungkan mereka. Beberapa keunggulan Vi antara lain terdapat di semua mesin UNIX dan juga menggunakan sedikit sumber daya, karena itu sangat berguna untuk mempelajarinya.

### 3.2.2 Modus Operasional Vi

Dalam pengoperasiannya Vi mempunyai dua mode : mode perintah dan mode input. Di mode perintah, karakter yang diketik mewakili suatu perintah seperti mengerakkan kursor, memotong atau menyalin teks, dan lain-lain). Di mode input, karakter yang diketik langsung masuk atau mnimpa teks yang sudah ada.

### 3.2.3 Memulai Editor Vi

Vi editor dapat digunakan untuk membuat file baru atau mengedit file yang sudah ada. Perintah untuk memulai editor Vi adalah *vi*, diikuti oleh nama file. Contoh, untuk mengedit file *temp*, cukup ketik *vi temp* kemudian tekan enter. Memulai vi dapat tanpa nama file, tetapi ketika akan menyimpan harus disebutkan nama filenya.

Ketika memulai vi untuk pertama kali akan terlihat layer di penuh oleh tilde (~) pada sisi kiri layer. Di bawah layer akan terdapat nama file apabila disebutkan, sedangkan akan tampil *newfile* apabila nama file kosong.



### 3.2.4 Perintah – Perintah Dasar pada Modus Perintah Vi

Berikut adalah referensi perintah-perintah dasar yang digunakan pada mode perintah dalam pengoperasian vi

#### 3.2.4.1 Menggerakkan Kursor

h	kiri
j	bawah
k	atas
l	kanan
^	awal baris
\$	akhir baris
1 G	awal dokumen
G	akhir dokumen
<n> G	ke baris <n>
^F	halaman berikutnya
^B	halaman sebelumnya
w	kata berikutnya
b	kata sebelumnya

#### 3.2.4.2 Menghapus Teks

Backspace	menghapus karakter sebelum kursor (hanya berlaku di mode input)
x	menghapus karakter di kursor
dw	menghapus kata
dd	menghapus baris (mengembalikan dengan p atau P)
<n> dd	menghapus n baris
d\$	menghapus sampai akhir baris
dG	menghapus sampai akhir file

#### 3.2.4.3 Copy dan Paste Teks

yy	menyalin baris (mengembalikan dengan p atau P)
<n> yy	menyalin baris (mengembalikan dengan p atau P)

#### 3.2.4.4 Membatalkan Perubahan

u	undo
:w	menyimpan file
:wq	menyimpan dan keluar
ZZ	menyimpan dan keluar
:q!	keluar tanpa menyimpan

#### 3.2.4.5 Mencari String

Editor VI memiliki dua jenis pencarian, yaitu : string dan karakter. Untuk pencarian string, perintah yang digunakan adalah perintah / dan ?. Pada saat kedua perintah ini digunakan, perintah yang diketikkan akan muncul dibagian bawah layar, tempat kita mengetikkan string yang ingin dicari.

Kedua perintah ini *berbeda hanya pada arah pencarian* yang akan dilakukan. Perintah / mencari ke arah bawah, sedangkan perintah ? mencari ke arah atas dari file. Perintah n dan N digunakan untuk melanjutkan proses pencarian yang sebelumnya. Perintah n akan mencari dalam arah yang sama, sedangkan perintah N akan mencari dalam arah yang berlawanan. Untuk melakukan pencarian suatu kata tertentu ( fixed ), bisa digunakan command seperti contoh berikut:

/\<de\>

artinya command ini hanya akan mencari string dengan nilai de saja, nilai desa, pakde, Denada akan diabaikan

#### 3.2.4.6 Mengganti String

Kita bisa juga melakukan penggantian satu string pada suatu file secara serentak atau biasa disebut *Replace-all*. Dengan menggunakan command ( tentu pada mode perintah ) :

:1,7s/balon/mobil/g

yang artinya kita ingin mengganti kata balon menjadi mobil mulai dari baris ke 1 sampai baris ke 7.

### 3.2.4.7 Pindah Layar

Editor VI juga menyediakan perintah-perintah untuk berpindah layar dengan cepat tanpa harus menggerakkan kursor perbaris. Perintah-perintah tersebut adalah :

#### Perintah Keterangan

ctrl-b	berpindah satu layar sebelumnya
ctrl-d	turun setengah bagian layar
ctrl-f	turun satu bagian layar
line no. G	memindahkan kursor ke line tertentu
Ctrl-j	naik setengah bagian layar

### 3.3 Latihan

1. Buatlah sebuah file bernama percobaan.txt dengan isi file sebaris kalimat berikut  
BELAJAR LINUX SANGAT MENYENANGKAN ?

Jawab:

touch percobaan.txt

nano percobaan.txt

Isi tulisan dengan

BELAJAR LINUX SANGAT MENYENANGKAN

save dengan cara tekan Ctrl + X dengan tekan Y untuk menyimpan

2. Amati dan catat ukuran file tersebut!

Jawab:

ls -al

ukuran isi file tersebut adalah 34 byte

3. Buatlah sebuah direktory bernama percobaan.dir ?

Jawab :

mkdir percobaan

4. Copylah file tersebut ke daalm direktory yang baru anda buat ?

Jawab :

cp percobaan.txt/home/percobaan

5. Copy sekali lagifile tersebut dengan nama percobaan new ke direktory yang baru saja anda buat ?

Jawab :

cp percobaan.txt percobaannew.txt

cp percobaannew.txt/home/percobaan

6. Hapuslah file percobaan.txt yang ada pada direktori kerja ( saat ini ) !

Jawab :

rm percobaan.txt

7. Pindahkanlah ke directori percobaan. dir !

Jawab :

cd percobaan.dir

8. Hapuslah file percobaan.txt yang ada pada directori kerja (saat ini) !

Jawab :

rm percobaan.txt

9. Ubahlah nama file percobaan.new menjadi percobaan.1 !

Jawab :

mv percobaannew.txt percobaan1.txt

10. Pindahkanlah file percobaan.1 ke directori yang berada satu tingkat di atas directori saat ini !

Jawab :

cp percobaan1.txt/home

11. Pindahlah satu tingkat di atas dan harus direktori percobaan.dir !

Jawab :

rm-r percobaan.dir

12. Amati lokasi home direktori anda ( Petunjuk gunakan variable home) !

Jawab :

cd ..

kemudian ketik perintah

pwd

ukuran file direktori home sebesar 4096 byte

13. Copykan file/bin/ls ke home direktori anda!

Jawab :

cp /bin/ls /home

14. Buatlah sebuah file kosong bernama kosong.txt

Jawab :

touch kosong.txt

15. Amati apakah file percobaan1 masih ada jika sudah tidak ada buatlah seperti soal 1 latihan 3.2 !

Jawab :

Tidak ada karena file tersebut sudah dihapus maka kita buat dengan perintah

touch percobaan.txt

16. Lakukan kompresi file ls ( hasil copy pada soal no 1) menjadi ls.tar.gz gunakan utilitas tar dan gzip !

Jawab :

tar -cf ls.tar /home

gzip ls.tar

17. Amati dan bandingkan ukuran file ls dengan ls.tar.gz, apakah ukuran file hasil kompresi lebih kecil ?

Jawab :

kita amati ukuran file ls.tar.gz dan bandingkan dengan file ls

ukuran file ls : 92376 byte

Ukuran file ls.tar.gz : 56376 byte

18. Lakukan hal serupa untuk file kosong.txt dan percobaan.1 beri nama kosong.tar.gz dan percobaan.tar-gz amati apakah file hasil kompresi selalu lebih kecil?

Jawab :

<!-- @page { size: 21cm 29.7cm; margin: 2cm } P { margin-bottom: 0.21cm } -->

untuk mengkompresi file kosong.txt menggunakan perintah :

tar -cf kosong.tar /home

gzip kosong.tar

untuk mengkompresi file percobaan.txt menggunakan perintah :

tar -cf percobaan.tar /home

gzip percobaan.tar

19. Buatlah direktori baru bernama kompres dan pindahkan filen kompresi ke direktori tersebut.

Jawab :

mkdir kompres.dir

cp percobaan.tar /home/kompres.dir

cp ls.tar.gz /home/ kompres.dir

cp kosong.tar / home/kompres.dir

20. Pindahkan ke dalam direktori kompres.

Jawab :

cd kompres.dir

21. Bukalah atau ekstrak file kompresi dan amati apakah ukuran file hasil ekstrak sama dengan aslinya!

Jawab :

tar -cf kompres.tar

gzip kompres.tar

ukuran file lebih kecil daripada sebelumnya.

# BAB IV

## Utilitas Tambahan pada Command Line Interface

---

### Tujuan:

1. Siswa mengenal utilitas-utilitas tambahan pada CLI yang sering digunakan
2. Siswa dapat menggunakan utilitas-utilitas tersebut untuk membantu proses administrasi sistem operasi
3. Siswa mengetahui fungsi dan dapat menggunakan wild card dan regular expression

### Deskripsi singkat:

Pada praktikum ini siswa akan diperkenalkan pada berbagai utilitas tambahan yang biasa digunakan pada command line interface Linux seperti bash yang dapat membantu mempermudah proses administrasi Linux. Utilitas-utilitas ini sangat membantu dalam mengolah informasi yang pada command line interface sebagian besar berbentuk teks.

# 1. Wild Card dan Regular Expression

Ketika berurusan dengan data teks seringkali kita diminta untuk dapat menemukan atau mengolah potongan data yang memiliki pola tertentu seperti nama user yang dimulai dengan huruf A atau NIM dari jurusan tertentu yang diawali dengan angka 3020, dan lain sebagainya. Untuk mempermudah pencarian dan pengolahan tersebut shell di Linux (dalam bab ini adalah bash) menyediakan sarana berupa pengenalan wild card dan regular expression.

## 2. Wild Card

Wild card adalah fitur bash berupa karakter-karakter khusus yang dapat digunakan untuk merepresentasikan pola-pola karakter tertentu. Karakter-karakter wild card yang tersedia beserta fungsinya dapat dilihat pada tabel di bawah ini.

Karakter	Fungsi
*(asterik)	Melambangkan tidak ada, satu, atau banyak karakter apa saja
?(tanda tanya)	Melambangkan tepat satu karakter apa saja
[<karakter>]	Melambangkan tepat satu karakter dari salah satu karakter yang tertulis di dalam tanda kurung
[<kar>-<kar>]	Melambangkan tepat salah satu karakter yang berada dalam rentang dua karakter yang tertulis di dalam tanda kurung, misal rentang a-z berarti melambangkan tepat satu karakter alfabet kecil di antara a hingga z termasuk a dan z
[!<karakter>]	Melambangkan tepat satu karakter dari semua karakter yang tidak tertulis di dalam tanda kurung setelah tanda seru (!)
[!<kar>-<kar>]	Melambangkan tepat salah satu karakter dari semua karakter yang berada diluar rentang dua karakter yang tertulis di dalam tanda kurung setelah tanda seru
{kata1,kata2}	Melambangkan salah satu kata dari kata-kata yang tertulis dalam tanda kurung

Tabel 1: wildcard pada bash shell dan fungsinya

Contoh:

pola	contoh kata yang cocok dengan pola
*.sh	setup.sh, config.sh, .sh, text.sh
3020??100	302007100, 302011100, 302001100
Ant[aiueo]	Anta,Anti,Antu,Ante,Anto
199[1-9]	1991, 1992, 1993, 1999
200[!4]	2001,2002,2003,2005,2006,200a, 200b,200x

R-[!A-C]S105	R-DS105, R- ES105, R- aS105, R- 2S105
{linux,unix}	linux atau unix atau keduanya
*int[AIUEO]	sintA, SintA, pintU, cintA, longintE, intO
30[1-3]0???	3010abc, 3020xyz, 3030lll

### 3. Regular Expression

Regular expression atau dapat disebut juga pattern (pola) adalah ekspresi yang mewakili sekelompok string (deretan karakter) atau sekelompok kata. Regular expression secara konsep memang mirip dengan wild card namun berbeda dalam notasi dan karakter-karakter khusus yang digunakan. Beberapa utilitas menyediakan opsi untuk memilih menggunakan aturan regular expression atau aturan wild card milik shell.

Regular expression sendiri memiliki beberapa standar aturan dan sistem operasi Unix dan Linux memilih menggunakan standar POSIX Basic Regular Expression (BRE) dan POSIX Extended Regular Expression (ERE).

Berikut ini adalah tabel aturan POSIX BRE

.	Cocok dengan tepat satu karakter apa saja. Pada ekspresi yang menggunakan karakter kurung ([]) titik tersebut cocok dengan karakter titik saja. Contohnya a.c cocok dengan "abc" tetapi [a.c] hanya cocok dengan "a", ".", "c".
[<karakter> ] [<kar>-<kar> ]	Kedua ekspresi kurung di samping memiliki fungsi yang sama dengan padanannya pada wildcard.
[^ ]	Melambangkan tepat satu karakter dari semua karakter yang tidak tertulis di dalam tanda kurung setelah tanda caret/circumflex "^".
^	Menandakan bahwa posisi ekspresi yang dicocokkan berada di depan baris atau deretan karakter (string)
\$	Menandakan bahwa posisi ekspresi yang dicocokkan berada di akhir baris atau deretan karakter (string) Contoh: [TC]elah\$ hanya cocok dengan "Telah" atau "Celah" yang berada di akhir string atau akhir baris
\( \)	Mendefinisikan sebuah subekspresi yang ditandai. String yang cocok dengan ekspresi dalam tanda kurung dapat digunakan di lain waktu melalui ekspresi \n yang dijelaskan setelah ini.
\n	Cocok dengan subekspresi ke-n yang ditandai di mana n bernilai 1 hingga 9.
*	Cocok dengan tidak ada, satu, atau perulangan dari karakter yang dituliskan sebelum karakter "*"

	<p>Contoh:</p> <p><math>ab^*c</math> cocok dengan "ac", "abc", "abbbc", dsb.</p> <p><math>[xyz]^*</math> cocok dengan "", "x", "y", "z", "zx", "zyx", "xyzzy", dsb.</p> <p><math>(ab)^*</math> cocok dengan "", "ab", "abab", "ababab", dsb.</p>
$\{m,n\}$	<p>Cocok dengan pola yang diletakkan sebelum ekspresi ini minimal sebanyak m kali dan tidak lebih dari n kali.</p> <p>Contoh:</p> <p><math>a\{3,5\}</math> hanya cocok dengan "aaa", "aaaa", dan "aaaaa".</p>

#### 4. Kelas-kelas Karakter pada POSIX

Untuk mempermudah mengingat dan menggunakan rentang karakter tertentu yang sering digunakan rentang-rentang karakter tersebut diberi simbol khusus yang disebut sebagai kelas karakter. Kelas-kelas karakter yang ada pada POSIX dapat dilihat pada tabel di bawah ini.

