

PERTEMUAN 1

INSTALASI UBUNTU

1. Tujuan

- Mengetahui dasar sistem operasi linux (ubuntu).
- Memahami keuntungan pemakaian sistem operasi linux (ubuntu).
- Melakukan instalasi linux (ubuntu) menggunakan vmware.

2. Dasar Teori

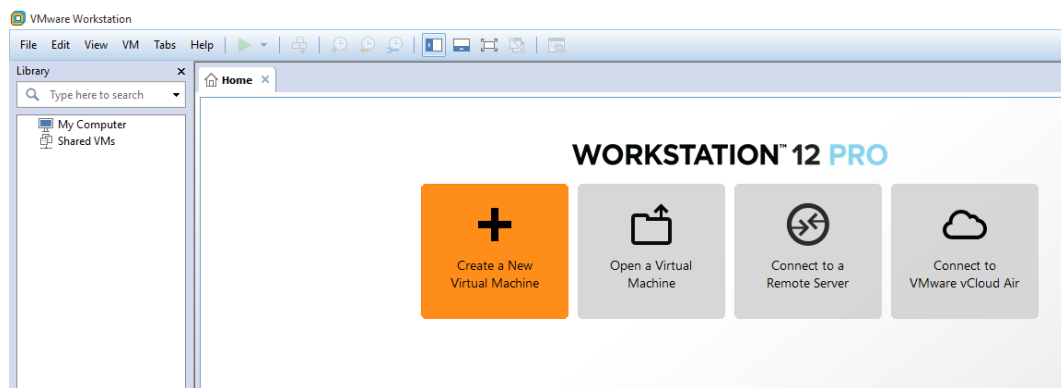
Linux adalah sebuah sistem operasi open source dan bebas (free) dibawah lisensi GNU (GNU is Not Unix) GPL (General Public Lisensi). Arti open source adalah kode sumber (source code) diikutsertakan dalam program linux sehingga dapat dilihat oleh siapa saja tanpa harus menandatangani suatu perjanjian khusus seperti NDA (Non Disclosur Agreement).

(Y.B.Mulyana, 2002:1)

Ubuntu merupakan salah satu distribusi linux yang berbasiskan debian dan didistribusikan sebagai perangkat lunak bebas .

Dalam materi ini akan dijelaskan langkah-langkah instalasi linux (ubuntu) menggunakan vmware, sistem operasi linux (ubuntu) yang dipakai adalah ubuntu 16.04 Desktop dan vmware worsktation 12 pro.

1. Persiapkan file iso ubuntu, yang dapat didownload di www.ubuntu.com.
2. Lakukan instalasi vmware terlebih dahulu jika belum melakukan instalasi.
3. Buka vmware dan pilih file, kemudian pilih new virtual mechine, atau dapat memilih menu + create a new virtual machine seperti gambar dibawah ini

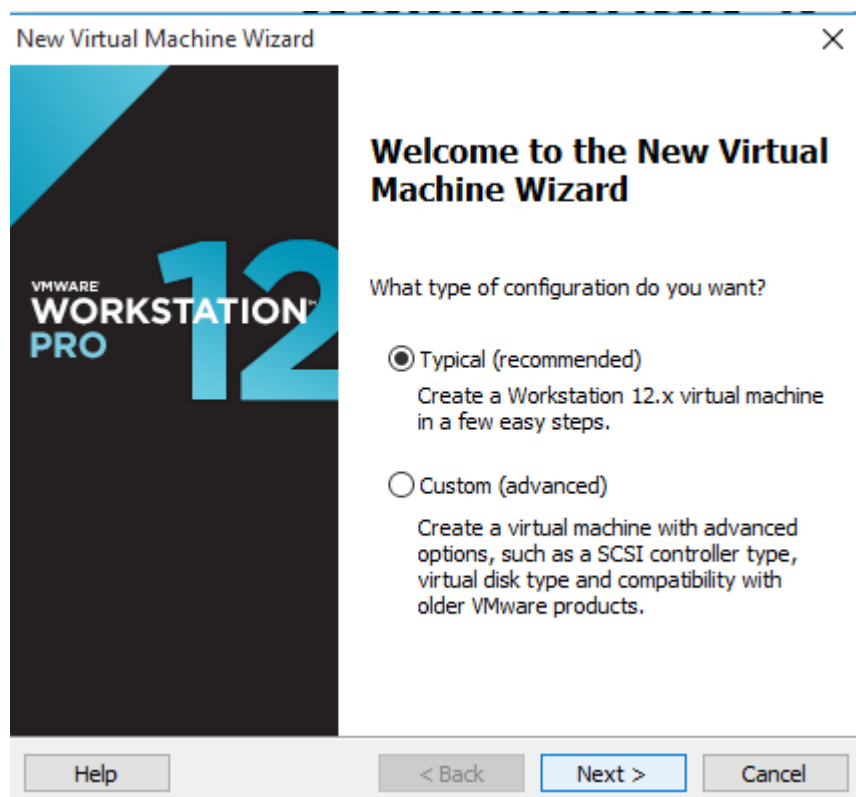


4. Pilih *Typical(Recomended)* untuk melanjutkan proses instalasi ubuntu dan membuat partisi memory pada proses indtalasi di vmware seperti gambar dibawah ini

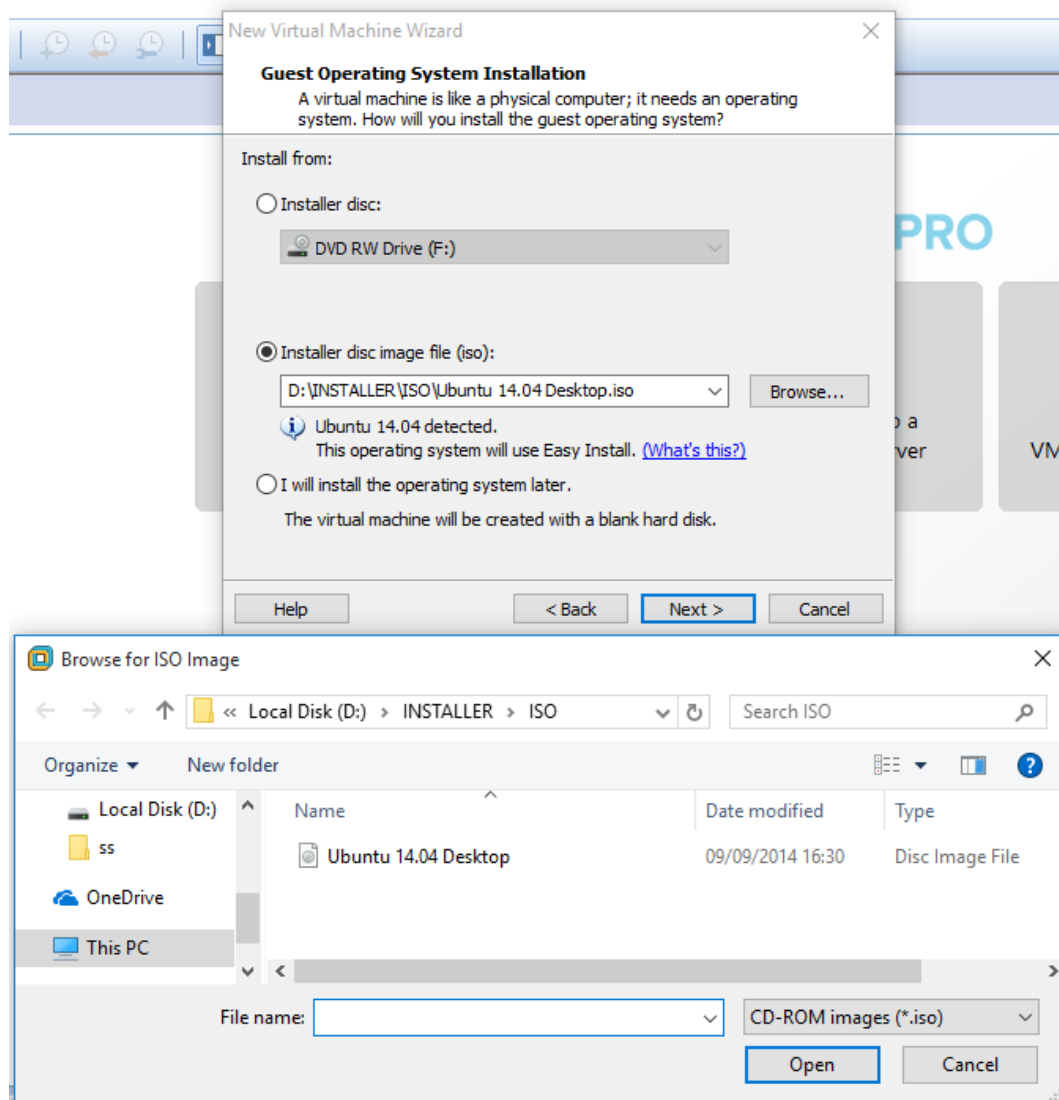
Perbedaan instalasi *Typical (Recomended)* dan *Custome (Advanced)* adalah :

- Instalasi dengan tipe *Typical (Recomended)* adalah jenis instalasi yang disarankan oleh ubuntu, karena jenis instalasi ini berisi aplikasi dan komponen ubuntu standard yang sering digunakan oleh pengguna ubuntu, dengan memilih instalasi typical tidak diwajibkan untuk mengatur partisi untuk pembagian memory.

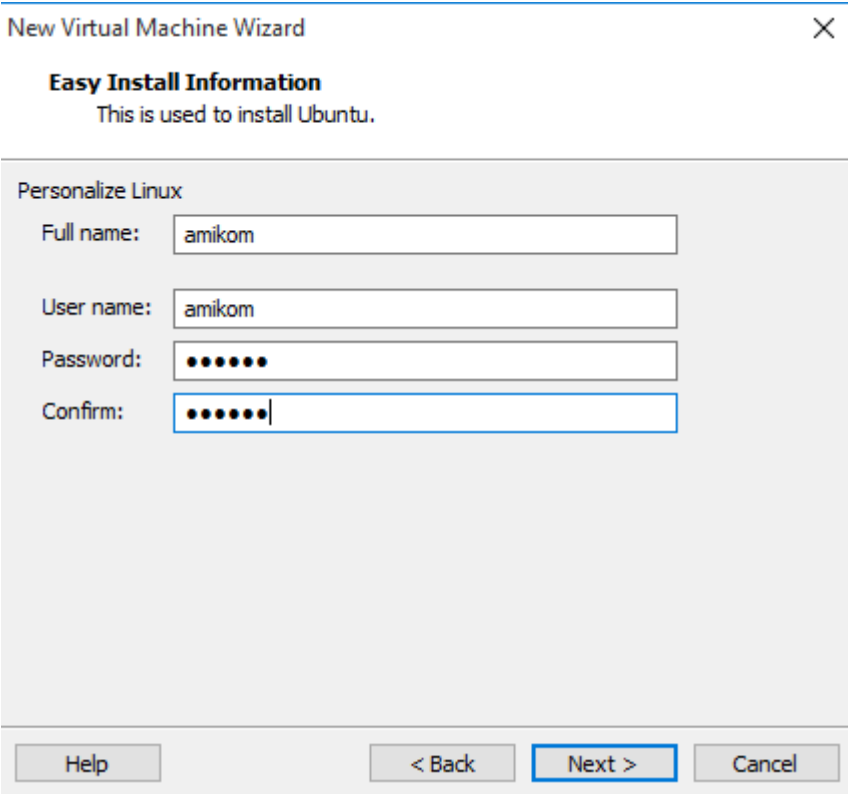
- Instalasi dengan tipe *Custom (Advanced)* mempunyai langkah berbeda, custom install disiapkan untuk instalasi dengan menghendaki pilihan yang dilakukan sesuai dengan kebutuhan pengguna, dengan memilih instalasi custom diwajibkan bagi pengguna untuk melakukan partisi atau pembagian memory.



5. Masukkan file iso ubuntu yang telah disiapkan pada menu “Guest Operating System Instalation” lalu pilih browse kemudian pilih “Next” untuk melanjutkan, seperti gambar dibawah ini.



6. Ketikkan nama, username, password untuk masuk / login pada ubuntu, kemudian pilih “Next” untuk melanjutkan, seperti gambar dibawah ini.



New Virtual Machine Wizard

Easy Install Information
This is used to install Ubuntu.

Personalize Linux

Full name: amikom

User name: amikom

Password: ●●●●●●●

Confirm: ●●●●●●●

Help < Back Next > Cancel

7. Pilih tempat penyimpanan instalasi, kemudian pilih “Next” untuk melanjutkan, seperti pada gambar dibawah ini.

New Virtual Machine Wizard

Name the Virtual Machine
What name would you like to use for this virtual machine?

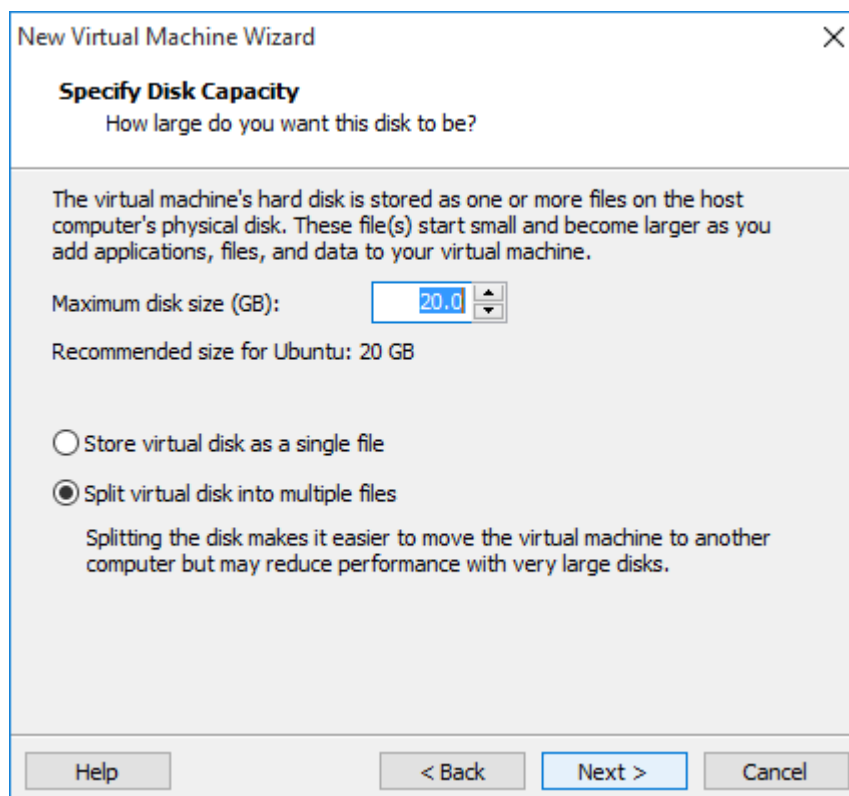
Virtual machine name:

Location:

The default location can be changed at Edit > Preferences.

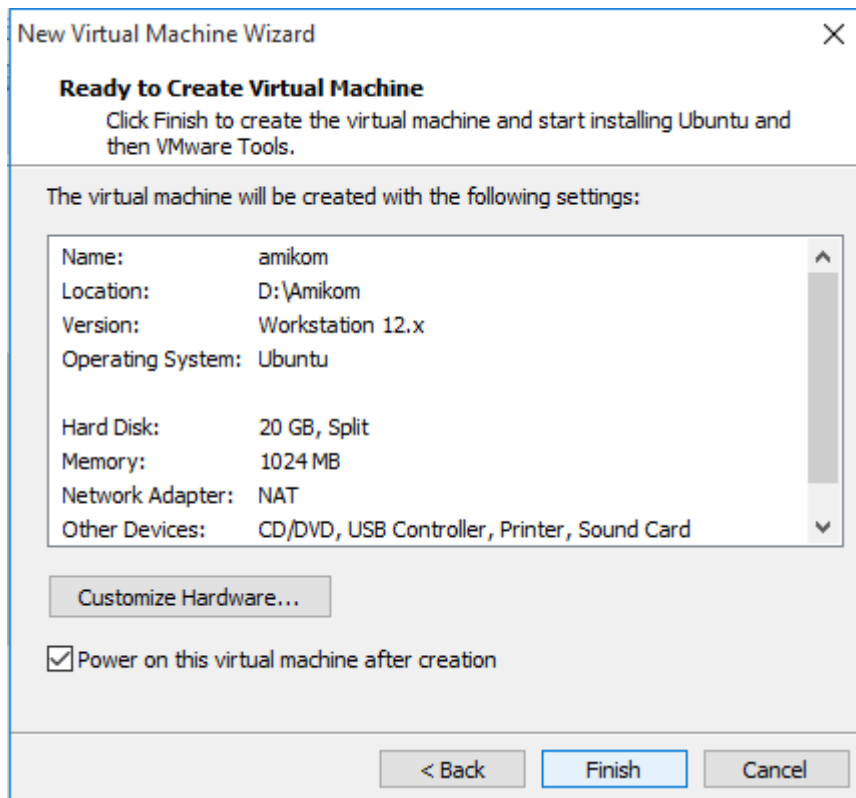
< Back Next > Cancel

8. Tentukan besar space memory untuk ubuntu yang akan diinstal, minimal adalah 20 GB, agar ubuntu dapat terinstal, kemudian pilih “Next” untuk melanjutkan seperti gambar dibawah ini.