POSIX	ASCII	Keterangan
<code>[[:alnum:]]</code>	<code>[A-Za-z0-9]</code>	Karakter-karakter alfanumerik
<code>[[:word:]]</code>	<code>[A-Za-z0-9_]</code>	Karakter-karakter alfanumerik dan "_" (garis bawah)
	<code>[^\w]</code>	Karakter yang tidak terdapat dalam kata
<code>[[:alpha:]]</code>	<code>[A-Za-z]</code>	Karakter-karakter alfabet
<code>[[:blank:]]</code>	<code>[ \t]</code>	Spasi dan tabulasi
<code>[[:cntrl:]]</code>	<code>[\x00-\x1F\x7F]</code>	Karakter <i>control/ctrl</i>
<code>[[:digit:]]</code>	<code>[0-9]</code>	Karakter-karakter numerik
	<code>[^\d]</code>	Karakter-karakter non-numerik
<code>[[:graph:]]</code>	<code>[\x21-\x7E]</code>	Karakter-karakter yang dapat dilihat dilayar
<code>[[:lower:]]</code>	<code>[a-z]</code>	Karakter-karakter huruf kecil
<code>[[:print:]]</code>	<code>[\x20-\x7E]</code>	Karakter-karakter yang dapat dilihat dilayar, spasi, dan tabulasi
<code>[[:punct:]]</code>	<code>[!'"#\$%&amp;'()*+,-./:;&lt;=&gt;?@[\\\_`{ }~]</code>	Karakter-karakter <i>punctuation</i>
<code>[[:space:]]</code>	<code>[\t\r\n\v\f]</code>	Karakter-karakter pemberi ruang
	<code>[^\s]</code>	Karakter-karakter non-pemberi ruang
<code>[[:upper:]]</code>	<code>[A-Z]</code>	Karakter-karakter huruf capital
<code>[[:xdigit:]]</code>	<code>[A-Fa-f0-9]</code>	Karakter-karakter heksadesimal



Kelas-kelas karakter POSIX ini hanya dapat digunakan dalam ekspresi-ekspresi yang menggunakan tanda kurung, seperti `[[[:upper:]]ab]` cocok dengan sebuah huruf kapital diikuti "a" dan "b".

## 5. Utilitas-utilitas Bantu pada Command Line Interface Linux

Bekerja pada command line interface yang sebagian besar (hampir semua) data dan informasi ditampilkan dalam bentuk teks bukanlah pekerjaan yang mudah. Seringkali isi file atau keluaran dari sebuah perintah atau aplikasi lebih banyak dari kapasitas tampilan layar monitor atau menggunakan simbol-simbol tertentu yang membuatnya sulit untuk dibaca.

Untuk memudahkan mengelola keluaran-keluaran tersebut *command line interface* di Linux/Unix menyertakan utilitas-utilitas tambahan. Bab ini akan memfokuskan pada utilitas-utilitas tersebut.

### 1. Utilitas untuk Menyaring Tampilan Keluaran

Untuk contoh-contoh penggunaan perintah pada bagian selanjutnya digunakan yang filecoba berisi:

```
113000124 :Tafta Z :3.14
113000135 : Deri T:3.2
613010125 : Dian:3.4
```

#### More

more adalah perintah yang digunakan untuk melihat isi dari file yang jika ditampilkan akan membutuhkan beberapa halaman layar tampilan. more akan menampilkan isi file halaman per halaman. Utilitas more ini hanya bisa menggulung layar ke bawah dan tidak bisa kembali ke halaman sebelumnya.

Utilitas more dituliskan dengan cara sebagai berikut:

`$ more [-opsi] file1 [file2]`

Opsi-opsi tambahan yang biasa menyertai utilitas ini antara lain:

-num	Opsi ini digunakan untuk menentukan jumlah baris yang akan diganti setiap kali menggulung layar tampilan
-d	Menampilkan pesan "Press spacebar to continue and q to quit" dan pesan "press h for instruction" jika tombol yang ditekan bukan spacebar atau q
-l	Mencegah more dari memperlakukan karakter "form feed" sebagai karakter khusus
-p	Opsi ini menyebabkan tampilan sebelumnya dihapus dari layar dan tampilan halaman berikutnya ditampilkan
-s	Mampatkan kemunculan baris-baris kosong yang berurutan

	menjadi satu baris kosong saja
+/string	Digunakan untuk menentukan kata tertentu yang akan dicari
+num	Mulai menampilkan dari baris tertentu saja

Contoh-contoh:

```
$ more /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
rpm:x:37:37:/:/var/lib/rpm:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
avahi:x:70:70:Avahi daemon:/:/sbin/nologin
mailnull:x:47:47:/:/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51:/:/var/spool/mqueue:/sbin/nologin
distcache:x:94:94:Distcache:/:/sbin/nologin
ntp:x:38:38:/:etc/ntp:/sbin/nologin
--More--(52%)
```

```
$ more -d /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
rpm:x:37:37:/:/var/lib/rpm:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
avahi:x:70:70:Avahi daemon:/:/sbin/nologin
mailnull:x:47:47:/:/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51:/:/var/spool/mqueue:/sbin/nologin
distcache:x:94:94:Distcache:/:/sbin/nologin
ntp:x:38:38:/:etc/ntp:/sbin/nologin
--More--(52%)IPress space to continue, 'q' to quit.]
```

## Less

Less memiliki fungsi yang sama dengan more tetapi dengan kelebihan less dapat melihat kembali halaman-halaman sebelumnya. Less dioperasikan mirip dengan editor teks vi. Tombol-tombol dan perintah-perintah pada vi dapat digunakan ketika kita sedang berada dalam lingkungan tampilan less. Meskipun mirip tetapi less tetaplah bukan sebuah editor teks yang dapat menyimpan hasil dalam file. Utilitas less dituliskan dengan cara sebagai berikut:

```
$ less [-opsi] file1 [file2]
```

Opsi-opsi tambahan yang biasa mengikuti utilitas ini antara lain:

-b <angka>	Menentukan jumlah memori buffer yang digunakan untuk sebuah file yang sedang dilihat. Secara default nilainya adalah 64 KB
-h <angka>	Menentukan jumlah baris maksimum yang bisa digulung kembali (melihat layar tampilan sebelumnya). Jika terminal tidak mendukung fungsi menggulung tampilan maka otomatis h bernilai 0.

## Head

Head adalah utilitas yang digunakan untuk melihat bagian awal dari isi file atau tampilan keluaran perintah/utilitas lain. Secara *default* perintah ini akan

menampilkan sepuluh baris pertama isi file atau tampilan keluaran perintah/utilitas lain.

Utilitas head dituliskan dengan cara sebagai berikut:

```
$ head [-opsi] file1 [file2]
```

Opsi-opsi tambahan yang biasa mengikuti utilitas ini antara lain:

-c <angka>	Menampilkan sebanyak jumlah byte tertentu dihitung mulai dari awal sebuah file atau tampilan. Jika diikuti nilai negatif maka yang akan ditampilkan adalah jumlah byte dihitung dari akhir file atau tampilan.
-n	Tampilkan sebanyak n baris pertama bukan 10 baris pertama yang menjadi default.
-q	Quiet berarti tidak menampilkan nama file yang dijadikan masukan.
-v	Verbose berarti selalu menampilkan nama file yang menjadi masukan.

Contoh-contoh:

```
$ head -5 /etc/passwd
```

```
[root@localhost ~]# head -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
$ head /etc/passwd
```

```
[root@localhost ~]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
[root@localhost ~]#
```

## Tail

Kebalikan dari head, tail digunakan untuk melihat bagian akhir dari isi file atau tampilan perintah/utilitas lain. Secara default perintah ini akan menampilkan sepuluh baris terakhir isi file atau tampilan keluaran perintah/utilitas lain.

Utilitas tail dituliskan dengan cara sebagai berikut:

```
$ tail [-opsi] file1 [file2]
```

Opsi-opsi tambahan yang biasa mengikuti utilitas ini antara lain:

-c <angka>	Menampilkan sebanyak jumlah byte tertentu dihitung mulai
------------	--

	dari akhir sebuah file atau tampilan. Jika diikuti nilai negatif maka yang akan ditampilkan adalah jumlah byte dihitung dari awal file atau tampilan.
-n	Tampilkan sebanyak n baris terakhir bukan 10 baris terakhir yang menjadi default.
-q	Quiet berarti tidak menampilkan nama file yang dijadikan masukan.
-v	Verbose berarti selalu menampilkan nama file yang menjadi masukan.

Contoh-contoh:

```
$ tail -5 /etc/passwd
```

```
[root@localhost ~]# tail -5 /etc/passwd
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
hsqldb:x:96:96::/var/lib/hsqldb:/sbin/nologin
gdm:x:42:42::/var/gdm:/sbin/nologin
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
Tafta:x:500:500:Tafta Zani:/home/Tafta:/bin/bash
[root@localhost ~]#
```

```
$ tail /etc/passwd
```

```
[root@localhost ~]# tail /etc/passwd
dovecot:x:97:97:dovecot:/usr/libexec/dovecot:/sbin/nologin
webalizer:x:67:67:Webalizer:/var/www/usage:/sbin/nologin
squid:x:23:23::/var/spool/squid:/sbin/nologin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
hsqldb:x:96:96::/var/lib/hsqldb:/sbin/nologin
gdm:x:42:42::/var/gdm:/sbin/nologin
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
Tafta:x:500:500:Tafta Zani:/home/Tafta:/bin/bash
```

## Grep

Grep adalah salah satu perintah favorit para pengguna command line interface di Linux. Dengan perintah ini kita dapat mencari dan menyaring data atau informasi tertentu dan menampilkannya dari sekumpulan data atau informasi dalam file atau tampilan keluaran perintah/utilitas lain.

Grep mencari dan mencocokkan setiap baris teks yang menjadi masukan (dapat berupa isi file atau hasil perintah lain) dengan suatu pola yang ditentukan oleh pengguna. Pola yang dimasukkan mengikuti aturan tertentu yang disebut sebagai *regular expression* yang telah dipelajari di awal bab ini.

Utilitas grep dituliskan dengan cara sebagai berikut:

```
$ grep [-opsi] file1 [file2]
```

Contoh-contoh:

Perintah untuk menampilkan hanya baris yang dimulai dengan karakter 6.

```
$ grep ^61 < filecoba
```

```
613010125 : Dian:3.4
```

Perintah untuk menampilkan hanya baris yang diakhiri dengan karakter 4.

```
$ grep 4$ < filecoba
```

```
113000124 :Tafta Z :3.14
```

```
613010125 : Dian:3.4
```

## 6. Utilitas untuk Penanggalan

### Date

Date adalah utilitas yang sangat berguna untuk menampilkan dan menentukan tanggal dan waktu dari komputer yang sedang digunakan. Date memberikan keleluasaan bagi pengguna untuk menentukan format penulisan tanggal dan waktu yang akan ditampilkan melalui simbol-simbol format waktu yang dapat disesuaikan urutan penulisannya. Date juga dapat menampilkan tanggal dan waktu yang sudah lampau dan masa depan (lihat bantuan perintah date untuk daftar lengkap simbol-simbol yang dapat digunakan)

Utilitas date dituliskan dengan cara sebagai berikut:

```
$ date [-opsi] string
```

Opsi-opsi tambahan yang biasa mengikuti utilitas ini antara lain:

-d <kata>	Menampilkan waktu sesuai kata setelah opsi -d seperti "yesterday", "tomorrow", "next week", dll.
-u	Menampilkan atau menetapkan waktu berdasarkan utc (Universal Coordinated Time).
-s <string>	Menentukan waktu sistem berdasarkan string setelah opsi -s.

Contoh-contoh:

```
$ date -d tomorrow
```

```
Sun Jan 11 15:14:31 WIT 2009
```

```
$ date +%D
```

```
01/10/09
```

```
$ date "+Bandung, %d %B %Y"
```

```
Bandung, 10 January 2009
```

## 7. Utilitas untuk Memformat Tampilan Keluaran

### Sed

Sed atau stream editor adalah utilitas bantu yang dapat digunakan untuk menyunting sebuah stream. Sebuah stream adalah aliran data yang sedang

diproses. Data tersebut dapat berasal dari sebuah file atau keluaran dari perintah/utilitas lain. Ilustrasi di bawah ini menjelaskan tentang sed dan stream.

Pada kenyataannya sed dapat langsung membaca isi file dan tidak perlu mendapat masukan dari perintah/utilitas lainnya seperti diilustrasikan pada gambar di atas.

Beberapa fasilitas penyuntingan yang disediakan oleh sed antara lain:

1. Mengganti kata atau pola tertentu
2. Menghapus baris dengan kata atau pola tertentu

Utilitas sed dituliskan dengan cara sebagai berikut:

```
$ sed [-opsi] < file
```

Contoh-contoh:

### *Mengganti karakter atau pola karakter tertentu*

```
$ sed s/T/t/ < filecoba
113000124 :tafta Z :3.14
113000135 : Deri T:3.2
613010125 : Dian:3.4
```

Perintah di atas menyebabkan kemunculan pertama huruf T pada isi file filecoba akan diganti dengan huruf t. Perlu diperhatikan bahwa pada perintah di atas hanya kemunculan pertama dari huruf T saja yang diganti. Untuk mengganti seluruh kemunculan huruf T gunakanlah perintah di bawah ini.

```
$ sed s/T/t/g < filecoba
113000124 :tafta Z :3.14
113000135 : Deri t:3.2
613010125 : Dian:3.4
```

Untuk mengganti pola kata tertentu gunakanlah regular expression yang sudah dipelajari sebelumnya. Perintah berikut akan mengganti semua kemunculan string dengan pola dimulai dengan huruf D dan terdiri dari 4 karakter menjadi karakter x.

```
$ sed s/D.../x/g < filecoba
113000124 :Tafta Z :3.14
113000135 : x T:3.2
613010125 : x:3.4
```

### *Menghapus baris yang mengandung karakter atau pola karakter tertentu*

Untuk menghapus baris yang mengandung karakter atau pola karakter tertentu gunakan opsi d pada perintah sed. Perintah berikut ini akan menghapus semua baris yang mengandung huruf a pada filecoba.

```
$ sed /a/d < filecoba
```



113000135 : Deri T:3.2

Perintah sed ini memiliki banyak sekali fitur lain. Untuk lebih lengkapnya dapat dilihat di internet seperti halaman <http://www.grymoire.com/Unix/Sed.html> yang berisi panduan menggunakan sed yang cukup lengkap dan disertai contoh yang mudah dimengerti.

## Tr

Tr adalah utilitas yang digunakan untuk mengganti, menghilangkan, atau menambahkan karakter tertentu pada suatu tampilan berupa isi file atau keluaran dari perintah/utilitas lainnya.

Utilitas tr dituliskan dengan cara sebagai berikut

**\$ tr [-opsi] SET1 [SET2]**

SET1 adalah karakter atau kumpulan karakter yang akan dikenakan operasi tertentu sedangkan SET2 adalah karakter atau kumpulan karakter yang akan dioperasikan pada masukan.

Opsi-opsi tambahan yang biasa mengikuti utilitas ini antara lain:

-s <karakter>	Menghilangkan pengulangan karakter tertentu yang muncul pada masukan
-d <karakter>	Menghapus karakter tertentu dari masukan
-c <karakter>	Menjadikan tr beroperasi pada karakter yang tidak disebutkan pada SET1

### Contoh-contoh:

```
$ tr -s " " < filecoba
113000124 :Tafta Z :3.14
113000135 : Deri T:3.2
613010125 : Dian:3.4
```

```
$ tr -d " " < filecoba
113000124:TaftaZ:3.14
113000135:DeriT:3.2
613010125:Dian:3.4
```

## Cut

Cut seperti terjemahannya yang berarti memotong, berguna untuk memotong/mengilangkan karakter atau kolom pada setiap baris tampilan berdasarkan posisi karakter atau kolom yang dipisahkan dengan karakter tertentu.