The screenshot shows a Windows-style dialog box titled "New Virtual Machine Wizard" with a close button (X) in the top right corner. The main heading is "Specify Disk Capacity" with the subtitle "How large do you want this disk to be?". Below this, a text box explains: "The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine." There is a label "Maximum disk size (GB):" followed by a numeric input field containing "20.0" and a small up/down arrow button. Below the input field, it says "Recommended size for Ubuntu: 20 GB". There are two radio button options: "Store virtual disk as a single file" (which is unselected) and "Split virtual disk into multiple files" (which is selected). Below the selected option, a note states: "Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks." At the bottom of the dialog, there are four buttons: "Help", "< Back", "Next >" (which is highlighted with a blue border), and "Cancel".

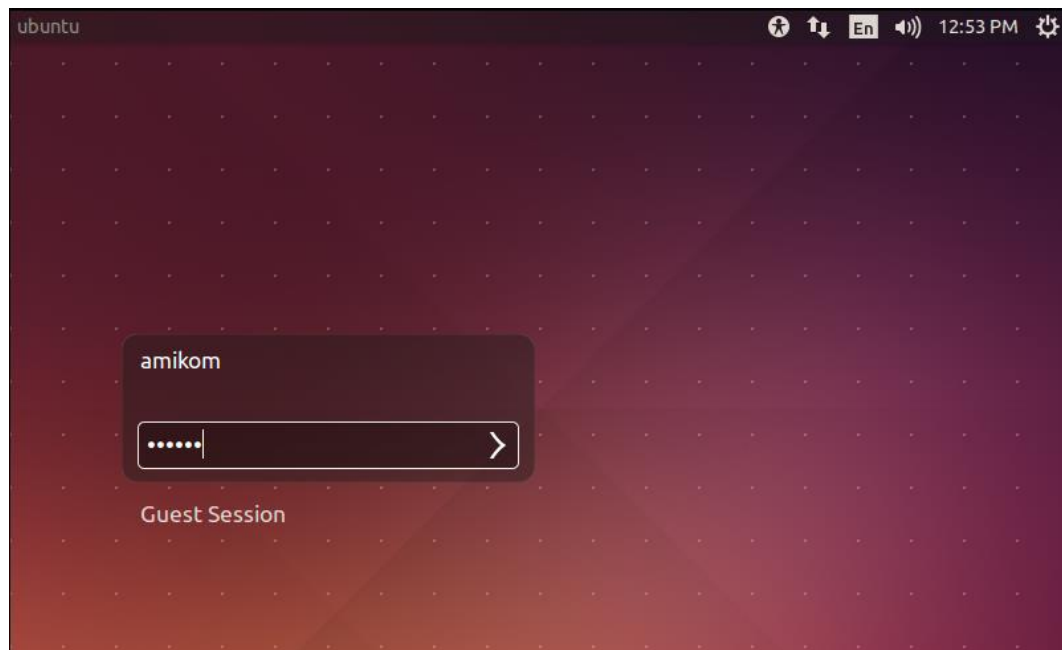
9. Pilih Finish untuk mengakhiri proses pengaturan vmware dan akan memulai proses instalasi ubuntu, seperti pada gambar dibawah ini



10. Tunggu hingga proses instalasi ubuntu selesai, seperti gambar dibawah ini.



11. Setelah proses instalasi berhasil, masukkan password untuk login, seperti pada gambar dibawah ini



PERTEMUAN 2

PERINTAH DASAR SISTEM OPERASI UBUNTU

1. Tujuan

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu

- Mengetahui perintah-perintah dasar sistem operasi Linux.
- Mengetahui instruksi pada sistem operasi Linux.
- Menggunakan perintah-perintah dasar sistem operasi Linux.

2. Dasar Teori

Command line interface adalah sebuah antarmuka berbasis teks yang memungkinkan pengguna untuk melakukan interaksi dengan sistem melalui baris-baris perintah atau yang disebut dengan Command line interface. Command line interface akan memberikan akses yang lebih untuk melakukan perubahan pada sistem yang digunakan. Linux memberi hak akses penuh untuk melakukan kontrol kepada sistem yang digunakan.

Setiap Linux mempunyai nama login (user account) yang didaftarkan pada administrator sistem. Prompt dari shell bash pada Linux menggunakan simbol / tanda \$. Sesi Linux ,terdiri dari : Login, Bekerja dengan Shell, Logout. Pada Linux bash proses Login akan mengeksekusi program /etc/profile untuk semua pemakai dan file bash_profile pada direktori awal (Home). Pada saat Logout shell bash akan mengeksekusi script bash_logout.

Format Instruksi Sistem Operasi Linux

Format instruksi standard pada sistem operasi Linux, adalah :

\$ (Nama Instruksi) [pilihan] [argumen]

Pilihan adalah option yang dimulai dengan tanda – (minus). Argumen dapat kosong satu atau beberapa argumen (parameter).

Contoh :

\$ ls = Tanpa argumen

\$ ls -a = Option adalah -a = all, tanpa argumen

\$ ls /bin = Tanpa option, argumen adalah /bin

\$ ls/bin/etc/usr = Terdapat 3 argumen

\$ ls -l /usr = 1 option dan 1 argumen = long list

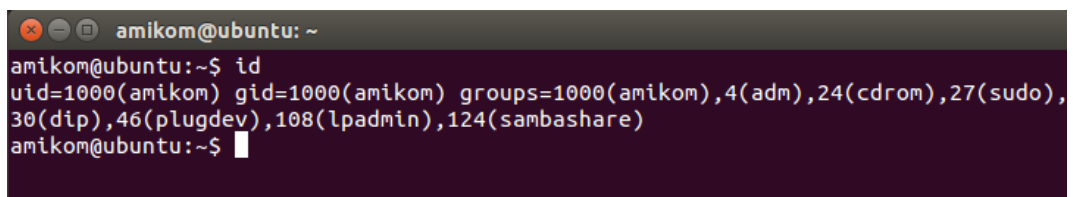
\$ ls -la /bin /etc = 2 option -l dan -a dan 2 argumen

Perintah-Perintah Dasar Linux

Berikut adalah macam-macam perintah dasar Linux :

- **\$ id**

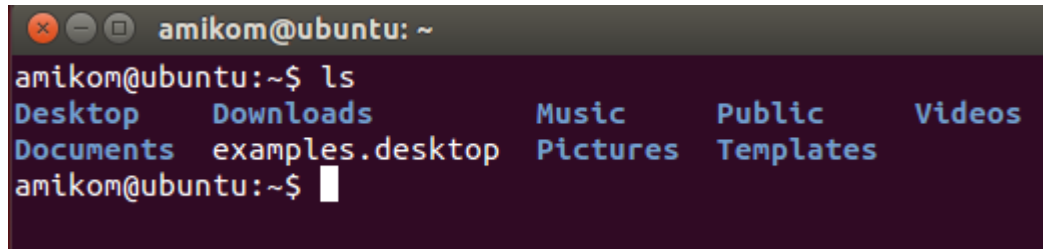
id digunakan untuk menampilkan identitas diri (nomor id dan nomor group).



```
amikom@ubuntu: ~  
amikom@ubuntu:~$ id  
uid=1000(amikom) gid=1000(amikom) groups=1000(amikom),4(adm),24(cdrom),27(sudo),  
30(dip),46(plugdev),108(lpadmin),124(sambashare)  
amikom@ubuntu:~$
```

- **\$ ls**

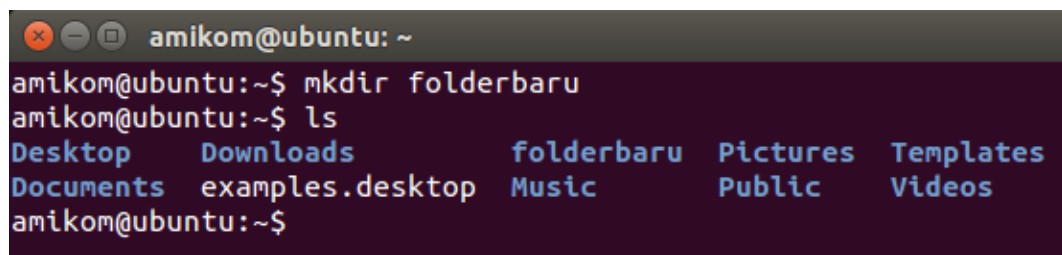
ls digunakan untuk list daftar direktori atau melihat isi dari sebuah direktori.



```
amikom@ubuntu:~$ ls
Desktop    Downloads      Music    Public    Videos
Documents  examples.desktop Pictures  Templates
amikom@ubuntu:~$
```

- **\$ mkdir**

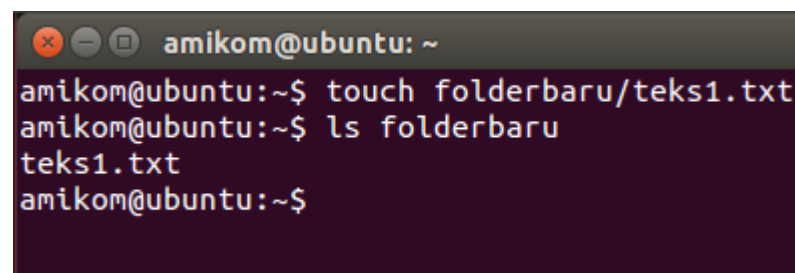
mkdir digunakan untuk membuat direktori baru.



```
amikom@ubuntu:~$ mkdir folderbaru
amikom@ubuntu:~$ ls
Desktop    Downloads      folderbaru  Pictures  Templates
Documents  examples.desktop Music       Public    Videos
amikom@ubuntu:~$
```

- **\$ touch**

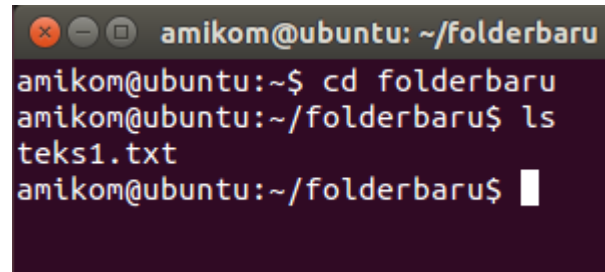
touch digunakan untuk membuat file baru.



```
amikom@ubuntu:~$ touch folderbaru/teks1.txt
amikom@ubuntu:~$ ls folderbaru
teks1.txt
amikom@ubuntu:~$
```

- **\$ cd**

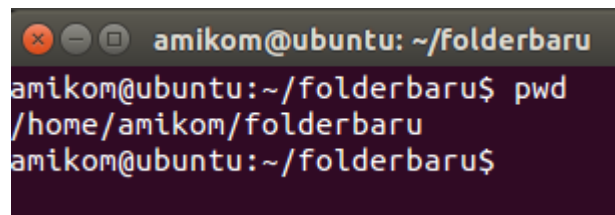
cd digunakan untuk berpindah direktori.

A terminal window titled 'amikom@ubuntu: ~/folderbaru'. The user enters 'cd folderbaru' and the prompt changes to '~/folderbaru\$'. Then, the user enters 'ls' and the output is 'teks1.txt'.

```
amikom@ubuntu: ~/folderbaru
amikom@ubuntu:~$ cd folderbaru
amikom@ubuntu:~/folderbaru$ ls
teks1.txt
amikom@ubuntu:~/folderbaru$
```

- **\$ pwd**

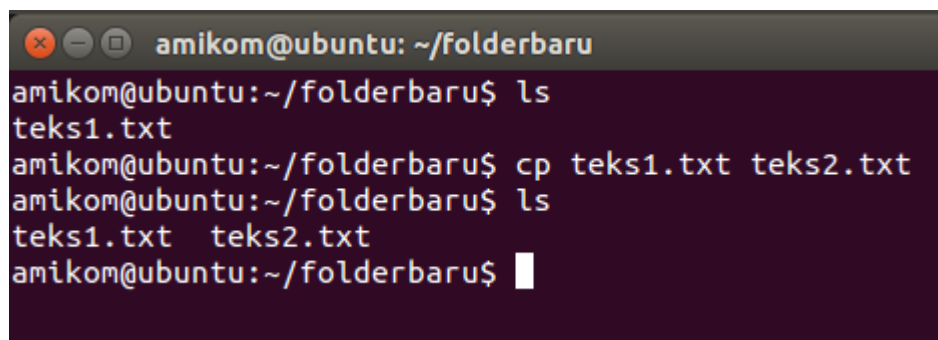
pwd digunakan untuk mencetak nama direktori yang sedang aktif.

A terminal window titled 'amikom@ubuntu: ~/folderbaru'. The user enters 'pwd' and the output is '/home/amikom/folderbaru'.

```
amikom@ubuntu:~/folderbaru$ pwd
/home/amikom/folderbaru
amikom@ubuntu:~/folderbaru$
```

- **\$ cp**

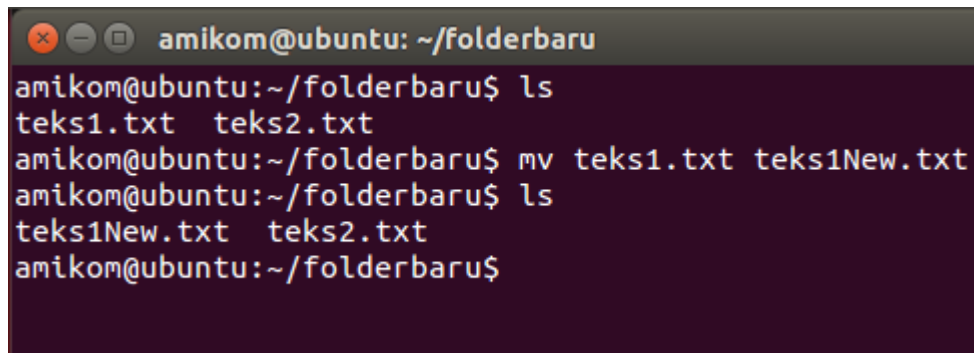
cp digunakan untuk mengcopy file atau direktori.

A terminal window titled 'amikom@ubuntu: ~/folderbaru'. The user enters 'ls' and the output is 'teks1.txt'. Then, the user enters 'cp teks1.txt teks2.txt'. Finally, the user enters 'ls' and the output is 'teks1.txt teks2.txt'.

```
amikom@ubuntu:~/folderbaru$ ls
teks1.txt
amikom@ubuntu:~/folderbaru$ cp teks1.txt teks2.txt
amikom@ubuntu:~/folderbaru$ ls
teks1.txt teks2.txt
amikom@ubuntu:~/folderbaru$
```

- **\$mv**

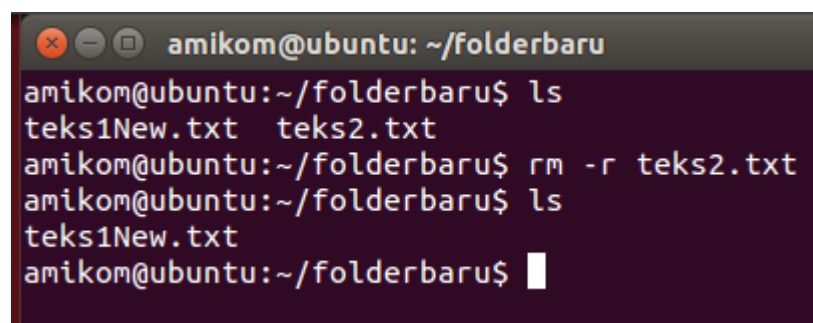
mv digunakan untuk mengganti nama suatu file.



```
amikom@ubuntu: ~/folderbaru
amikom@ubuntu:~/folderbaru$ ls
teks1.txt  teks2.txt
amikom@ubuntu:~/folderbaru$ mv teks1.txt teks1New.txt
amikom@ubuntu:~/folderbaru$ ls
teks1New.txt  teks2.txt
amikom@ubuntu:~/folderbaru$
```