Utilitas join dituliskan dengan cara sebagai berikut:

**\$ cut [-opsi] filel**

Opsi-opsi tambahan yang biasa mengikuti utilitas ini antara lain:

-b <angka>	Tampilkan karakter pada byte tertentu
-c <angka>	Tampilkan karakter pada posisi tertentu
-d <karakter>	Menentukan karakter yang dianggap sebagai pembatas antar kolom dari masukan.
-f <kolom>	Menentukan kolom yang akan ditampilkan
-s	Hilangkan baris-baris yang tidak mengandung karakter pembatas antar kolom

Contoh-contoh:

```
$ cut -d ":" -f 2 filecoba
```

```
Tafta Z
```

```
Deri T
```

```
Dian
```

```
$ cut -b 3 filecoba
```

```
3
```

```
3
```

```
3
```

Perlu diperhatikan bahwa **satu karakter ASCII setara dengan satu byte**.

## Join

Join berguna untuk menampilkan hasil penggabungan dua buah file. Penggabungan dua file tersebut dilakukan berdasarkan kolom-kolom tertentu yang identik yang pada masing-masing file. Isi kolom identik pada masing-masing file harus berada dalam kondisi terurut.

Utilitas join dituliskan dengan cara sebagai berikut:

```
$ join [-opsi] file1 file2
```

Opsi-opi tambahan yang biasa mengikuti utilitas ini antara lain:

-j <kolom>	Ekivalen dengan -l kolom -2 kolom.
-t <karakter>	Menentukan pembatas antar kolom yang digunakan.
-o <file>.<kolom>	Menentukan format penampilan kolom-kolom dari kedua file yang digabung
-l <kolom>	Menentukan kolom rujukan penggabungan pada file pertama.
-2 <kolom>	Menentukan kolom rujukan penggabungan pada file ke-dua.
-e <karakter>	Menentukan pemisah antar kolom yang digunakan pada masukan dan tampilan keluaran.

Contoh-contoh:

Kedua file bernama file1 dan file2 berikut ini akan digunakan pada contoh-contoh utilitas join.

Isi file1:

```
Tafta:Jaringan Komputer:tetap
```

```
Prajna:Jaringan Komputer:luar biasa
```

```
Heru:Statistika:luar biasa
```

```
Dewi:Perancangan Basis Data:tetap
```

```
Henry:Sistem Komputer:tetap
```

Isi file2:

```
Tafta:08156425141:Bandung
```

```
Prajna:022915462:Bandung
```

```
Heru:02276124575:Cirebon
```

```
Dewi:0853214054::Cirebon
```

Henry:081355457845:Banyuwangi

Kolom pertama pada kedua file identik dan akan menjadi acuan perintah join dalam melakukan penggabungan.

```
$ join -t ":" -1 1 -2 1 -o "1.1 1.2 2.2 " file1 file2
```

Tafta:Jaringan Komputer: 08156425141

Prajna:Jaringan Komputer: 022915462

Heru:Statistika: 02276124575

Dewi:Perancangan Basis Data: 0853214054

Henry:Sistem Komputer: 081355457845

Perintah di atas melakukan penggabungan file1 dan file2 dengan acuan kolom pertama pada kedua file dan dengan pemisah antar kolom pada masing-masing file adalah karakter ":" kemudian hasilnya ditampilkan dengan urutan kolom pertama dari file1 kemudian kolom ke-dua dari file1 kemudian kolom ke-dua dari file2.

# BAB V

## Manajemen User, Group, dan Hak Akses di Linux

---

### Tujuan:

4. Siswa memahami konsep user dan group di Linux.
5. Siswa dapat menggunakan utilitas-utilitas untuk membantu proses administrasi user dan group.
6. Siswa memahami konsep kepemilikan resource pada Linux.
7. Siswa dapat menggunakan utilitas-utilitas untuk manajemen hak akses terhadap resource di Linux.

### Deskripsi singkat:

Pada praktikum ini siswa akan diperkenalkan konsep user dan group disertai berbagai utilitas tambahan yang biasa digunakan pada command line interface untuk manajemen user. Selain itu siswa akan belajar mengenai hak akses dan kepemilikan pada sumber daya di Linux dan cara untuk manajemen hak akses dan kepemilikan tersebut menggunakan utilitas-utilitas yang disediakan pada command line interface. ini sangat membantu dalam mengolah informasi yang pada command line interface sebagian besar berbentuk teks.

## 8. Konsep user dan group

Pada komputer yang dapat digunakan oleh banyak orang pada satu waktu (multiuser) sistem operasi harus dapat membedakan setiap pengguna yang menggunakan komputer tersebut. Pada Sistem Operasi Linux setiap pengguna diwakilkan melalui sebuah entitas yang disebut sebagai **user**. User memiliki beberapa atribut yang membedakannya dengan user lain pada sistem operasi yang sama atau bahkan sekumpulan computer dalam jaringan seperti nama user, no identitas, lokasi direktori penyimpanan data-data user tersebut, kata kunci, hak akses, dsb.

Dikarenakan tidak semua user berbeda sama sekali dari user lainnya maka pada kasus tertentu sekelompok user yang memiliki atribut yang sama dikelompokkan dalam suatu entitas yang disebut **group**. Contoh kasusnya adalah sebuah komputer yang digunakan untuk mengajar oleh banyak dosen. Pada system operasi komputer tersebut setiap user yang merupakan dosen dikelompokkan dalam sebuah group bernama dosen. Pengelompokan ini akan sangat membantu dalam manajemen para user yang merupakan dosen. Misalkan para dosen akan diberikan hak khusus untuk menggunakan fasilitas jaringan dan mengakses daftar nilai maka hak akses tersebut dapat diberikan melalui pemberian hak akses kepada group dosen di mana mereka bergabung.

Seperti halnya benda konkrit, informasi dan sumber daya yang bersifat abstrak pada komputer juga memiliki pemilik. Sistem operasi memberikan identitas kepada pemilik sumber daya dan informasi tersebut agar dapat mengamankan informasi dan sumber daya tersebut dari akses oleh pihak yang tidak berwenang. Pada Sistem Operasi Linux identitas itu dikenal sebagai user.

### File /etc/passwd dan /etc/group

Pada Sistem Operasi Linux sebagian informasi mengenai user dan group disimpan dalam file /etc/passwd dan /etc/group. Berikut ini adalah contoh dari isi file /etc/passwd.



```
tafta@localhost/etc
File Edit View Terminal Tabs Help
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
rpm:x:37:37:/:/var/lib/rpm:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
netdump:x:34:34:Network Crash Dump user:/var/crash:/bin/bash
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
```

File ini berbentuk plaintext dan setiap baris mewakili setiap user yang ada. Informasi tentang user yang tersimpan dalam setiap baris dipisahkan menggunakan karakter “:”. Informasi yang tersimpan antara lain:

tafta:x:500:500:Tafta Zani:/home/tafta:/bin/bash

1. Nama user,
2. Password (agar lebih aman kebanyakan distro Linux meletakkan password yang sudah dienkripsi ke dalam file /etc/shadow),
3. Nomor User ID,
4. Nomor Group ID,
5. Komentar atau keterangan mengenai user
6. Lokasi *home directory*,
7. Shell yang digunakan ketika user telah login.

Informasi mengenai group dan user yang menjadi anggotanya tersimpan dalam file /etc/group.



```
tafta@localhost:/etc
File Edit View Terminal Tabs Help
users:x:100:
dbus:x:81:
floppy:x:19:
vcsa:x:69:
nscd:x:28:
rpm:x:37:
haldaemon:x:68:
utmp:x:22:
netdump:x:34:
slocate:x:21:
sshd:x:74:
rpc:x:32:
rpcuser:x:29:
nfsnobody:x:65534:
mailnull:x:47:
smmsp:x:51:
pcap:x:77:
apache:x:48:
squid:x:23:
webalizer:x:67:
xfs:x:43:
ntp:x:38:
gdm:x:42:
tafta:x:500:
```

File ini seperti halnya kebanyakan file konfigurasi di Linux adalah plaintext dan seperti halnya `/etc/passwd` setiap baris berisi informasi mengenai satu group yang ada. Informasi disimpan dalam kolom-kolom yang dipisahkan dengan karakter “:” dan berisi antara lain:

`bin:x:1:root,bin,daemon`

1. Nama group,
2. Password (jika kosong maka password tidak dibutuhkan)
3. Group ID,
4. User yang menjadi anggota group tersebut.

## 9. Manajemen User

### a.....Menambah User

Menambah user dapat dilakukan dengan menggunakan perintah **useradd** pada *command line interface*. Perintah sederhananya adalah sebagai berikut:

```
# useradd [opsi] nama_user
```



Dengan perintah tersebut system operasi akan membuat sebuah user baru dengan nama `_user` sebagai nama user baru tersebut.

Dengan menambahkan opsi pada perintah `useradd` atribut-atribut tertentu dari sebuah user dapat ditentukan ketika user tersebut baru dibuat. Berikut ini beberapa atribut yang dapat ditentukan dan opsi yang digunakan dalam perintah `useradd`:

-c	Menambahkan keterangan/komentar mengenai user yang akan dibuat. Keterangan akan disimpan pada file <code>/etc/passwd</code> pada baris user yang bersangkutan
-d	Menentukan direktori yang akan dijadikan home directory untuk user yang akan dibuat. Secara default jika tidak diubah maka user mendapatkan home directory di <code>/home/nama_user</code> . Contoh: <code>/home/tafta</code> .
-e	Menentukan expire date (batas kadaluwarsa) yaitu tanggal ketika user yang akan dibuat menjadi tidak aktif dan tidak bisa digunakan. Opsi ini bisa digunakan untuk membuat user yang bersifat sementara seperti tamu atau pengunjung. Format waktu yang digunakan adalah <code>YYY-MM-DD</code> . Contohnya: <code>2009-06-07</code> berarti tanggal 7 Juni 2009
-f	Menentukan jumlah hari sebelum user menjadi tidak aktif secara permanen setelah password kadaluwarsa. Jika diisi nilai 0 maka user akan langsung menjadi tidak aktif dan jika diisi nilai -1 maka fitur kadaluwarsa ini tidak diaktifkan.
-g	Menentukan group id user yang akan dibuat. Secara default adalah 1 atau berapapun yang ditentukan dalam file <code>/etc/default/useradd</code> .
-G	Menentukan daftar group di mana user ini akan bergabung. Secara default seorang user akan dibuatkan sebuah group dengan nama group yang sama dengan nama usernya.
-m	Dengan opsi ini home directory dari user akan dibuat dengan menyalin isi direktori tertentu ke dalam home directory user jika opsi <code>-k</code> digunakan. Jika opsi <code>-k</code> tidak digunakan maka secara default isi direktori <code>/etc/skel</code> lah yang akan disalin ke dalam home directory user. Pada beberapa distro Linux perintah <code>useradd</code> akan secara otomatis membuatkan home directory untuk user tanpa perlu mengikutkan opsi <code>-m</code> .
-k	Opsi ini digunakan hanya jika opsi <code>-m</code> juga diikuti. Opsi ini berguna untuk menentukan direktori tertentu yang akan disalin ke dalam home directory user.
-s	Menentukan shell tertentu untuk menjadi shell yang diberikan

	kepada user ketika user login. Contohnya: /bin/bash, /bin/sh, atau /bin/csh.
-p	Menentukan password untuk user. Opsi ini digunakan bersama utilitas <b>crypt</b> . Jika tidak diberikan nilai tertentu maka user akan dinon-aktifkan
-o	Mengijinkan membuat user dengan UID yang sama dengan user lain.
-u	Menentukan UID dari user. Nilai yang diberikan harus unik (tidak boleh sama dengan user lain kecuali opsi <code>-o</code> diikutkan). Nilai 0 – 99 biasanya digunakan oleh sistem dan tidak untuk user biasa.

## 10. Contoh-contoh

- Membuat user dengan nama student  
# `useradd student`
- Membuat user bernama student dengan home directory /home/students/student  
# `useradd -m /home/students/student student`
- Membuat user student dengan user id yang sudah digunakan oleh user yang lain (misalkan 10)  
# `useradd -o -uid 10 student`
- Membuat user student dengan batas waktu kadaluwarsa password tanggal 8 Februari 2009  
# `useradd -e 2009-02-08 student`

## b.....Memodifikasi User

Atribut dari user yang telah dibuat dapat diubah/dimodifikasi sesuai keperluan. Perintah yang digunakan adalah `usermod`. Perintah `usermod` dapat digunakan dengan cara yang mirip dengan perintah `useradd`.

```
# usermod [opsi] namauser
```

Kebanyakan opsi yang bekerja pada perintah `useradd` dapat diaplikasikan pada perintah `usermod` namun ada pula yang bekerja hanya pada perintah `usermod` seperti:

-l	Mengganti nama user lama dengan nama user baru yang diketikkan setelah opsi <code>-l</code> tersebut.
----	---

-L	Mengunci password user sehingga tidak dapat digunakan untuk login. Hal ini dilakukan dengan menambahkan karakter tanda seru (!) di depan hasil enkripsi password user pada file /etc/passwd atau /etc/shadow.
-U	Membuka kunci password user sehingga kembali dapat digunakan untuk login. Hal ini dilakukan dengan menghilangkan karakter tanda seru (!) dari depan hasil enkripsi password user pada file /etc/passwd atau /etc/shadow.
-d	Mengubah home directory user ke lokasi lain. Jika opsi -m diikuti maka isi home directory lama akan dipindahkan ke lokasi yang baru pula.
-a	Jika digunakan dengan opsi -G maka akan keanggotaan user pada group lama tidak akan dihilangkan.

## 11. Contoh-contoh

- Mengubah nama user student dengan nama user mahasiswa  
# usermod -l mahasiswa student
- Mengubah home directory user bernama student menjadi /home/students/mhs  
# usermod -d /home/students/mhs student
- Mengubah uid user student dengan user id yang lain  
# usermod -u 10 student
- Membuat user student dengan batas waktu kadaluwarsa password tanggal 8 Februari 2009  
# usermod student -e 2009-02-08

### c. Menghapus User

User yang tidak diperlukan lagi sebaiknya segera dihapus dengan alasan keamanan. Perintah userdel dapat digunakan untuk menghapus sebuah user.

```
# userdel namauser
```

Selain itu ada -r yang dapat disertakan pada perintah user del. Opsi ini akan memerintahkan userdel untuk menghapus home directory user.

```
# userdel -r namauser
```

## **12.....Memanajemen Group**

Seperti halnya user, group juga perlu dimanajemen. Terkadang group baru perlu dibuat atau ada user baru yang harus ditambahkan dalam sebuah group atau ada group yang harus dihapus karena sudah tidak digunakan lagi.

### **d.....Menambah Group**

Menambah user dapat dilakukan dengan menggunakan perintah `useradd` pada command line interface. Perintah sederhananya adalah sebagai berikut:

```
# groupadd nama_group
```

Dengan perintah tersebut sistem operasi akan membuat sebuah group baru dengan `nama_group` sebagai nama group baru tersebut.