- **\$ rm**

rm digunakan untuk menghapus file atau direktori.



```
amikom@ubuntu: ~/folderbaru
amikom@ubuntu:~/folderbaru$ ls
teks1New.txt  teks2.txt
amikom@ubuntu:~/folderbaru$ rm -r teks2.txt
amikom@ubuntu:~/folderbaru$ ls
teks1New.txt
amikom@ubuntu:~/folderbaru$
```

- **\$ history**

history digunakan untuk menampilkan riwayat perintah yang telah kita execute.

```

amikom@ubuntu:~/folderbaru$ history
1  ls
2  clear
3  mkdir folderbaru
4  ls
5  clear
6  cd folderbaru
7  ls
8  clear
9  cd
10 clear
11 cd folderbaru
12 clear
13 pwd

```

Percobaan :

1. Login sebagai user
2. Buka terminal dan lakukan perintah-perintah dasar linux dibawah ini :

\$ date	Menampilkan tanggal saat ini.
\$ cal 3 2003	Menampilkan kalender.
\$ cal -y	
\$ hostname	Menampilkan identitas mesin.
\$ uname	
\$ uname -a	
\$ w	Menampilkan siapa saja yang sedang aktif.
\$ who	
\$ whoami	
\$ finger	Menampilkan informasi finger.
\$ finger <user>	
\$ man ls	Menggunakan manual.
\$ man man	
\$ man -k file	
\$ man 5 passwd	
\$ clear	Menghapus layar.
\$ apropos date	

\$ apropos mail	Mencari perintah yang deskripsinya mengandung kata kunci yang dicari.
\$ apropos telnet	
\$ whatis date	Mencari perintah yang tepat sama dengan kunci yang dicari.
\$ file	Melihat tipe file.
\$ file *	
\$ file /bin/ls	
\$ cat f1	Melihat isi file.
\$ more f1	
\$ pg f1	

PERTEMUAN 3

MANAJEMEN PROSES

1. Tujuan

- Memahami konsep pros pada sistem operasi Linux.
- Menampilakn beberapa cara menampilkan hubungan proses.
- Menampilkan status proses dengan beberapa format berbeda.
- Melakukan pengontrolan proses pada shell.
- Memahami penjadwalan proses.

2. Dasar Teori

1. Konsep Proses Pada Sistem Operasi Linux

Proses adalah program yang sedang dieksekusi. Setiap kali menggunakan utilitas sistem atau program aplikasi dari shell, satu atau lebih proses “child” akan dibuat oleh shell sesuai perintah yang diberikan. Setiap kali intruksi diberikan pada Linux shell, maka karnel akan menciptakan sebuah proses-id. Proses ini disebut juga dengan teminology Unix sebagai sebuah Job. Proses Id (PID) dimulai dari 0, yaitu proses INIT, kemudian diikuti oleh proses berikutnya (terdaftar pada /etc/inittab).

Beberapa tipe proses :

- **Foreground**
proses yang diciptakan oleh pemakai langsung pada terminal (interaktif, dialog).
- **Batch**
proses yang dikumpulkan dan dijalankan secara sekuensial (satu persatu). Proses Batch tidak diasosiasikan (berinteraksi) dengan terminal.

- Daemon

Proses yang menunggu permintaan (request) dari proses lainnya dan menjalankan tugas sesuai dengan permintaan tersebut. Bila tidak ada request, maka program ini akan berada dalam kondisi “idle” dan tidak menggunakan waktu hitung CPU. Umumnya nama proses daemond di UNIX berakhiran d, misal inetd, named, popd dll.

```

370 ?      00:00:01  systemd-journal
392 ?      00:00:01  systemd-udevd
467 ?      00:00:00  systemd-timesyn
865 ?      00:00:00  acpid
869 ?      00:00:03  dbus-daemon
880 ?      00:00:00  cron
882 ?      00:00:00  rsyslogd
891 ?      00:00:00  accounts-daemon
894 ?      00:00:00  systemd-logind
899 ?      00:00:00  ModemManager
905 ?      00:00:00  whoopsie
910 ?      00:00:00  NetworkManager
1100 ?     00:00:00  dhclient
1159 ?     00:00:00  dnsmasq
926 ?      00:00:00  avahi-daemon
950 ?      00:00:00  avahi-daemon
947 ?      00:00:00  snapd
1021 ?     00:00:00  polkitd
1043 tty1   00:00:00  agetty
1073 ?      00:00:00  lightdm
1089 tty7   00:00:39  Xorg
1405 ?     00:00:00  lightdm

```

2. Sinyal

Proses dapat mengirim dan menerima sinyal dari dan ke proses lainnya. Proses mengirim sinyal melalui interaksi “kill” dengan format :

kill [- nomor sinyal] PID

Nomor sinyal : 1 sampai dengan maksimum nomor sinyal yang didefinisikan sistem standar nomor sinyal yang terpenting adalah :

No Sinyal	Nama	Deskripsi
1	SIGHUP	Hangup, sinyal dikirim bila proses terputus, misalnya melalui putusnya hubungan modem
2	SIGINT	Sinyal interrupt, melalui ^C
3	SIGQUIT	Sinyal Quit, melalui ^\
9	SIGKILL	Sinyal Kill, menghentikan proses
15	SIGTERM	Sinyal terminasi software

3. Mengirim Sinyal

Mengirim sinyal adalah satu alat komunikasi antar proses, yaitu memberitahukan proses yang sedang berjalan bahwa ada sesuatu yang harus dikendalikan. Berdasarkan sinyal yang dikirim ini maka proses dapat bereaksi dan administrator/programer dapat menentukan reaksi tersebut. Mengirim sinyal menggunakan intruksi :

kill [- nomor sinyal] PID

Sebelum mengirim sinyal PID proses yang akan dikirim harus diketahui terlebih dahulu. Utilitas untuk melakukan kontrol proses dapat ditemukan pada sistem UNIX dalam perintah killall. Perintah ini akan menghentikan proses sesuai PID.

4. Mengontrol Proses Pada Shell

Shell menyediakan fasilitas job control yang memungkinkan mengontrol beberapa job atau proses yang sedang berjalan pada waktu yang sama. Misalnya bila melakukan pengeditan file teks dan ingin melakukan interrupt pengeditan untuk mengerjakan hal lainnya. Bila selesai, dapat kembali (switch) ke editor dan melakukan pengeditan file teks kembali.

Job pada foreground kemungkinan dihentikan sementara (suspend), dengan menekan [Ctrl + Z]. Job yang diberhentikan sementara dapat

dijalankan kembali pada foreground atau background sesuai keperluan dengan menekan “fg” atau “bg”.Menghentikan job sementara sangat berbeda dengan melakukan interrupt job, dimana job yang diinterrupt akan dimatikan secara permanen dan tidak dapat dijalankan lagi.

5. Mengontrol Proses Lain

Perintah ps dapat digunakan untuk semua proses yang sedang berjalan pada mesin (bukan hanya pada shell saat ini) dengan format :

ps -fae atau

ps -aux

Beberapa versi UNIX mempunyai utilitas sistem yang disebut top yang menyediakan cara interaktif untuk memonitor aktifitas sistem.Statistik secara detail dengan proses yang berjalan ditampilkan dan secara terus menerus di refresh.Proses ditampilkan secara terurut dari utilitas CPU.Kunci yang berguna pada top adalah :

s – set update frequency

u – display proses dari satu user

k – kill proses (dengan PID)

q – quit

Percobaan :

1. Login sebagai user.
2. Lakukan percobaan – percobaan dibawah ini kemudian analisa hasil percobaan.
3. Selesaikan soal – soal latihan.

Percobaan 1 : Status Proses

1. Pindah ke command line terminal (tty2) dengan menekan Crl + Alt + F2 dan login ke terminal sebagai user.
2. Interuksi ps (proses status) digunakan untuk melihat kondisi proses yang sedang berjalan.

\$ ps

```
amikom@ubuntu:~$ ps
  PID TTY          TIME CMD
 2811 tty2        00:00:00 bash
 2825 tty2        00:00:00 ps
amikom@ubuntu:~$
```

3. Untuk melihat elemen lainnya ,gunakan option -u (user).Akan tampil %CPU adalah presentasi CPU time yang digunakan pada proses tersebut.%MEM adalah presentasi sistem memori yang digunakan pada proses.SIZE adalah jumlah memori yang digunakan.START adalah kapan proses tersebut aktif.

\$ ps -u

```
amikom@ubuntu:~$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
amikom    2427  0.0  0.5 29492 4988 pts/1    Ss+  03:52   0:01 bash
amikom    2532  0.0  0.1 14516 1756 pts/1    T   04:08   0:00 cat filebaru.txt - surat
amikom    2811  0.2  0.5 29516 4984 tty2     S   04:43   0:00 -bash
amikom    2837  0.0  0.3 44432 3284 tty2     R+   04:45   0:00 ps -u
amikom@ubuntu:~$
```

4. Melihat proses yang sedang berjalan pada user.

\$ ps -u <user>

```
      PID TTY          TIME CMD
    1671 ?            00:00:00 systemd
    1673 ?            00:00:00 (sd-pam)
    1685 ?            00:00:00 gnome-keyring-d
    1687 ?            00:00:01 upstart
    1769 ?            00:00:00 upstart-udev-br
    1775 ?            00:00:02 dbus-daemon
    1787 ?            00:00:00 window-stack-br
    1818 ?            00:00:00 upstart-dbus-br
    1824 ?            00:00:00 upstart-dbus-br
    1831 ?            00:00:00 upstart-file-br
    1840 ?            00:00:05 bamfdaemon
    1842 ?            00:00:11 ibus-daemon
    1844 ?            00:00:00 gpg-agent
    1850 ?            00:00:00 gvfsd
    1857 ?            00:00:00 gvfsd-fuse
    1864 ?            00:00:00 ibus-dconf
    1870 ?            00:00:02 ibus-ui-gtk3
    1878 ?            00:00:00 at-spi-bus-laun
    1879 ?            00:00:00 ibus-x11
```

5. Melihat proses opsi -a (all) atau -au (all user)

\$ ps -a

\$ ps -au

```
amikom@ubuntu:~$ ps -a
PID TTY          TIME CMD
2532 pts/1      00:00:00 cat
2811 tty2       00:00:00 bash
2844 tty2       00:00:00 ps
amikom@ubuntu:~$ ps -au
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1027  0.0  0.1 23008 1612 tty1      Ss+   03:47   0:00 /sbin/agetty --noclear tty1 linux
root        1063  4.3  3.1 301112 31624 tty1      Ss+   03:48   2:33 /usr/lib/xorg/Xorg -core :0 -seat
amikom      2427  0.0  0.5 29492 4988 pts/1    Ss+   03:52   0:01 bash
amikom      2532  0.0  0.1 14516 1756 pts/1    T     04:08   0:00 cat filebaru.txt - surat
root        2757  0.0  0.3 67324 3524 tty2      Ss    04:43   0:00 /bin/login --
amikom      2811  0.1  0.5 29516 4988 tty2      S     04:43   0:00 -bash
amikom      2845  0.0  0.3 44432 3376 tty2      R+    04:46   0:00 ps -au
amikom@ubuntu:~$
```

Percobaan 2 : Menampilkan Hubungan Proses Parent dan Child

1. Ketikan ps -eH.Opsi e untuk melihat semua proses dan opsi H untuk menampilkan proses secara hierarki.Prose child muncul dibawah proses parent.Proses child ditandai dengan awalan beberapa spasi.

\$ ps -eH

```
evolution-daemon
gvfs-gphoto2-volume
gvfs-mtp-volume      parent
gvfs-goa-volume
gvfs-afc-volume
sh
zeitgeist-daemon     child
zeitgeist-rtss
gconfd-2
gvfsd-trash
unity-scope-hom
unity-scope-loa
unity-files-dae
```

2. Perintah ps -e f untuk menampilkan hal serupa dengan langkah 1 hanya berbeda ps -e f dengan karakter grafis dalam menampilkannya.

\$ ps -e f

```
UID        PID     PPID  C STIME TTY          TIME CMD
root         1         0  1 20:02 ?        00:00:08 /sbin/init auto noprompt
root         2         0  0 20:02 ?        00:00:00 [kthreadd]
root         3         0  0 20:02 ?        00:00:01 [ksoftirqd/0]
root         5         2  0 20:02 ?        00:00:00 [kworker/0:0H]
root         7         2  0 20:02 ?        00:00:05 [rcu_sched]
root         8         2  0 20:02 ?        00:00:00 [rcu_bh]
root         9         2  0 20:02 ?        00:00:00 [migration/0]
root        10         2  0 20:02 ?        00:00:00 [lru-add-drain]
root        11         2  0 20:02 ?        00:00:00 [watchdog/0]
root        12         2  0 20:02 ?        00:00:00 [cpuhp/0]
root        13         2  0 20:02 ?        00:00:00 [kdevtmpfs]
root        14         2  0 20:02 ?        00:00:00 [netns]
root        15         2  0 20:02 ?        00:00:00 [khungtaskd]
root        16         2  0 20:02 ?        00:00:00 [oom_reaper]
root        17         2  0 20:02 ?        00:00:00 [writeback]
root        18         2  0 20:02 ?        00:00:00 [kcompactd0]
root        19         2  0 20:02 ?        00:00:00 [ksmd]
root        20         2  0 20:02 ?        00:00:00 [khugepaged]
root        21         2  0 20:02 ?        00:00:00 [crypto]
root        22         2  0 20:02 ?        00:00:00 [integrityd]
root        23         2  0 20:02 ?        00:00:00 [hisset]
root        24         2  0 20:02 ?        00:00:00 [kblockd]
root        25         2  0 20:02 ?        00:00:00 [ata_sff]
root        26         2  0 20:02 ?        00:00:00 [mail]
root        27         2  0 20:02 ?        00:00:00 [devfreq_wq]
root        28         2  0 20:02 ?        00:00:00 [watchdogd]
root        30         2  0 20:02 ?        00:00:00 [kworker/0:1]
root        31         2  0 20:02 ?        00:00:07 [kswapd0]
root        32         2  0 20:02 ?        00:00:00 [vmstat]
root        33         2  0 20:02 ?        00:00:00 [ecryptfs-kthrea]
root        72         2  0 20:02 ?        00:00:00 [kthrotld]
root        73         2  0 20:02 ?        00:00:00 [acpi_thermal_pm]
root        74         2  0 20:02 ?        00:00:00 [bioset]
root        75         2  0 20:02 ?        00:00:00 [bioset]
root        76         2  0 20:02 ?        00:00:00 [bioset]
--More--
```

3. Menampilkan proses yang ada di sistem dalam hirarki

\$ pstree

```
systemd--+-ManagementAgent---6*[{ManagementAgent}]
          |
          |--ModemManager--+-{gdbus}
          |                  |--{gmain}
          |
          |--NetworkManager--+-dhclient
          |                    |--dnsmasq
          |                    |--{gdbus}
          |                    |--{gmain}
          |
          |--VGAuthService
          |
          |--accounts-daemon--+-{gdbus}
          |                   |--{gmain}
          |
          |--acpid
          |
          |--agetty
          |
          |--avahi-daemon---avahi-daemon
          |
          |--colord--+-{gdbus}
          |           |--{gmain}
          |
          |--cron
          |
          |--cups-browsed--+-{gdbus}
          |                |--{gmain}
          |
          |--dbus-daemon
          |
          |--fwupd--+-3*[{GUsbEventThread}]
          |          |--{fwupd}
          |          |--{gdbus}
          |          |--{gmain}
          |
          |--gnome-keyring-d--+-{gdbus}
          |                   |--{gmain}
          |                   |--{timer}
          |
          |--lightdm--+-Xorg
          |             |
          |             |--lightdm--+-upstart--+-at-spi-bus-laun--+-dbus-daemon
          |             |               |               |               |
          |             |               |               |               |--{dconf worker}
          |             |               |               |               |--{gdbus}
          |             |               |               |               |--{gmain}
          |             |               |               |
          |             |               |               |--at-spi2-registr--+-{gdbus}
          |             |               |               |               |--{gmain}
```

4. Ketikkan `pstree | grep mingetty`, akan menampilkan semua proses `mingetty` yang berjalan pada sistem yang berupa console virtual. Selain menampilkan semua proses, proses dikelompokkan dalam satu baris dengan suatu angka sebagai jumlah proses yang berjalan.