Berikut ini beberapa atribut yang dapat ditentukan dan opsi yang digunakan dalam perintah `groupadd`:

-g	Menentukan nomor gid untuk group yang akan dibuat. Group ID yang diinginkan dituliskan setelah opsi -g.
-r	Menentukan bahwa group yang dibuat adalah group untuk sistem yang berarti akan memiliki gid lebih kecil dari 499.
-f	Opsi ini akan membuat groupadd keluar dengan kondisi error ketika nama group atau group id yang diinginkan sudah ada. Opsi ini berlaku untuk opsi -g dan -o.
-o	Mengizinkan membuat group baru dengan gid yang sudah digunakan untuk group lain.

### 13. Contoh-contoh

- Membuat group baru dengan nama students  
# groupadd students
- Membuat group students dengan gid yang sudah digunakan oleh group yang lain (misalkan 501)  
# groupadd -g 501 -o student
- Membuat group baru dengan nama students yang merupakan group sistem (untuk kepentingan administrasi sistem atau layanan tertentu).  
# groupadd -r students

#### e. Memodifikasi Group

Atribut dari group yang telah dibuat dapat diubah/dimodifikasi sesuai keperluan. Perintah yang digunakan adalah usermod. Perintah usermod dapat digunakan dengan cara yang mirip dengan perintah useradd.

```
# groupmod [opsi] namagroup
```

Berikut ini adalah opsi-opsi yang dapat diberikan untuk groupmod, yaitu:

-g	Mengubah nomor gid untuk group yang telah ada. Group ID yang diinginkan dituliskan setelah opsi -g.
-n	Mengubah nama group untuk group yang telah ada. Nama group yang diinginkan dituliskan setelah opsi -n.

#### f. Menghapus Group

Group yang tidak diperlukan lagi sebaiknya segera dihapus dengan alasan keamanan. Perintah groupdel dapat digunakan untuk menghapus sebuah group.

```
# groupdel namagroup
```

## 14.....Memajemen Password

Untuk alasan keamanan terkadang password perlu ditetapkan defaultnya kemudian diganti secara berkala atau dinonaktifkan. Untuk memajemen password user dapat menggunakan perintah passwd.

Perintah passwd memiliki pola sebagai berikut:

# passwd [opsi] namauser

-l	Mengunci sebuah user. Opsi ini hanya tersedia untuk root.
-u	Membuka user yang dikunci dengan opsi -l. Opsi ini hanya tersedia untuk root.
-d	Menyebabkan password untuk user dikosongkan. Opsi ini hanya tersedia untuk root.
-n	Menentukan jumlah hari minimum sebelum password kadaluwarsa dan diberikan peringatan oleh sistem operasi untuk mengganti password. Opsi ini hanya tersedia untuk root.
-x	Menentukan jumlah hari maksimum sebelum password kadaluwarsa dan sistem operasi akan mengunci user. Opsi ini hanya tersedia untuk root.
-w	Menentukan jumlah hari user akan mulai diberi peringatan bahwa password akan kadaluwarsa sebelum password kadaluwarsa. Opsi ini hanya tersedia untuk root.
-i	Menentukan jumlah hari sebelum sebuah user yang passwordnya kadaluwarsa akan dinon-aktifkan/dikunci. Opsi ini hanya tersedia untuk root.
-S	Memberikan informasi singkat mengenai status password sebuah user. Opsi ini hanya tersedia untuk root.

## 15. Contoh-contoh

- Mengubah password user yang sedang login  
# passwd
- Mengubah password user lain (hanya untuk root)  
# passwd namauser
- Mengunci dan membuka user  
# passwd -l namauser  
# passwd -u namauser
- Menentukan bahwa password akan kadaluwarsa dalam 3 hari  
# passwd -n 3 namauser

## 16. Konsep Kepemilikan dan Hak Akses pada Sumber Daya di Linux

Pada setiap file dan direktori di Sistem Operasi Linux terdapat atribut-atribut yang menentukan kepemilikan dan izin akses terhadap file dan direktori tersebut. Untuk melihat atribut-atribut tersebut user dapat menggunakan perintah:

```
# ls -l
```

pada salah satu direktori di filesystem Linux.

```
-rw----- 1 tafta tafta 66 Feb 2 10:03 .Xauthority
```

Keterangan:

`-rw-----` = Memberikan keterangan jenis file dan hak akses.

`tafta tafta` = User dan group pemilik file.

Sepuluh karakter pertama yang berisi keterangan jenis file dan hak akses terbagi-bagi menjadi 4 bagian yaitu:

1. Karakter pertama menunjukkan jenis file.
  - a. `-` berarti file tersebut adalah file biasa
  - b. `d` berarti file adalah direktori
  - c. `p` berarti file adalah file berjensi pipe
  - d. `c` berarti file adalah file device berbasis karakter (keyboard, modem, printer)
  - e. `b` berarti file adalah file device berbasis blok (harrdisk, USB storage, CDROM, dsb)
2. Tiga karakter (karakter ke-2 s.d ke-4) berikutnya adalah izin akses untuk pemilik file yang nama usernya tertera pada atribut file..
3. Tiga karakter (karakter ke-2 s.d ke-4) berikutnya adalah izin akses untuk group pemilik file yang nama groupnya tertera pada atribut file.
4. Tiga karakter (karakter ke-2 s.d ke-4) berikutnya adalah izin akses untuk user atau group lain dalam sistem operasi.

Tiga karakter izin akses tersebut berisi `r` (read) untuk izin baca, `w` (write) untuk izin tulis, dan `x` (untuk izin eksekusi jika file tersebut adalah aplikasi).

Misalkan: `-rwxrw----`

Berarti file tersebut adalah file biasa dengan izin akses untuk pemilik adalah baca, tulis, dan eksekusi sedangkan untuk group adalah baca dan tulis saja, dan untuk user atau group lainnya (dikenal sebagai other) tidak memiliki

izin akses apa-apa terhadap file tersebut. Jika file tersebut ternyata adalah direktori maka izin akses eksekusi (x) berarti direktori tersebut boleh dijadikan direktori aktif (user yang mendapatkan izin x boleh pindah ke dalamnya).

## **g.....Mengubah Kepemilikan**

Untuk mengubah user pemilik sebuah file user dapat menggunakan utilitas chown.

```
# chown namauser file
```

Untuk mengubah group pemilik sebuah file user dapat menggunakan utilitas chown.

```
# chgrp namagroup file
```

## **h.....Mengubah Izin Akses**

Untuk mengubah izin akses terhadap suatu file user dapat menggunakan perintah chmod sebagai berikut:

```
# chmod <izin akses> file
```

Pada perintah chmod user pemilik dilambangkan dengan karakter u, group pemilik dilambangkan dengan karakter g, dan other dilambangkan dengan o. Izin akses baca dilambangkan dengan r, izin mengubah/menulis dengan w, dan izin eksekusi dengan x. Untuk menambahkan izin tertentu gunakanlah karakter + dan untuk menghilangkan izin tertentu gunakanlah karakter -.

## **17. Contoh-contoh**

- Menambahkan izin akses read untuk group  

```
# chmod g+r file
```
- Mengubah izin akses untuk group sehingga hanya bisa membaca saja  

```
# chmod g=r file
```
- Mengubah izin akses untuk user pemilik sehingga bisa membaca, mengubah, dan mengeksekusi  

```
# chmod u+rw file
```

atau

```
# chmod u=rx file
```

Selain itu user dapat mengganti seluruh hak akses dengan notasi angka dengan aturan sebagai berikut:



1. Izin akses r diberi nilai 4, izin akses w bernilai 2, dan izin akses x bernilai 1.
2. Jika izin akses akan untuk user, group, atau other pada awalnya bernilai 0. Setiap menambahkan izin akses tertentu tambahkan dengan nilai izin akses tersebut dan setiap menghilangkan izin akses kurangi dengan nilai izin akses yang dihilangkan.

Contoh:

-rw-rw-r-- berarti

Jenis file		Izin akses user pemilik			Izin akses group pemilik			Izin akses other	
-	r	w	x	r	w	-	r	-	-
File biasa	4	2	1	4	2	0	4	0	0
		7			6			4	

Untuk mengubah izin akses menjadi seperti di atas maka gunakan perintah sebagai berikut:

```
# chmod 764 namafile
```

# Bab VI

## Shell Programming I

---

### **Tujuan:**

8. Siswa mengetahui fungsi dan cara menggunakan karakter-karakter khusus dalam shell Linux.
9. Siswa memahami konsep dan dapat menggunakan variabel dalam shell Linux.
10. Siswa memahami konsep dan dapat menggunakan standard input, output, dan error melalui piping dan redirection.

### **Deskripsi singkat:**

Pada praktikum ini siswa akan diperkenalkan karakter special dan cara menggunakannya dalam lingkungan shell Linux. Siswa juga diajak mengenal dan menggunakan variabel untuk mendukung pemrograman shell lanjutan. Berikutnya siswa akan diperkenalkan konsep standard input, output, dan error yang dimiliki oleh aplikasi-aplikasi di Sistem Operasi Linux dan memanfaatkan fasilitas tersebut menggunakan redirection dan piping.

## 18. Karakter-karakter Khusus dalam Shell, Arti, dan Penggunaannya

Pada saat bekerja dengan shell di Linux tidak semua karakter dapat digunakan secara bebas. Shell di Linux termasuk Bash memanfaatkan beberapa karakter khusus untuk fungsi-fungsi tertentu contohnya adalah karakter-karakter yang digunakan pada wildcard dan regular expression yang sudah diterangkan pada bab sebelumnya. Berikut ini adalah tabel ringkasan karakter-karakter tersebut beserta keterangan singkatnya.

Karakter	Nama	Keterangan Kegunaan
" "	Double quote (kutip ganda)	Berguna untuk menghilangkan fungsi khusus pada beberapa karakter khusus sehingga diartikan sebagai karakter biasa. Karakter-karakter yang bisa dinetralkan oleh karakter kutip ganda antara lain: ~ @ # % ^ & * ( ) _ +   { } [ ] : ; < > ? , . / = termasuk juga karakter spasi
' '	Single quote (kutip tunggal)	Sama seperti kutip ganda tetapi dapat menetralkan lebih banyak karakter termasuk karakter kutip ganda. Karakter-karakter yang dapat dinetralkan antara lain: \ ! " \$ %
` ``	Back quote (kutip terbalik)	Menyebabkan apapun yang ditulis di antara dua karakter ini akan dianggap sebagai perintah yang akan dieksekusi
\	Backslash	Sama seperti kutip ganda dan tunggal tetapi dapat mematikan semua karakter khusus yang ada. Dapat juga digunakan untuk

		menandakan bahwa baris perintah berlanjut di baris berikutnya.
#	Sharp (pagar)	Penanda baris komentar
\$	Dolar	Penanda bahwa kata setelah tanda ini adalah nama variabel
()	Bracket (tanda kurung)	Berguna untuk mengelompokkan beberapa perintah sehingga dianggap sebagai suatu eksekusi tunggal, membuat suatu subshell, dsb
{ }	Curly bracket (kurung kurawal)	Digunakan pada saat mengakses elemen dari array atau membuat blok kode dalam script
~	Tilde	Simbol yang mewakili home directory
;	Semicolon (titik koma)	Digunakan untuk memisahkan beberapa perintah yang diketik dalam satu baris, selain itu juga dapat digunakan untuk ??
	Vertical bar	Digunakan untuk operasi piping yang memungkinkan keluaran suatu perintah menjadi masukan bagi perintah lain
> dan <	Angle bracket (kurung siku)	Digunakan untuk operasi redirection yang memungkinkan keluaran suatu perintah disimpan ke dalam file atau device tertentu

## 19. Standard Input, Standard Output, Standard Error

Setiap perintah dalam shell memiliki satu jalur masukan dan dua jalur keluaran. Jalur masukan perintah disebut standard input sedangkan jalur keluaran terbagi menjadi standard output dan standard error. Jika perintah berhasil dijalankan tanpa ada kesalahan maka hasilnya akan dikeluarkan melalui standard output sedangkan jika perintah mengalami kegagalan atau error maka hasilnya akan dikeluarkan melalui standard error. Ilustrasi di bawah ini akan menggambarkan diagram suatu perintah dalam shell.

Pada ilustrasi di atas perintah

```
$ cat Linux.txt
```

akan menampilkan isi file Linux.txt yaitu “Linux” pada layar monitor sebagai default standard output.

Perintah di atas setara dengan menuliskan

```
$ cat < Linux.txt
```

Jika file Linux.txt ada maka perintah cat tidak akan menampilkan pesan kesalahan namun jika ternyata file Linux.txt tidak ada maka perintah cat akan gagal dan menampilkan pesan kesalahan.

Pada awalnya masukan terhadap suatu perintah yang melalui standard input dapat berasal dari keyboard atau file namun user disediakan fasilitas untuk mengubah asal dari masukan tersebut. Begitu pula dengan keluaran yang melalui standard output atau standard error pada awalnya selalu dikeluarkan pada layar monitor namun user dapat mengubah tujuan keluarannya jika dibutuhkan. Fasilitas yang disediakan oleh sistem operasi Linux untuk mengubah asal masukan dan tujuan keluaran perintah tersebut disebut **redirection** dan **piping**.

## 20.....Redirection

Redirection adalah fasilitas untuk mengubah asal masukan dan tujuan keluaran suatu perintah. Redirection digunakan jika asal masukan atau tujuan keluaran yang diinginkan berupa file atau perangkat tertentu (yang pada Linux direpresentasikan dalam bentuk file juga).

Redirection dilakukan dengan menggunakan karakter-karakter khusus di bawah ini.

Karakter	Kegunaan
>	Mengalihkan standard output perintah/proses dari layar monitor ke dalam file. Jika file sudah ada maka isi file akan ditimpa (overwrite) dan jika belum ada maka file akan dibuat.
>>	Mengalihkan output perintah/proses dari layar monitor ke dalam file. Jika file sudah ada maka isi file akan ditambah (append) dengan keluaran perintah/proses di bagian akhir file dan jika belum ada maka file akan dibuat.
2>	Mengalihkan standard error perintah/proses dari layar monitor ke dalam file. Jika file sudah ada maka isi file akan ditimpa (overwrite) dan jika belum ada maka file akan dibuat.
2>>	Mengalihkan standard error perintah/proses dari layar monitor ke dalam file. Jika file sudah ada maka isi file akan ditambah (append) dengan keluaran perintah/proses di bagian akhir file dan jika belum ada maka file akan dibuat.
2&>	Mengalihkan standard error perintah/proses dari layar monitor ke dalam file. Jika file sudah ada maka isi file akan ditimpa (overwrite) dan jika belum ada maka file akan dibuat.

<	Mengalihkan standard input perintah/proses dari keyboard menjadi dari file.
---	---

Contoh:

Perintah berikut akan mengubah standard output dari layar monitor menjadi sebuah file sehingga perintah cat dapat digunakan untuk menyalin file:

```
$ cat Linux.txt > Linux-copy.txt
```

Pada contoh perintah berikut jika perintah mengalami kegagalan maka pesan kesalahan akan disimpan dalam file dan tidak ditampilkan di layar monitor. Jika tidak gagal maka file yang menjadi tujuan baru dari standard error tidak akan berisi pesan kesalahan (sama seperti kondisi awal).

```
$ cat Linux.txt 2> Linux-copy.txt
```

## 21. Pemipaan (Piping)

Konsep piping sangat mirip dengan redirection hanya saja asal masukan dan tujuan keluaran bukanlah file melainkan proses lainnya. Piping dilakukan dengan menggunakan karakter “|” (vertical bar) untuk menandakan adanya piping antar dua perintah atau lebih.