\$ `pstree | grep mingetty`

- \$ pstree -p

9)

- \$ pstree -h

--More--

Percobaan 3 : Menampilkan status Proses Dengan Berbagai Format

1. Ketikkan `ps -e | more`. Opsi `-e` untuk menampilkan semua proses dalam bentuk 4 kolom yang terdiri dari : PID, TTY, TIME, CMD.

\$ `ps -e | more`

Jika halaman penuh terlihat prompt `--More--` dibagian bawah screen,

```

      PID TTY          TIME CMD
      1 ?            00:00:08 systemd
      2 ?            00:00:00 kthreadd
      3 ?            00:00:01 ksoftirqd/0
      4 ?            00:00:00 kworker/0:0
      5 ?            00:00:00 kworker/0:0H
      6 ?            00:00:00 kworker/u256:0
      7 ?            00:00:04 rcu_sched
      8 ?            00:00:00 rcu_bh
      9 ?            00:00:00 migration/0
     10 ?            00:00:00 lru-add-drain
     11 ?            00:00:00 watchdog/0
     12 ?            00:00:00 cpuhp/0
     13 ?            00:00:00 kdevtmpfs
     14 ?            00:00:00 netns
     15 ?            00:00:00 khungtaskd
     16 ?            00:00:00 oom_reaper
     17 ?            00:00:00 writeback
     18 ?            00:00:00 kcompactd0
     19 ?            00:00:00 ksmd
     20 ?            00:00:00 khugepaged
     21 ?            00:00:00 crypto
     22 ?            00:00:00 kintegrityd
     23 ?            00:00:00 bioset
     24 ?            00:00:00 kblockd
     25 ?            00:00:00 ata_sff
     26 ?            00:00:00 md
     27 ?            00:00:00 devfreq_wq
     28 ?            00:00:00 watchdogd
     30 ?            00:00:00 kworker/0:1
     31 ?            00:00:06 kswapd0
     32 ?            00:00:00 vmstat
     33 ?            00:00:00 ecryptfs-kthrea
     72 ?            00:00:00 kthrotld
     73 ?            00:00:00 acpi_thermal_pm
     74 ?            00:00:00 bioset
--More--
```

2. Ketikkan `ps ax | more`. Opsi `a` untuk menampilkan semua proses yang dihasilkan terminal (TTY). Opsi `x` untuk menampilkan semua proses yang tidak dihasilkan terminal. Secara logika opsi ini sama dengan opsi `-e`. Terdapat 5 kolom yaitu : PID, TTY, STAT, TIME dan COMMAND.

PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:08	/sbin/init auto noprompt
2	?	S	0:00	[kthreadd]
3	?	S	0:01	[ksoftirqd/0]
5	?	S<	0:00	[kworker/0:0H]
6	?	S	0:00	[kworker/u256:0]
7	?	S	0:05	[rcu_sched]
8	?	S	0:00	[rcu_bh]
9	?	S	0:00	[migration/0]
10	?	S<	0:00	[lru-add-drain]
11	?	S	0:00	[watchdog/0]
12	?	S	0:00	[cpuhp/0]
13	?	S	0:00	[kdevtmpfs]
14	?	S<	0:00	[netns]
15	?	S	0:00	[khungtaskd]
16	?	S	0:00	[oom_reaper]
17	?	S<	0:00	[writeback]
18	?	S	0:00	[kcompactd0]
19	?	SN	0:00	[ksmd]
20	?	SN	0:00	[khugepaged]
21	?	S<	0:00	[crypto]
22	?	S<	0:00	[kintegrityd]
23	?	S<	0:00	[bioaset]
24	?	S<	0:00	[kblockd]
25	?	S<	0:00	[ata_sff]
26	?	S<	0:00	[md]
27	?	S<	0:00	[devfreq_wq]
28	?	S<	0:00	[watchdogd]
30	?	S	0:00	[kworker/0:1]
31	?	S	0:07	[kswapd0]
32	?	S<	0:00	[vmstat]
33	?	S	0:00	[ecryptfs-kthrea]
72	?	S<	0:00	[kthrotld]
73	?	S<	0:00	[acpi_thermal_pm]
74	?	S<	0:00	[bioaset]
75	?	S<	0:00	[bioaset]

3. Ketikkan `ps -ef | more`. Opsi `-ef` untuk menampilkan semua proses dalam format daftar penuh.

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	1	20:02	?	00:00:08	/sbin/init auto noprompt
root	2	0	0	20:02	?	00:00:00	[kthreadd]
root	3	0	0	20:02	?	00:00:01	[ksoftirqd/0]
root	5	2	0	20:02	?	00:00:00	[kworker/0:0H]
root	7	2	0	20:02	?	00:00:05	[rcu_sched]
root	8	2	0	20:02	?	00:00:00	[rcu_bh]
root	9	2	0	20:02	?	00:00:00	[migration/0]
root	10	2	0	20:02	?	00:00:00	[lru-add-drain]
root	11	2	0	20:02	?	00:00:00	[watchdog/0]
root	12	2	0	20:02	?	00:00:00	[cpuhp/0]
root	13	2	0	20:02	?	00:00:00	[kdevtmpfs]
root	14	2	0	20:02	?	00:00:00	[netns]
root	15	2	0	20:02	?	00:00:00	[khungtaskd]
root	16	2	0	20:02	?	00:00:00	[oom_reaper]
root	17	2	0	20:02	?	00:00:00	[writeback]
root	18	2	0	20:02	?	00:00:00	[kcompactd0]
root	19	2	0	20:02	?	00:00:00	[ksmd]
root	20	2	0	20:02	?	00:00:00	[khugepaged]
root	21	2	0	20:02	?	00:00:00	[crypto]
root	22	2	0	20:02	?	00:00:00	[kintegrityd]
root	23	2	0	20:02	?	00:00:00	[bioaset]
root	24	2	0	20:02	?	00:00:00	[kblockd]
root	25	2	0	20:02	?	00:00:00	[ata_sff]
root	26	2	0	20:02	?	00:00:00	[md]
root	27	2	0	20:02	?	00:00:00	[devfreq_wq]
root	28	2	0	20:02	?	00:00:00	[watchdogd]
root	30	2	0	20:02	?	00:00:00	[kworker/0:1]
root	31	2	0	20:02	?	00:00:07	[kswapd0]
root	32	2	0	20:02	?	00:00:00	[vmstat]
root	33	2	0	20:02	?	00:00:00	[ecryptfs-kthrea]
root	72	2	0	20:02	?	00:00:00	[kthrotld]
root	73	2	0	20:02	?	00:00:00	[acpi_thermal_pm]
root	74	2	0	20:02	?	00:00:00	[bioaset]
root	75	2	0	20:02	?	00:00:00	[bioaset]
root	76	2	0	20:02	?	00:00:00	[bioaset]

Jika halaman penuh terlihat prompt `--More--` dibagian bawah screen, tekan `q` untuk kembali ke prompt perintah.

4. Ketikkan `ps -eo pid,cmd | more`. Opsei `-eo` untuk menampilkan semua proses dalam format sesuai definisi user, yaitu terdiri dari kolom PID dan CMD.

```
PID CMD
1 /sbin/init auto noprompt
2 [kthreadd]
3 [ksoftirqd/0]
5 [kworker/0:0H]
7 [rcu_sched]
8 [rcu_bh]
9 [migration/0]
10 [lru-add-drain]
11 [watchdog/0]
12 [cpuhp/0]
13 [kdevtmpfs]
14 [netns]
15 [khungtaskd]
16 [oom_reaper]
17 [writeback]
18 [kcompactd0]
19 [ksmd]
20 [khugepaged]
21 [crypto]
22 [kintegrityd]
23 [bioset]
24 [kblockd]
25 [ata_sff]
26 [md]
27 [devfreq_wq]
28 [watchdogd]
30 [kworker/0:1]
31 [kswapd0]
32 [vmstat]
33 [ecryptfs-kthrea]
72 [kthrotld]
73 [acpi_thermal_pm]
74 [bioset]
75 [bioset]
76 [bioset]
--More--
```

Jika halaman penuh terlihat prompt `--More--` dibagian bawah screen, tekan `q` untuk kembali ke prompt perintah.