Contoh:

Perintah berikut akan menampilkan bagian tanggal dari keluaran perintah date.

```
$ date | cut -d" " -f3
```

Pada baris perintah di atas perintah date tidak mendapat masukan dan hanya mengeluarkan tanggal dan tahun saat ini. Keluaran ini akan menjadi masukan bagi perintah cut yang akan memotong dan membuang bagian lain selain tanggal (tanpa bulan dan tahun) dan menampilkannya di layar monitor.

Piping dapat juga digunakan berkali-kali dalam satu baris perintah namun piping harus dilakukan antar proses dan hanya satu kali di antara 2 proses.

## **22.....Variabel dan Array**

Pada lingkungan pemrograman data atau informasi yang akan diolah membutuhkan tempat penyimpanan yang memiliki identitas agar dapat dibedakan dengan tempat penyimpanan lainnya. Tempat penyimpanan itu lebih dikenal sebagai variabel.

Pada shell Bash variabel dapat langsung dibuat dan diisi dengan menuliskan nama variabel diikuti dengan karakter “=” dan langsung diberi nilai tertentu.

```
# nama="kaprodi_ganteng"
```

Suatu variabel ketika hendak digunakan dalam perintah ditandai dengan awalan karakter \$ (dolar). Contohnya:

```
# echo $nama
```

```
kaprodi_ganteng
```

Tidak seperti bahasa pemrograman lainnya yang setiap variabel memiliki tipe data tertentu untuk menyimpan data dalam bentuk tertentu (contoh: string, integer, real/float, Boolean) maka Bash sebagai Linux shell tidak melekatkan tipe data pada variabel. Sebuah variabel berisi deretan karakter “123” dapat dianggap sebagai sebuah teks atau pada saat yang lain adalah integer (bilangan bulat) bergantung kebutuhan program. Contohnya:

```
# angka=123
```

```
echo "isi variabel angka adalah : $angka"
```

```
# expr $angka + 10
```

```
133
```

Pada contoh di atas isi variabel angka dapat dianggap sebagai teks tapi pada saat yang lain ia dapat dianggap sebuah bilangan bulat yang dapat dijumlahkan melalui perintah expr.



## 23. Ruang Lingkup Variabel

Variabel dalam shell programming seperti halnya variabel dalam pemrograman lain memiliki ruang lingkup (scope) di mana variabel tersebut dapat dikenali. Variabel yang dibuat dalam sebuah tingkatan shell tidak dikenali ditingkatan shell lainnya. Contohnya jika sebuah shell dimulai dari shell lain yang sudah aktif maka variabel pada kedua shell tidak dapat saling dikenali. Selain itu jika sebuah variabel dibuat pada sebuah blok (blok dapat dibuat menggunakan tanda kurung buka “(“ dan kurung tutup” )” ) maka variabel tidak akan dikenali dibagian lain di luar blok tersebut.

Meski demikian Linux menyediakan suatu perintah untuk membuat suatu variabel dapat dikenali pada subshell yang dimulai dari shell yang telah aktif. Perintah tersebut adalah `export`. Contoh penggunaan `export` adalah sebagai berikut:

```
# variabel=123
```

Tekan enter dan buat subshell baru:

```
# bash
```

Tekan enter dan sebuah subshell akan dijalankan dari shell yang aktif kemudian lihat isi variabel `variabel`.

```
# echo $variabel
```

```
#
```

Tekan `ctrl-d` untuk keluar dari subshell dan kembali ke shell awal dan lakukan perintah di bawah ini untuk mengekspor variabel:

```
# export variabel
```

Buatlah subshell yang baru:

```
# bash
```

tekan enter dan sebuah subshell akan dijalankan dari shell yang aktif kemudian lihatlah isi variabel `variabel`.

```
# echo $variabel
```

```
123
```

Sebuah shell script juga dianggap sebagai subshell dari sebuah shell yang aktif.

## 24. Array

Array merupakan variabel dengan kemampuan untuk menyimpan banyak nilai. Berbeda dengan bahasa pemrograman lain array pada bash script tidak perlu diinisialisasikan.

Perintah-perintah berikut akan menerangkan bagaimana cara mengoperasikan array:

1. Membuat array dengan isi tertentu:

```
$ nama=( Tafta Rossi Fitri Qoim )  
$ nama=( [0]=Tafta [1]=Rossi  
[2]=Fitri \  
[5]=Qoim )
```

2. Mengisi array pada indeks tertentu

```
$ nama[4]=Dahliar
```

3. Melihat jumlah elemen pada sebuah array

```
$ echo ${#nama[@]}  
$ echo ${#nama[*]}
```

4. Mengakses array pada indeks tertentu

```
$ echo ${nama[4]}
```

5. Melihat seluruh elemen pada sebuah array

```
$ echo ${nama[*]}
```

## BAB 7

### Shell Programming II

#### 7.1 Pengenalan Shell

Apa itu shell ? shell adalah program (penterjemah perintah) yang menjembatani user dengan sistem operasi dalam hal ini kernel (inti sistem operasi), umumnya shell menyediakan prompt sebagai user interface, tempat dimana user mengetikkan perintah-perintah yang diinginkan baik berupa perintah internal shell (internal command), ataupun perintah eksekusi suatu file program (eksternal command), selain itu shell memungkinkan user menyusun sekumpulan perintah pada sebuah atau beberapa file untuk dieksekusi sebagai program.

##### 7.1.1 Tipe – Tipe Shell

Tidak seperti sistem operasi lain yang hanya menyediakan satu atau 2 shell, sistem operasi dari keluarga unix misalnya linux sampai saat ini dilengkapi oleh banyak shell dengan kumpulan perintah yang sangat banyak, sehingga memungkinkan pemakai memilih shell mana yang paling baik untuk membantu menyelesaikan pekerjaannya, atau dapat pula berpindah-pindah dari shell yang satu ke shell yang lain dengan mudah, beberapa shell yang ada di linux antara lain:

- Bourne shell(sh),
- C shell(csh),
- Korn shell(ksh),
- Bourne again shell(bash),
- dsb.

Masing - masing shell mempunyai kelebihan dan kekurangan yang mungkin lebih didasarkan pada kebutuhan pemakai yang makin hari makin meningkat, untuk dokumentasi ini shell yang digunakan adalah bash shell dari GNU, yang merupakan pengembangan dari Bourne shell dan mengambil beberapa feature (keistimewaan) dari C shell serta Korn shell, Bash shell merupakan shell yang cukup banyak digunakan pemakai linux karena kemudahan serta banyaknya fasilitas perintah yang disediakan

##### 7.1.2 Tools Editor Shells Script

##### 7.1.3 Penggunaan Shell

kegunaan shell script antara lain :

- shell scripts dapat mengambil input dari user, file dan menampilkannya di layar
- bisa gunakan untuk membuat perintah sendiri
- menghemat waktu
- membuat kerja lebih efisien, dengan mengurangi pengetikan perintah berulang-ulang dan panjang.
- Untuk mengontrol *session* UNIX, contohnya: Set home dir., mail directory, startup file

#### 7.2 Memulai Script

##### 7.2.1 Membuat Script

Untuk membuat script cukup dengan mengedit file dengan menggunakan editor seperti vi atau emacs.

##### 7.2.2 Menjalankan Script

Setelah script dibuat sesuai dengan kebutuhan, maka agar bisa dijalankan maka harus diberi hak akses untuk eksekusi untuk itu cukup ketik :

```
$ chmod +x test_script.sh
```

Perintah diatas akan merubah hak akses file menjadi dapat dieksekusi. Kemudian ketik :

```
$ ./script.sh
```

Dan script akan jalan lalu menampilkan output yang diinginkan

### 7.3 Dasar – Dasar Shell

#### 7.3.1 Rutinitas menggunakan Variabel

##### 7.3.1.1 Lingkup penggunaan Variabel

##### 7.3.1.2 Overiding pada Variabel

#### 7.3.2 Karakter Pemisah

Ketika kamu mengeksekusi perintah di dalam program shell dipisah oleh baris baru, artinya perintah tersebut dieksekusi secara sequensial atau sesuai urutan didalam file. Pemisah berikut ini untuk mengontrol urutan eksekusi perintah.

Karakter &&

Pemisah ini berguna sebagai pemisah dengan kondisi dimana kondisi itu adalah perintah selanjutnya akan dieksekusi apabila perintah sebelumnya hasilnya sukses.

Contoh :

```
$ test -d /users/rhonda/tools && cd /users/rhonda/tools
```

Artinya sebelumnya akan melihat apakah /users/rhonda/tools merupakan direktori, jika iya maka perintah cd akan dieksekusi. Jika tidak, maka tidak ada tindakan selanjutnya.

Karakter ||

Fungsinya kebalikan karakter &&, yaitu pemisah dua garis vertical akan mengeksekusi perintah apabila perintah sebelumnya hasilnya tidak sukses

Contoh :

```
$ test -d /users/rhonda/tools || echo "Direktori tidak ada"
```

Artinya sebelumnya akan melihat apakah /users/rhonda/tools merupakan direktori, jika tidak maka perintah echo akan dieksekusi. Jika iya, maka tidak ada tindakan selanjutnya.

Apabila akan dipakai secara berbarengan maka kita gunakan karakter ; dan & untuk menghentikan rangkaian perintah yang bersangkutan

Contoh :

```
$ test -d /tools && cd /tools; test -z "$fn" || sort -o $fn $fn &
```

Artinya terdapat dua rangkaian perintah pertama adalah `test -d /tools && cd /tools` dan kedua adalah `test -z "$fn" || sort -o $fn $fn &`. Rangkaian pertama dieksekusi sebelumnya yang kedua ( karena ada operator `;` ). Jika yang rangkaian pertama hasilnya sukses maka perintah `cd` dieksekusi. Dan kemudia rangkaian kedua dieksekusi di background ( karena ada operator `&` ). Perintah kedua dijalankan dan jika hasilnya tidak sukses, maka perintah `sort` dilakukan.

### 7.3.3 Parameter

Parameter merupakan hal yang penting didalam pemrograman shell karena mereka lah yang memberikan nilai. Dan parameter dapat kita buat sendiri.

Contoh :

*parameter=nilai*

perhatikan bahwa harus tidak ada spasi antara *parameter*, tanda sama dengan (`=`), dan *nilai*. Parameter ini dapat dibuat pada saat di shell, lihat seperti contoh :

```
$ x=linux
```

Artinya variable `x` telah dibuat dan nilai diberikan nilai “linux”. Untuk mengakses variable `x`, cukup menambahkan tanda `$` sebelumnya nama variable.

```
$ echo $x
```

Perintah `echo` akan menampilkan output dari `x` ke layar.

### 7.3.4 Input Output pada Shell

#### 7.3.4.1 Pernyataan printf

Berguna untuk menampilkan output ke layar seperti halnya perintah `echo`. Pernyataan `printf` dapat diketik dengan :

```
$ printf ( parameter )
```

Dalam hal ini *parameter* bisa berupa kalimat atau suatu parameter yang telah didefinisikan sebelumnya.

#### 7.3.4.2 Pernyataan read

Jikalau ingin menerima input pada saat eksekusi program shell, maka dapat digunakan pernyataan `read` dengan format sebagai berikut :

```
$ read [ parameter ]
```

Ketika computer mengeksekusi pernyataan ini, maka akan menerima input dari keyboard ( kecuali input diarahkan dari suatu file ).

### 7.3.5 Menggunakan Kondisi

Didalam pembuatan script dibolehkan untuk melakukan suatu perintah apabila memenuhi kondisi tertentu yang disebut pernyataan kondisi, antara lain :

#### 7.3.5.1 Pernyataan if

Pernyataan if bisa mengeksekusi satu atau beberapa perintah jika kondisi tertentu terpenuhi.

Contoh :

```
if kondisi
then perintah_1
else perintah_2
fi
```

Artinya pernyataan if melihat apabila *kondisi* itu benar atau terpenuhi, maka *perintah\_1* dilaksanakan jika tidak terpenuhi maka *perintah\_2* akan dilaksanakan.

#### 7.3.5.2 Pernyataan case

Pernyataan *case* memudahkan untuk melihat beberapa kondisi dan melaksanakan perintah –perintah apabila memenuhi criteria kondisi

Contoh :

```
case string in
kriteria1 [ | kriteria2... ] ) perintah1 ;;
kriteria3 [ | kriteria4... ] ) perintah2 ;;
...
esac
```

Baris pertama menerima *string* yang kemudian di bandingkan dengan criteria-kriteria yang ada untuk dicocokkan. Jika ada criteria yang cocok, maka akan dilakukan perintah-perintah yang berkaitan.

#### 7.3.5.3 Pernyataan test

Pernyataan test ini mengevaluasi *expr*; jika *expr* dievaluasi hasilnya benar, maka test akan menghasilkan status 0 exit. Jika *expr* dievaluasi hasilnya salah, maka test akan menghasilkan status non 0 exit.

Contoh :

```
test expr
Atau
[ expr ]
```

Seperti contoh diatas pernyataan test dapat diganti oleh [ ]. Untuk lengkapnya bisa dilihat dengan perintah man test.

### 7.3.6 Menggunakan Pengulangan

Penggunaan proses sequential kadang tidak mencukupi kebutuhan operasional. Untuk dibutuhkan mekanisme yang memungkinkan pengulangan beberapa set perintah menggunakan nilai parameter yang berbeda. Untuk mendapatkan hal ini disediakan macam-macam pengulangan, antara lain :

#### 7.3.6.1 Pernyataan for

Berfungsi untuk mengeksekusi beberapa set perintah setiap nilai diberikan ke parameter.

Contoh :

```
for parameter [ in daftar_kata ]  
do daftar_perintah  
done
```

*parameter* bisa merupakan nama parameter apa saja, *daftar\_kata* merupakan rangkaian satu atau lebih nilai yang diberikan ke parameter, dan *daftar\_perintah* adalah rangkaian perintah yang akan dieksekusi setiap kali pengulangan. Jika *daftar\_kata* dihilangkan (termasuk in), maka *parameter* diberikan nilai konstan sebelumnya.

#### 7.3.6.2 Pernyataan while

Berfungsi untuk melakukan serangkaian perintah berulang-ulang.

Contoh :

```
While daftar_perintah1  
Do daftar_perintah2  
Done
```

Semua perintah di *daftar\_perintah1* dieksekusi. Jika perintah terakhir dalam daftar itu hasilnya sukses (dengan status exit adalah 0), maka perintah-perintah didalam *daftar\_perintah2* dieksekusi. Kemudian kembali lagi ke *daftar\_perintah1* sampai perintah terakhir hasilnya tidak sukses maka akan berhenti.

#### 7.3.7 Menggunakan Fungsi

Apabila tingkat kerumitan pemrograman meningkat maka akan membutuhkan pemilahan-pemilahan menggunakan fungsi. Dengan cara ini dapat memudahkan dengan membuat fungsi yang dapat dipakai kembali dan menghilangkan kode yang berulang.

Untuk membuat fungsi digunakan syntax sebagai berikut :

```
Nama() { daftar ; }
```

*nama* ini menjelaskan nama fungsi yang akan dibuat, dan *daftar* sendiri artinya serangkaian perintah yang digunakan di dalam fungsi tersebut. Sedangkan untuk memanggilnya atau mengaksesnya cukup ketik :

```
nama [ parameter ... ]
```

Dimana *nama* merupakan nama fungsi yang dipanggil, dan [ *parameter* ... ] merujuk kepada parameter opsional yang akan dimasukkan.

#### 7.3.8 Operasi Aritmatika pada Shell

Di dalam pembuatan shell kita akan menemui beberapa operasi aritmatika yang dibutuhkan dalam suatu operasi, untuk itu bisa kita gunakan sebagai berikut :

##### 7.3.8.1 Perintah expr

Fungsi dari perintah *expr* sangat berguna sekali untuk melakukan operasi aritmatika di dalam program shell. Dan juga mempunyai fungsi lain untuk memanipulasi kalimat.

Contoh :

Expr *ekspresi* { + - / \* } *ekspresi*

+	= penjumlahan
-	= pengurangan
/	= pembagian
*	= perkalian

### 7.3.9 Operator Logika pada Shell

Untuk memudahkan dalam pemrograman shell, maka perlu di masukkan operator logika apabila kita ingin menggunakan pengulangan atau menentukan suatu kondisi.

#### 7.3.9.1 Operator logika

Beberapa operator logika yang dapat kita gunakan antara lain :

==	= sama
!=	= tidak sama
!	= tanda negative atau kebalikannya
>	= lebih dari
<	= kurang dari
>=	= lebih dari atau sama dengan
<=	= kurang dari atau sama dengan
>>	= bergerak ke kanan
<<	= bergerak ke kiri
&&	= dan
	= atau

### 7.3.10 Konsep Return Code

Ketika fungsi atau perintah berhenti, maka akan terbentuk flag yang menunjukkan status dari berhenti tersebut. Dengan kata lain, jika suatu fungsi atau perintah hasilnya sukses setelah dieksekusi, maka akan menghasilkan nilai yang tandanya sukses. Nilai tersebut ( atau return code ) biasanya terdapat di bagian exit.

Jika mengeksekusi perintah yang salah, biasanya akan mendapat pesan kesalahan. Yang biasa terjadi adalah perintah shell membentuk return code. Jika seumpama return codenya adalah 2, maka akan muncul pesan kesalahan yang menandakan adanya kesalahan syntax atau perintah shell.

## 7.4 Mendebug Sscript

Ketika menggunakan pipa pada program shell akan menyulitkan pada saat mendebugnya karena hasil atau output dari perintah tidak terlihat. Satu saran untuk membantu pada saat mendebug pipa adalah menambahkan pernyataan cat di dalam pipa untuk menunjukkan hasil sementara. Sebagai contoh, cukup tambahkan perintah cat diikuti oleh sebuah exit di suatu tempat dalam pipa. Pipa tentunya akan menampilkan output pada saat itu, dan kemudian keluar dari program ( untuk menghindari kesalahan berikutnya, dan untuk menunjukkan dilokasi pipa yang mana saat itu ).

Sekarang, ketika akan siap untuk mengetes program, kita tidak perlu keluar dari editor (dengan asumsi yang digunakan adalah vi ), menjalankan program, melihat output, kemudian masuk kedalam vi lagi untuk melakukan perubahan. Dengan cara yang lebih efisien, cukup dengan menyimpan program terlebih



dahulu menggunakan perintah vi “:w”; jalankan program dari vi dengan perintah  
:  
: ! *skrip* [ *argument* ]

Perintah : ! artinya menjalankan program di dalam shell diluar vi. Ketika outputnya keluar, lalu kembali ke dalam vi ( ketika diminta ) lakukan perubahan-perubahan yang diperlukan dan dicoba lagi. Dapat juga mengeksekusi shell dari dalam vi ( dengan mengetik “: sh” ) kemudian mengeksekusi skrip.

## 7.5 **Latihan**

# MANAJEMEN FILESYSTEM DI LINUX

---

## Konsep dan Penomoran Device

Dalam Linux pada umumnya semua direpresentasikan sebagai file, termasuk juga dengan device. Setiap sistem dalam Linux mempunyai direktori yang berisi file yang special yang dinamakan *dev*. Setiap file yang ada pada direktori *dev* mewakili sebuah *device* atau *pseudo device*. File *device* mempunyai dua buah angka yang terasosiasi di dalamnya, nomor *device* mayor dan minor. Kernel Linux dapat mengetahui *device* apakah yang di representasikan dengan nomor tersebut. Sebagai contoh

```
$ file /dev/zero
/dev/zero: character special (1/5)
```

Perintah ini dapat digunakan untuk menentukan tipe sebuah file. File tertentu ini dikenali sebagai file *device* yang mempunyai angka mayor 1 dan 5 sebagai angka *device* minornya

Ketika kita mengakses sebuah file *device*, bilangan mayor akan menentukan *device driver* mana yang di panggil untuk melakukan operasi input/output. Pemanggilan *device driver* ini dilakukan dengan bilangan minor sebagai parameternya, dan interpretasi dari bilangan minor tersebut tergantung kepada *device driver* tadi. Dokumentasi *device driver* biasanya menjelaskan bagaimana *device driver* menggunakan bilangan minor

Kernel dalam Linux dapat menangani dua buah tipe *device*, yaitu *character* dan *block device*. *Character device* dapat membaca byte per byte sementara *block device* tidak. Apakah sebuah device di Linux akan digolongkan kedalam *character device* ato *block device* tergantung kepada karakteristik dan sifat yang di miliki oleh device tersebut. Sebagai contoh pada umumnya media penyimpanan akan digolongkan kedalam *block device* sementara input device tergolong kedalam *character device*

Sesuai konvensi, suatu file device memiliki nama tertentu dan program-program sistem mengetahui nama-nama tsb karena telah

tercompile kedalamnya. Program-program itu menginginkan disk IDE diberi nama `/dev/hd*` sedangkan disk SCSI diberi nama `/dev/sd*`. Disk-disk dinamai dengan a, b, c, dst, sehingga `/dev/hda` adalah disk IDE Anda yang pertama dan `/dev/sda` adalah disk SCSI Anda yang pertama. Kedua device tsb mewakili keseluruhan disk dimulai dari blok satu. Penulisan kedalam disk dengan alat yang salah akan merusak master boot loader dan tabel partisi dari disk tsb, menyebabkan semua data dalam disk tsb tidak terbaca atau membuat sistem anda tidak dapat di-boot

Penomoran device dalam Linux disesuaikan dengan perangkatnya, misalkan untuk melakukan penomeran pada hardisk dan DVD-ROM ATA dapat dilakukan sebagai berikut

Notasi	Arti
<code>/dev/hda</code>	perangkat master pada kanal ATA pertama
<code>/dev/hdb</code>	perangkat master pada kanal ATA pertama
Notasi	Arti
<code>/dev/hdd</code>	perangkat slave pada kanal ATA kedua

Misalkan pada sebuah komputer yang menggunakan dua buah hard disk dan sebuah DVD-ROM maka, hard disk pertama adalah perangkat master pada kanal ATA pertama dan hard disk kedua adalah slave pada kanal ATA pertama

Notasi	Arti
<code>/dev/hda1</code>	partisi pertama dari hard disk yang di representasikan oleh hda device file

/dev/hda2	partisi kedua dari hard disk yang direpresentasikan oleh hda device file
/dev/hdb1	partisi pertama dari hard disk yang direpresentasikan oleh hdb device file
/dev/hdb2	partisi kedua dari hard disk yang direpresentasikan oleh hdb device file

Untuk device lainnya seperti hard disk SCSI dan drive CD-ROM mengikuti konvensi penamaan lainnya SCSI tidak umum di pakai pada mesin biasa, tetapi perangkat USB dan Serial ATA (SATA) juga direpresentasikan sebagai disk SCSI maka, notasi berikut digunakan untuk penomoran perangkat SCSI

<b>Notasi</b>	<b>Arti</b>
/dev/sda	disk SCSI pertama
/dev/sdb	disk SCSI kedua
/dev/scd0	CD-ROM pertama
/dev/scd1	CD-ROM kedua

Nama partisi selalu dibuat dengan cara yang sama dengan disk ATA, jadi partisi pertama pada disk SCSI pertama akan di notasikan dengan */dev/sda1*

## Partisi Harddisk dan Manajemen Partisi Harddisk dengan Fdisk

Pada sistem operasi berbasis *Unix* dan *Unix-like* seperti Linux dan Mac OS X, pembuatan partisi dilakukan terpisah untuk */boot*, */home*, */tmp*, */user*, */var*, */opt*, *swap* dan seluruh file yang lain sangatlah memungkinkan. Skema partisi seperti ini memiliki beberapa keuntungan diantaranya jika salah satu *filesystem* mengalami kerusakan, data ataupun file-file milik sistem operasi tidak akan bermasalah. Untuk meningkatkan keamanan dari kehilangan sistem, partisi dapat dibuat dengan mode *read only*. Metode dengan membagi partisi menjadi ukuran-ukuran fix memiliki ketidaknyamanan, salah satunya yaitu pengguna dapan kehabisan *space* di partisi */home* walaupun *space* di partisi yang lain masih tersisa banyak. Oleh karena itu *user* diharapkan mampu memprediksi rencana penggunaan partisinya.

Didalam sistem operasi Linux terdapat 2 tipe partisi secara umum

1. Data partition : berisi data sistem operasi Linux, termasuk di dalamnya partisi root, terdapat semua data untuk melakukan start up dan menjalankan sistem.
2. Swap partition : ekspansi dari memori fisik sebuah komputer atau extra memori yang ada pada hard disk.

Untuk dapat membuat partisi di Linux ada beberapa syarat yang sebaiknya di penuhi:

1. Partisi yang hendak di partisi jangan sedang dalam keadaan terpakai atau termount,
2. Partisi sebaiknya dibuat dengan besar cylinder yang berurutan, jadi sebelumnya jangan menghapus partisi di tengah, hal ini bisa menghasilkan eror bahkan dengan partion magic hal ini akan menjadi masalah.
3. Partisi extended jangan di tempatkan di tengah tetapi di bagian belakang harddisk. Hal ini suka menjadi masalah

karena partisi selanjutnya yang bisa kita buat hanya partisi primary, hal ini menyebabkan keterbatasan dalam membuat partisi dalam jumlah banyak dan selain itu juga akan terjadi beberapa eror.

Pada sistem operasi Linux partisi dapat dilakukan dengan menggunakan perintah *fdisk*. Untuk menggunakan *fdisk* user harus login sebagai administrator terlebih dahulu, user bisa login sebagai administrator dengan menggunakan perintah *sudo* atau *su*.

Beberapa perintah penting dalam *fdisk* :

Command	Deskripsi
p	Menampilkan partition table harddisk
m	Menampilkan perintah-perintah yang bisa di eksekusi
n	Membuat partisi baru
d	Menghapus partisi
t	Mengganti tipe dari file system
l	Menampilkan jenis file system yang di support oleh linux
q	Keluar tanpa menyimpan partition table
w	Menyimpan partition table dan keluar

## 2.1 Melihat Daftar Partisi

Untuk melihat daftar partisi kita dapat mengetikan command *fdisk -l*

Contoh :

```
# fdisk-l
Device Boot      Start      End  Blocks   Id
System
/dev/sdb1    *          1      787 1007340    c   W95
FAT32 (LBA)
```

Disk /dev/sdc: 120.0 GB, 120034123776 bytes  
 255 heads, 63 sectors/track, 14593 cylinders  
 Units = cylinders of 16065 \* 512 = 8225280  
 bytes

```
Device Boot      Start      End  Blocks   Id
System
/dev/sdc1          1      5222  41945683+   7
HPFS/NTFS
/dev/sdc2      5223     10444  41945715    7
HPFS/NTFS
/dev/sdc3     10445     14593  33326842+   7
HPFS/NTFS
```

## 2.2 Membuat Partisi Pada Linux

Setelah kita mengetahui device yang akan kita buat maka kita akan memulai untuk membuat partisi, maka kita dapat mengetikkan perintah

```
# fdisk [device]
```

Contoh

1. Misalkan kita akan membuat partisi dalam device hda maka kita dapat menggunakan perintah sebagai berikut:

```
# fdisk/dev/hda
```

Setelah itu akan muncul tampilan:



The number of cylinders for this disk is set to 38166

There is nothing wrong with that, but this is larger than 1024, and could in certain setups cause problem with

- 1) software that runs at boot time (e.g., old version of LILO)

- 2) booting and partitioning software from other OSs (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):

## 2. Pada menu utama tulis n

Command (m for help): n

Command action

- e extended

- p primary partition (1-4)

Pilih partisi jenis p untuk menjadikannya primary partition, dan e untuk extended partition. Tampilan seperti di atas hanya muncul ketika harddisk yang akan kita partisi dalam keadaan kosong. Misalkan kita akan membuat sebuah primary partition dalam harddisk tersebut maka ketikkan p

Partition number (1-4):

Setelah kita menekan p maka akan ada menu imputan lagi untuk partition numbernya, sebaiknya kita menuliskannya secara berurutan maka ketikkan angka 1

First cylinder (1-3738, default 1):

Kemudian kita tentukan cylinder awalnya dari partisi yang akan kita buat. Untuk menempatkannya secara otomatis pada cylinder yang paling awal dari partisi yang kosong maka tekan saja enter

```
Last cylinder or +sizeM or +sizeK (1-3728, default 3738):
```

Setelah itu kita akan menentukan cylinder terakhir dari partisi yang akan kita buat. Kalau dengan cylinder kita hanya tinggal menuliskan nomor cylindernya misal 2000. Sedangkan menulis besarnya dengan cara menulis tanda + di depan kemudian besarnya berapa kalau damlah jumlah byte kita tidak perlu menambahkan apa-apa dilakangm kalau dalam kilobyte tambahkan K di belakang sedangkan apabila dalam megabyte tambahkan M dibelakang sebagai contoh: +100 = 100byte, +100K = 100Kbyte dan +1000M = 1000mbyte.

Misalkan angka yang kita masukan adalah +1000M maka tampilan partisi yang telah dibuat akan menjadi:

Device	Boot	Start	End	Blocks	Id
System					
/dev/hda1		1	123	987966	83
Linux					

Setelah melakukan semua proses, ketika partisi yang kita susun sedemikian rupa, hal ini belum di rubah pada harddisk kita maka kita harus menyimpan setingan partisi dengan menggunakan w pada menu utama dan apabila kita hendak membatalkan setingan tersebut kita dapat mengetikan q pada menu utama. Setelah kita membuat partisi maka kita harus mereboot terlebih dahulu pc kita.

## 2.3 Mengganti Tipe Partisi Harddisk

File sistem yang dibentuk secara default adalah *ext3*. Agar kita mendapatkan tipe partisi yang sesuai dengan yang kita inginkan maka kita dapat mengganti tipe partisi yang sudah kita buat di Linux dengan menggunakan *t*

Contoh :

```
# fdisk/dev/hda
```

```
The number of cylinders for this disk
is set to 38166
```

```
There is nothing wrong with that, but
this is larger than 1024, and could in
certain setups cause problem with
```

```
1)software that runs at boot time
(e.g., old version of LILO)
```

```
2)booting and partitioning software
from other OSs (e.g., DOS FDISK, OS/2
FDISK)
```

```
Command (m for help):
```

**Ketikan t pada menu**

```
Command (m for help): t
```

```
Partition number (1-5):
```

**Pilih partisi yang hendak di rubah tipenya**

```
Hex code (type L to list the code):
```

**Pilih tipe partisi yang akan digunakan:**

Tipe partisi	Code
FAT 32	b

extended	5
ext3	83
swap	82

## 2.4 Menghapus Partisi Pada Linux

Untuk melakukan penghapusan partisi pada Linux kita dapat melakukannya dengan menuliskan *d* pada menu

Contoh:

```
# fdisk/dev/hda
```

```
The number of cylinders for this disk
is set to 38166
```

```
There is nothing wrong with that, but
this is larger than 1024, and could in
certain setups cause problem with
```

```
1)software that runs at boot time
(e.g., old version of LILO)
```

```
2)booting and partitioning software
from other OSs (e.g., DOS FDISK, OS/2
FDISK)
```

```
Command (m for help):d
```

```
Partition number (1-5):
```

Pilihlah partisi yang akan dihapus

Setelah memilih partisi yang akan kita hapus dan menekan enter maka secara otomatis partisi yang kita pilih tersebut akan terhapus dari Linux.

## Pengenalan /etc/fstab Pada Linux

Pada sistem operasi Linux ada sebuah file yang disebut **/etc/fstab**. Pada bagian ini kita akan mempelajari tentang *fstab* bagaimana file tersebut digunakan dan apa keterkaitannya dengan perintah

*mount*. Apabila kita telah memahami dan mempelajari file *fstab* tersebut maka kita akan dapat dengan mudah mengedit dan melakukan konfigurasi terhadap isinya

*Fstab* pada Linux adalah sebuah file konfigurasi yang berisi informasi dari semua partisi dan media penyimpanan yang ada pada PC kita. File tersebut berada didalam device */etc*.

*/etc/fstab* menyimpan informasi dimana partisi dan media penyimpanan kita harus di mount dan bagaimana caranya. Ketika kita tidak bisa mengakses partisi Windows kita ketika kita berada di dalam Linux, tidak bisa memount DVD drive atau menulis data pada floppy disk sebagai user biasa maka kemungkinan terdapat kesalahan konfigurasi pada */etc/fstab* dan kita bisa memperbaikinya dengan mengedit file *fstab* tersebut.

*/etc/fstab* merupakan sebuah file text biasa, jadi kita dapat membuka dan mengeditnya di dalam berbagai text editor yang biasa kita pakai. Tetapi kita harus login terlebih dahulu sebagai root untuk dapat mengedit file yang ada pada */etc/fstab*. Contoh isi yang terdapat dalam file */etc/fstab*

#<device>	<mountpoint>	<fstype>	<option>	<dump>	<fsckorder>
/dev/hda2	/	ext2	defaults	1	1
/dev/hda1	/dos	vfat	defaults	0	0
/dev/hda4	/usr	ext2	defaults	1	2
/dev/hda3	swap	swap	defaults	0	0
/dev/fd0	/mnt/floppy	ext2	noauto	0	0
/dev/cdrom	/mnt/cdrom	iso9660	noauto,ro	0	0
/none	/proc	proc	defaults	0	0

Didalam file */etc/fstab* terdapat beberapa kolom, kolom-kolom tersebut mempunyai arti:

- *Device name*, dengan kata lain menunjukan lokasi dari sebuah partisi atau sumber data.
- *UUID*, merupakan acronim dari *Universally Unique Identifier*. Merupakan string unik yang di tujuan untuk device kita, pada Linux modern seperti *Ubuntu* dan *Fedora* menggunakan parameter *UUID* menggantikan device name sebagai parameter pertama pada */etc/fstab*

- *Mount point*, menunjukan dimana data tersebut di *attachkan* di dalam filesystem.
- *Filesystem type*, menunjukan tipe file sistem atau algoritma yang di gunakan untuk menginterpretasikan file sistem.
- *Option*, disertakan apabila file sistem harus dimount terlebih dahulu pada saat booting.
- *Dump*, berhubungan dengan program *dump* nilai opsi pada *dump* adalah 1 atau 0, apabila value nya kita set menjadi 0 maka *utility dump* tidak akan membackup file sistem, apabila kita set menjadi 1 maka *utility dump* akan membackup file sistem tersebut.
- *Fsck order*, opsi ini juga memiliki nilai antara 0,1 dan 2. 0 mendenotasikan program *fsck* tidak melakukan pengecekan terhadap file sistem. Apabila value tersebut kita ubah menjadi 1 maka, program *fsck* akan melakukan *checking* terhadap file sistem secara periodik atau ketika terjadi *physical errors* pada device tersebut

### 3.2 Membuat File Sistem Dalam Linux

Apabila kita telah membuat sebuah partisi di dalam Linux kemudian kita ingin menentukan file sistem untuk partisi tersebut, kita dapat menggunakan perintah **mkfs**.

Sintaks umum pada **mkfs**:

**Mkfs** [-V] [-t fstype] [**fs-options**] filesys [blocks]

Perintah **mkfs** digunakan untuk membangun sebuah filesystem didalam Linux, biasanya pada partisi sebuah harddisk, filesys biasanya berupa *device name* (misalkan */dev/hda1*, */dev/sdb2*) ataupun bisa berupa mount pointnya (misal, */*, */usr*, */home*) untuk file sistemnya *blocks* adalah jumlah block yang digunakan untuk file sistem tersebut.

Nilai kode yang dikembalikan oleh perintah **mkfs** adalah 0 apabila **mkfs** berhasil dijalankan dan 0 apabila proses tersebut gagal.

Opsi	Keterangan
-V	Menghasilkan output yang verbose, termasuk didalamnya semua file perintah <i>system-specific</i> yang dieksekusi. Menspesifikasikan opsi ini lebih dari sekali akan mempengaruhi file sistem yang lain, hal ini berguna untuk testing saja

-t fstype	Menspesifikasikan file sistem yang akan dibangun, apabila tidak maka file sistem default yang digunakan adalah ext2
fs-option	File sistem yang mempunyai opsi yang spesifik akan dilanjutkan pada file <i>system builder</i> yang sesungguhnya, meskipun tidak ada jaminan tetapi opsi ini telah banyak didukung oleh kebanyakan <i>system builders</i>
-l filename	Membaca <i>bad block list</i> dari sebuah <i>filename</i>
-v	Menghasilkan output secara verbose

Contoh:

Misalkan kita ingin membuat filesistem pada partisi yang baru saja kita buat dengan file sistem default (ext2) maka perintahnya adalah :

```
# ./mkfs /dev/hdb1
mke2fs 1.32 (09-Nov-2002)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
9781248 inodes, 19537040 blocks
976852 blocks (5.00%) reserved for the super user
First data block=0
597 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200,
884736, 1605632, 2654208,
    4096000, 7962624, 11239424
```

```
Writing inode tables: done
Writing superblocks and filesystem accounting
information: done
```

This filesystem will be automatically checked every

```
23 mounts or  
180 days, whichever comes first. Use tune2fs -c or  
-i to override.
```

Lalu edit pada file `/etc/fstab`, tambahkan pada baris terakhir file tersebut

```
[b]/dev/hda1 /mnt/hddext ext2 defaults 1 1 [/b]
```

### 3.3 Mounting Pada Linux

Dalam mengenali suatu device Linux akan mengenalnya sebagai suatu file yang letaknya berada di bawah direktori `/dev`, untuk dapat mempergunakan file yang ada dalam suatu device terlebih dahulu kita harus memount file tersebut ke suatu direktori agar dapat di pergunakan.

Semua file yang dapat diakses di dalam sistem Unix tersusun dalam sebuah tree yang besar, file tersebut mempunyai hierarki dan berakar ke `/`. File ini dapat disebarkan pada berbagai macam device. Perintah **mount** digunakan untuk mengaitkan atau menghubungkan file sistem yang ada pada sebuah device menuju tree tersebut

Perintah mount standar dapat dinotasikan sebagai berikut:

```
#mount -t type device dir
```

Notasi ini akan memberi informasi kepada kernel untuk menghubungkan sebuah file sistem yang terdapat pada sebuah device (yang mempunyai tipe, *type*) dan direktori *dir*. Content sebelumnya dan owner serta mode dari *dir* menjadi invisible selama file tersebut tetap dalam keadaan *termount*.

Sementara itu reverse dari perintah *mount* yakni *unmount* digunakan untuk memutuskan hubungan dari sebuah file sistem yang di tuju daari sebuah hierarki file, sebuah file sistem tidak dapat di unmount ketika file sistem tersebut dalam keadaan *busy* misalkan, ketika ada file yang di buka pada file sistem tersebut atau ketika sebuah proses yang sedang bekerja ada dalam direktori tersebut, atau ketika swap file sedang digunakan.

Pada bagian ini kita akan mencoba melakukan mounting untuk CD-ROM dan USB.

#### 3.3.1 Mounting CD-ROM



misalkan kita ingin melakukan mounting untuk CD-ROM pada direktori */mnt/cdrom*. Apabila kita belum memiliki direktori */mnt/cdrom* maka kita harus membuat direktori *cdrom* di */mnt* terlebih dahulu. Setelah kita memastikan telah memiliki direktori */mnt/cdrom* maka untuk memount CD-ROM tersebut kita harus mengetikkan command:

```
#mount /dev/cdrom /mnt/cdrom
```

perintah ini akan mengkoneksikan device pada *"/dev/cdrom"* (pada umumnya adalah CD-ROM drive) menuju direktori *"/mnt/cdrom"* sehingga kita dapat mengakses file dan direktori yang ada pada CD-ROM disk pada direktori *"/mnt/cdrom"*.

Mount point pada Linux untuk berbagai device telah didefinisikan di dalam file */etc/fstab*. Beberapa distro Linux menggunakan program yang bernama *automount*, yang secara otomatis akan melakukan mount partisi dan device sesuai dengan definisi yang ada pada file */etc/fstab*.

Sedangkan untuk melakukan *unmount* pada CD-ROM yang telah kita mount pada direktori */mnt/cdrom* tadi kita dapat melakukannya dengan mengetikkan perintah:

```
#unmount /mnt/cdrom
```

### 3.3.2 Mounting USB Pada Linux

Pada bagian ini kita akan melakukan mount sebuah usb flasdisk di Linux, sama seperti halnya dengan CD-ROM pertama-tama terlebih dahulu kita buat direktori **usbsaia** atau dengan nama lainnya di direktori */mnt* dengan menggunakan perintah **mkdir**

```
#mkdir /mnt/usbsaia
```

setelah direktori terbentuk maka kita bisa langsung melakukan mount usb kita pada direktori yang telah kita buat tersebut.

```
#mount /dev/sda1 /mnt/usbsaia
```

Apabila kita tidak ingin melakukan mount ulang terhadap device-device yang sering kita pakai setelah kita mereboot komputer, kita dapat melakukan mount secara otomatis, caranya adalah dengan menambahkan script di **/etc/fstab**

Untuk menulis di **/etc/fstab** ini kita bisa memakai vi, vim, kate, emacs, kwrite dan sebagainya.

Contoh:

Misalkan kita ingin untuk setiap reboot kita tidak harus memount ulang usb kita agar file yang ada di dalamnya bisa kita akses maka caranya adalah sebagai berikut

Penulisan di dalam file **/etc/fstab** adalah sebagai berikut

Nomor partisi, mount point dan format opsi

Sehingga notasinya menjadi seperti ini

```
/dev/sda1 /mnt/usbsaia default 0 0
```

0 = tidak akan di check oleh sistem ketika melakukan booting

1 = di check oleh sistem ketika komputer melakukan booting

# Manajemen Proses dan Penjadwalan Proses di Linux

---

## 1. Pengertian Proses

Proses adalah program yang sedang berjalan dalam sistem komputer, baik dalam kondisi aktif maupun berada di background. Setiap proses yang berjalan dalam sistem komputer memiliki status atau keadaan dalam suatu waktu. Dalam Linux status tersebut antara lain :

1. TASK\_RUNNING

Pada status ini, proses sedang ataupun siap dieksekusi oleh CPU.

2. TASK\_INTERRUPTIBLE

Pada status ini, proses sedang menunggu sebuah kondisi. Interupsi, sinyal, ataupun pelepasan sumber daya akan membangunkan proses.

3. TASK\_UNINTERRUPTIBLE

Pada status ini, proses sedang tidur dan tidak dapat dibangunkan oleh suatu sinyal.

4. TASK\_STOPPED

Pada status ini proses sedang dihentikan, misalnya oleh sebuah debugger.

5. TASK\_ZOMBIE

Pada status ini proses telah terhenti, tetapi masih memiliki struktur data `task_struct` di task vector dan masih memegang sumber daya yang sudah tidak digunakan lagi.

## 2. Manajemen Proses

Aktivitas yang berkaitan dengan manajemen proses antara lain :

1. Melihat status proses-proses yang dijalankan sistem
2. Menghentikan proses tertentu yang sedang berjalan
3. Memulai, menunda, dan menghentikan proses di console

### 2.1 Melihat Proses

Untuk melihat proses yang sedang berjalan dapat menggunakan perintah **ps** (*process status*) dan **top**. Perbedaan antara perintah **ps** dan **top** adalah, jika pada perintah **ps** sistem hanya akan menampilkan daftar dan keadaan proses satu kali setelah perintah **ps** dijalankan, sedangkan pada perintah **top**, sistem akan menampilkan daftar dan status proses selama user belum keluar

dari utilitas **top**. Selain itu daftar dan status proses akan diperbaharui setiap saat secara periodik.

### 2.1.1 Perintah **ps**

Sintaks untuk menggunakan perintah **ps** adalah : **ps –option**

Cara mudah untuk menggunakan perintah **ps** adalah dengan mengetikkan :

```
$ ps -aux
```

Maka perintah **ps** tersebut akan menampilkan semua proses yang dijalankan oleh user dalam format sebagai berikut :

<b>User</b>	: Pemilik proses
<b>PID</b>	: Identitas unik proses berupa bilangan
<b>TTY</b>	: Terminal dimana proses berjalan
<b>TIME</b>	: Waktu prosesor yang digunakan sejak proses dimulai
<b>%CPU</b>	: Penggunaan CPU oleh proses
<b>%MEM</b>	: Penggunaan memori (dalam KB) oleh proses
<b>VSZ</b>	: Penggunaan Virtual Memory dalam proses
<b>RSS</b>	: Penggunaan memori yang resident oleh proses
<b>TTY</b>	: terminal (tty), tempat asal proses tersebut dijalankan
<b>STAT</b>	: Status dari proses (Running, Sleep, Zombie, dll)
<b>-R</b>	: Proses dalam kondisi Running
<b>-S</b>	: Proses dalam kondisi Sleeping
<b>-I</b>	: Proses dalam kondisi Idle
<b>-Z</b>	: Proses Zombie
<b>-D</b>	: Uninterruptible sleep (biasanya IO)
<b>-&lt;</b>	: Proses dalam prioritas tinggi
<b>-N</b>	: Proses dalam prioritas rendah
<b>-</b>	dan lain-lain.
<b>START</b>	: Saat proses dijalankan
<b>TIME</b>	: Lama proses dijalankan
<b>COMMAND</b>	: Baris perintah yang dijalankan

### 2.1.2 Menggunakan pstree

Perintah **pstree** juga merupakan perintah yang digunakan untuk menampilkan daftar proses pada sistem, tetapi perintah ini akan menampilkan dalam bentuk tree.

Sintaks :

```
pstree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u] [-Z] [-G]
[-U] [pid|user]
```

Contoh :

```
# pstree root
```

Perintah di atas akan menampilkan proses-proses yang digunakan oleh root dalam bentuk tree.

### 2.1.3 Perintah top

Sintaks :

```
# top -hv | -bcisS -d delay -n iterations [- u user | -U
user] -p pid [, pid ...]
```

Perintah **top** memiliki beberapa tombol shortcut yang dapat digunakan untuk mengakses fungsionalitas yang disediakan oleh perintah **top**, antara lain :

Tombol	Fungsi
h	Menampilkan halaman bantuan
a	Mengurutkan tampilan proses dari yang terbaru s/d yang paling tua
m	Menampilkan atau menghilangkan tampilan informasi memori
k	Menghentikan proses tertentu
q	keluar

Contoh :

```
$ top
```

(akan ditampilkan daftar proses dan status proses yang sedang berjalan, dan daftar ini akan di-update setiap saat secara periodik.)

(tekan [q] untuk keluar dari utilitas top)

## 2.2 Menghentikan dan mematikan proses

Untuk menghentikan sebuah proses yang sedang berjalan langkah pertama adalah harus mengetahui PID dari proses tersebut. Untuk itu kita dapat menggunakan perintah **top** dan **ps** untuk dapat mengetahui PID dari proses tersebut. Setelah kita mengetahui PID dari proses tersebut maka proses tersebut dapat dihentikan dengan menggunakan perintah kill.

Sintaks :

```
kill -signal pid
```

- signal → mengirimkan sinyal dengan nilai tertentu (Default 15) pada SIGTERM (Software Termination Program)
- pid → proses id (dapat dilihat dengan perintah ps)

Contoh :

```
$ updated &  
[1] 9659  
$ kill 9659
```

Perintah di atas akan mengirimkan sinyal 15 pada proses 9659

Perintah kill dapat disertai dengan opsi tipe sinyal yang akan digunakan untuk menghentikan proses, yaitu :

Signal (Nomor)	Keterangan
HUP (1)	Hangup
INT (2)	Terminal Interrupt
QUIT (3)	Terminal Quit
ABRT (6)	Process Abort
KILL (9)	Kill
ALRM (14)	Alarm Clock
TERM (15)	Termination

Untuk mengetahui jenis sinyal yang lebih lengkap, jalankan perintah kill -l.

Proses yang berjalan bisa saja dihentikan atau berhenti Karena berbagai sebab seperti yang ditulis dalam kolom keterangan. Salah satu kegunaan mengirimkan sinyal yang dapat disesuaikan dengan kondisi tertentu adalah untuk menguji program yang memiliki prosedur penanganan sendiri ketika diberi sinyal tertentu.

### **2.3 Memulai, Menunda, dan Menghentikan Proses di Console**

Proses dapat dibagi menjadi 2 jenis yaitu proses yang berjalan di foreground dan proses yang berjalan di background.

Perbedaan proses foreground dengan proses background antara lain, proses yang berjalan di foreground adalah proses yang berinteraksi dengan user dan bahkan beberapa proses foreground juga menampilkan user sebuah user interface. Sedangkan proses yang berjalan di background adalah proses yang tidak berkaitan dengan user tertentu tetapi memiliki fungsi-fungsi khusus. Selain itu pada proses foreground, user harus menunggu sebuah proses selesai baru dapat menjalankan proses yang lain. Sedangkan pada proses background tidak harus menunggu sampai sebuah proses selesai untuk menjalankan proses yang lain.

#### **2.3.1 Memulai proses di foreground**

Untuk memulai sebuah proses di foreground dapat menggunakan sintaks :

```
$ command
```

Contoh :

```
$ updatedb
```

Jika menjalankan proses foreground maka prompt selanjutnya tidak akan muncul hingga proses selesai.

#### **2.3.2 Memulai proses di background**

Untuk memulai sebuah proses di background hampir sama dengan menjalankan proses di foreground hanya saja untuk menjalankan proses di background tambahkan spasi dan tanda ‘&’ (ampersand) di akhir perintah.

Sintaks :

```
$ command &
```

Contoh :

```
$ updatedb &
```



Seperti yang telah dijelaskan sebelumnya proses di background tidak harus menunggu sampai proses selesai dijalankan untuk menjalankan proses lain. Setelah memasukkan perintah untuk menjalankan proses di background maka prompt akan kembali muncul meskipun proses yang berjalan di background belum selesai. Sehingga user dapat memasukkan perintah lagi untuk menjalankan proses baik itu di foreground ataupun background. Proses background juga berada dalam prioritas yang lebih rendah dibandingkan dengan proses di foreground.

### **2.3.3 Menunda, menghentikan, dan melanjutkan proses**

Proses yang berjalan di console dapat ditunda atau dihentikan sementara dengan menekan **[Ctrl]+[z]**, maka proses yang berjalan akan ditunda dan diberi nomor identitas penundaan. Kemudian untuk melanjutkan kembali proses yang ditunda ketikkan **%[nomor\_id\_penundaan]** pada prompt untuk melanjutkan kembali proses yang ditunda.

Contoh :

```
$ info ls
(tekan [Ctrl]+[z])
[1]+stopped info ls
```

Kemudian untuk melanjutkannya kembali ketikkan :

```
$ %1
```

Selanjutnya untuk membatalkan atau menghentikan proses yang sedang berjalan di console tekan **[Ctrl]+[c]**, maka proses akan dibatalkan atau dihentikan.

Contoh :

```
$ info ls
(tekan [Ctrl]+[c])
```

### **2.3.4 Memindahkan proses ke background**

Adakalanya jika ingin untuk menjalankan berbagai macam perintah secara bersamaan tetapi console sedang menjalankan proses di foreground dan tidak ingin pindah ke console lain, sedangkan proses foreground yang berjalan belum selesai. Salah satu solusinya adalah dengan memindahkan proses yang berjalan

di foreground ke background sehingga prompt console dapat tersedia untuk perintah lain.

Langkah pertama adalah menunda proses yang sedang berjalan di foreground, caranya adalah dengan menekan **[Ctrl]+[z]**. Kemudian sebelum memindahkan proses ke background sebelumnya harus diketahui **jobnumber** dari proses yang sedang berjalan di console. Untuk mengetahui proses yang sedang dijalankan oleh console dan **jobnumber**-nya gunakan perintah **jobs**.

Contoh :

```
$ info ls
(tekan [Ctrl]+[z])
$ jobs
[1]+stopped info ls
```

Setelah mendapatkan jobnumber dari proses yang ingin dipindahkan ke background kemudian gunakan perintah **bg %jobnumber** untuk memindahkan proses dari foreground ke background.

Contoh :

```
$ bg %1
```

Jika ingin memindahkan kembali ke foreground gunakan perintah **fg %jobnumber**.

Contoh :

```
$ fg %1
```

### 3. Penjadwalan dengan at dan cron

Di Linux, user dapat menentukan kapan suatu tugas atau program di-eksekusi. Pengaturan waktu atau penjadwalan ini dapat berlangsung secara berulang maupun sesaat (hanya sekali eksekusi) diwaktu yang akan datang. Misalkan jika ingin menjalankan pekerjaan pada saat tidur, atau ingin menjalankan suatu tugas atau pekerjaan yang rutin dilakukan tanpa harus menjalankannya secara manual, maka gunakan fasilitas penjadwalan. Penjadwalan di linux dapat dilakukan dengan menggunakan perintah **at** dan **cron**.

### 3.1 Penjadwalan dengan menggunakan **at**

Perintah **at** digunakan untuk melakukan penjadwalan pekerjaan yang dilakukan hanya satu kali pada waktu tertentu dan tidak periodik.

Sintaks :

```
$ at [-v] [-q Queue] [-f File] [-mldbv] TIME
```

- Opsi **-v** berfungsi untuk menampilkan waktu kapan suatu pekerjaan akan dilakukan sebelum membaca pekerjaan tersebut.
- Opsi **-q** berfungsi untuk meningkatkan nilai antrian dari pekerjaan.
- Opsi **-f** berfungsi untuk membaca perintah yang akan dijalankan dari file tertentu.
- Opsi **-m** untuk mengirim mail kepada user yang menjalankan **at**, **-l** artinya list yaitu menampilkan list penjadwalan yang telah dibuat, **-d** artinya delete.
- **TIME** artinya waktu kapan pekerjaan yang dijadwalkan akan dijalankan.

Langkah-langkah penjadwalan dengan menggunakan perintah **at** adalah sebagai berikut :

1. Ketikkan perintah **at** diikuti waktu kapan pekerjaan akan dilaksanakan, maka anda akan memasuki subshell **at**.
2. Ketikkan perintah yang ingin dikerjakan. Untuk membuat baris baru tekan **[ENTER]**.
3. Setelah selesai mengetikkan perintah yang ingin dikerjakan buat baris baru dan tekan **[Ctrl]+[d]** untuk mengakhiri.

Contoh :

```
$ at 21.00
```

```
Warning : commands will be executed using (in order) a)
```

```
#SHELL b) login shell c) /bin/sh
```

```
at>echo hai
```

```
at><d^>
```

```
job 11 at 2009-04-27 21.00
```

Maka **at** akan menjalankan perintah “echo hai” pada jam 21.00. Hal yang perlu diingat bahwa **at** tidak akan menampilkan output dari perintah yang dijalankan di layar.

Format waktu yang diterima oleh perintah **at** diantaranya : now, noon, midnight, today, 4pm, 4 am, dll. Waktu yang digunakan dapat ditulis secara mutlak atau relatif. Misalnya jika ingin menjalankan sebuah program pada pukul 4 sore, 3 hari kemudian (misalkan hari ini tanggal 27 April 2009), maka ada 2 cara untuk menuliskan waktu yang digunakan :

at 4pm + 3days

atau

at 4pm April 30

Untuk keterangan yang lebih lengkap dapat dilihat pada file :

/usr/share/doc/at-no.versi/timespec

Untuk menampilkan penjadwalan yang telah dibuat gunakan perintah **atq** atau **at -l**.

Sintaks :

```
atq [-V] [-q queue]
```

Untuk membatalkan penjadwalan yang telah dibuat gunakan perintah **atrm job\_id** atau **at -d job\_id**. Job\_id dapat diperoleh dari perintah **atq**.

Sintaks :

```
atrm [-V] job [job...]
```

Contoh membuat, menampilkan, dan membatalkan penjadwalan :

```
$ at 21.00
```

```
Warning : commands will be executed using (in order) a)
#SHELL b) login shell c) /bin/sh
```

```
at>echo hai
```

```
at><d^>
```

```
job 11 at 2009-04-27 21.00
```

```
$ atq
```

```
11
```

```
$ atrm 11
```

Fungsionalitas **at** diaktifkan oleh sebuah daemon (program yang berjalan di background) bernama atd. User account yang diperbolehkan menggunakan fasilitas at dapat ditentukan melalui

file `/etc/at.deny` dan `/etc/at.allow`. Gunakan salah satunya saja. Jika file `/etc/at.allow` ditemukan maka konfigurasi pada file `/etc/at.deny` akan diabaikan. Isi dari file-file konfigurasi ini hanyalah daftar nama-nama user yang ada pada sistem.

Contoh :

```
$ cat /etc/at.deny
```

```
Guest
```

```
Cracker
```

```
Lazy
```

### 3.2 Penjadwalan dengan menggunakan cron

**Cron** adalah utilitas lain yang dapat digunakan untuk melakukan penjadwalan selain **at**. Jika pada **at** penjadwalan dilakukan hanya pada pekerjaan yang dilakukan sekali pada waktu tertentu, maka **cron** berfungsi untuk menjalankan pekerjaan yang dijadwalkan secara regular atau periodik. Biasanya file crontab yang dibuat disimpan di `/etc/spool/cron/nama_user`

Sintaks :

```
crontab [-u user] file
```

```
crontab [-u user] { -l | -r | -e }
```

- Opsi `-l` digunakan untuk menampilkan penjadwalan yang telah dibuat oleh user.
- Opsi `-r` digunakan untuk menghapus penjadwalan yang telah dibuat.
- Opsi `-e` digunakan untuk mengedit penjadwalan yang telah dibuat dengan menggunakan editor default pada variable environment system.

Langkah-langkah menggunakan **cron** :

- Ketikkan **crontab -e** pada prompt console.
- Anda akan memasuki editor vi pada mode perintah, tekan tombol `[ i ]` untuk mulai menuliskan perintah beserta waktunya.
- Tekan tombol `[escape]`, kemudian ketikkan **:wq** dan tekan `[enter]`.

Setiap crontab memiliki enam bagian inputan atau entry, yaitu :

1. Menit

2. Jam
3. Tanggal (dalam 1 bulan)
4. Bulan
5. Hari
6. Perintah

Format inputan yang digunakan untuk menit adalah 0 – 59, untuk jam adalah 0 – 23, sedangkan untuk tanggal dan bulan adalah 1 – 31 dan 1 – 12, sedangkan untuk hari 0 – 6 dimana 0 adalah hari minggu dan 6 adalah hari sabtu, atau dapat juga ditulis sun, mon, tue, dst. Entry yang berupa waktu pada crontab dapat ditulis tunggal atau berupa rentang waktu seperti 0 – 10 atau sun – fri. Selain itu dalam satu bagian entry dapat juga berisi beberapa waktu yang berbeda dan penulisannya dipisahkan dengan koma. Contoh : 0, 10, 20, 30.

Contoh penggunaan **cron** :

```
$ crontab -e
```

(setelah menekan tombol i, ketik baris di bawah)

```
0,20,40 22-23 * 7 fri-sat rm -f /tmp/* 2>& /var/log
```

(tekan esc, ketikkan :wq dan tekan enter)

Penulisan waktu di atas berarti setiap 20 menit pada jam 10 sampai 11 malam pada tanggal berapapun pada hari jumat dan sabtu di bulan Juli perintah `rm -f /tmp/* 2>& /var/log` akan dijalankan.

Untuk menampilkan dan menghapus penjadwalan yang telah dibuat gunakan opsi `-l` dan `-r` pada perintah **crontab**.

Contoh :

`$ crontab -l` → menampilkan penjadwalan yang telah dibuat.

```
0,20,40 22-23 * 7 fri-sat rm -f /tmp/* 2>& /var/log
```

`$ crontab -r` → menghapuskan penjadwalan yang telah dibuat.

```
$ crontab -l
```

```
No crontab for root
```

Pada contoh di atas akan di tampilkan penjadwalan yang telah dibuat dengan menggunakan crontab. Kemudian penjadwalan

tersebut dihapus dan jika coba ditampilkan kembali maka akan muncul tulisan “No crontab for root”, artinya sudah tidak ada lagi penjadwalan untuk user root.