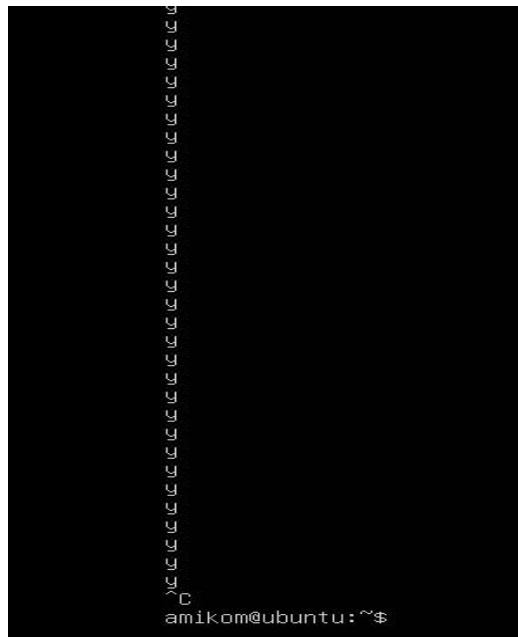
5. Ketikkan `ps -eo pid,ppid,%mem,cmd | more`. Akan menampilkan kolom PID, PPID, dan %MEM. PPID adalah proses ID dari proses parent, %MEM untuk menampilkan presentasi memori sistem yang digunakan pada proses. Jika proses hanya menggunakan sedikit memori sistem akan ditampilkan 0.

Percobaan 4 : Mengontrol Proses Pada Shell

1. Gunakan perintah `yes` yang mengirim output y yang tidak pernah berhenti.

`$ yes`

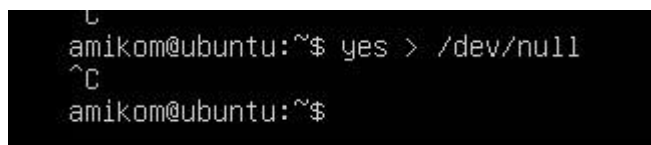
Untuk menghentikannya gunakan `Ctrl + C`.

A terminal window with a black background. A vertical column of 'y' characters is visible on the left side of the screen. At the bottom, the prompt 'amikom@ubuntu:~\$' is shown, with a '^C' character above it indicating the command was interrupted.

```
amikom@ubuntu:~$
```

2. Belokkan standar output ke `/dev/null`.

`$ yes > /dev/null`

A terminal window with a black background. The prompt 'amikom@ubuntu:~\$' is shown, followed by the command 'yes > /dev/null'. A '^C' character is shown above the prompt, indicating the command was interrupted.

```
amikom@ubuntu:~$ yes > /dev/null  
^C  
amikom@ubuntu:~$
```

Untuk menghentikannya gunakan `Ctrl + C`.

3. Salah satu perintah `yes` tetap dijalankan tetapi shell tetap digunakan untuk hal yang lain dengan meletakkan proses pada background dengan menambahkan karakter `&` pada akhir perintah.

`$ yes > /dev/null &`

Angka dalam “[] ” merupakan job number diikuti PID.

```
amikom@ubuntu:~$ yes > /dev/null&
[1] 2896
amikom@ubuntu:~$ _
```

4. Untuk melihat status proses menggunakan perintah `jobs`.

`$ jobs`

```
amikom@ubuntu:~$ jobs
[1]+  Running                  yes > /dev/null &
amikom@ubuntu:~$
```

5. Untuk menghentikan job menggunakan perintah `kill` diikuti job number atau PID proses. Untuk identifikasi job number, diikuti prefix dengan karakter “%”.

`$ kill %<nomor job>` contoh : `kill %1`

```
amikom@ubuntu:~$ kill %1
amikom@ubuntu:~$
```

6. Setelah dikill lihat status job setelah diterminasi.

`$ jobs`

Percobaan 5 : Menghentikan Dan Memulai Kembali Job

1. Cara lain meletakkan job pada background dengan memulai job secara normal (pada foreground), stop job dan memulai lagi pada background.

```
$ yes > /dev/null
```

Hentikan sementara job (suspend), bukan menghentikannya (terminate), tetapi menghentikan sementara job sampai restart. Untuk menghentikan sementara job gunakan Ctrl + Z.

```
amikom@ubuntu:~$ yes > /dev/null
^Z[1] Terminated                  yes > /dev/null

[2]+ Stopped                      yes > /dev/null
amikom@ubuntu:~$
```

2. Untuk restart job pada foreground menggunakan perintah fg.

```
$ fg
```

```
amikom@ubuntu:~$ fg
yes > /dev/null
```

3. Shell akan menampilkan nama perintah yang diletakkan di foreground. Stop job lagi dengan Ctrl + Z, kemudian gunakan perintah bg untuk meletakkan job pada background.

```
$ bg
```

```
amikom@ubuntu:~$ fg
yes > /dev/null
^Z
[2]+ Stopped                      yes > /dev/null
amikom@ubuntu:~$ bg
[2]+ yes > /dev/null &
amikom@ubuntu:~$ _
```

Job tidak bisa dihentikan dengan Ctrl + Z karena job berada pada background. Untuk menghentikannya letakkan job pada foreground dengan fg dan kemudian hentikan sementara dengan Ctrl + Z.

```
$ fg
```

4. Job pada background dapat digunakan untuk menampilkan teks pada terminal, dimana dapat diabaikan jika mencoba mengerjakan job lain.

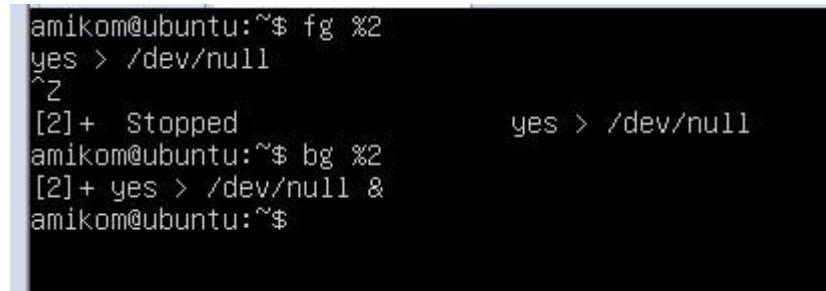
`$ yes &`

Untuk menghentikannya tidak dapat menggunakan Ctrl + C. Job harus dipindah ke foreground baru dihentikan dengan cara tekan fg dan tekan Enter, kemudian dilanjutkan dengan Ctrl + Z untuk menghentikan sementara.

5. Apabila ingin menjalankan banyak job dalam satu waktu, letakkan job pada foreground atau background dengan memberikan job ID.

`$ fg %2` atau `$ %2`

`$bg %2`



```
amikom@ubuntu:~$ fg %2
yes > /dev/null
^Z
[2]+  Stopped                  yes > /dev/null
amikom@ubuntu:~$ bg %2
[2]+ yes > /dev/null &
amikom@ubuntu:~$
```

6. Tekan fg dan Enter, kemudian dilanjutkan dengan Ctrl + Z untuk menghentikan sementara.
7. Lihat job dengan perintah `ps -fae`. Kemudian hentikan dengan perintah `kill`.

`$ ps -fae`

`$ kill -9 <nomor PID>`

8. Untuk Logout kembali ke mode grafis dengan tekan Alt + F7.

Percobaan 6 : Percobaan Dengan Penjadwalan Prioritas

1. Login sebagai root.
2. Buka 3 terminal, tampilan pada screen yang sama.
3. Pada setiap terminal ketik PS1 – “\W:” diikuti Enter. \w untuk menampilkan path pada direktori home.

4. Karena login sebagai root, maka akan ditampilkan ~: pada setiap terminal. Untuk setiap terminal ketik pwd dan Enter untuk melihat bahwa anda sedang pada direktori /root.
5. Buka terminal lagi (keempat) atur posisi sehingga terminal terlihat pada screen.
6. Pada terminal keempat, ketikkan top dan lalu tekan Enter. Maka program top akan muncul. Ketik i. Top akan menampilkan proses yang aktif. Ketik Imt. Top tidak lagi menampilkan informasi pada bagian atas screen. Pada percobaan ini, terminal ke empat sebagai jendela Top.
7. Pada terminal 1 bukalah program executable C++ dengan mengetik program yes dan tekan Enter.
8. Ulangi langkah 7 pada terminal 2.
9. Jendela Top akan menampilkan dua program yes sebagai proses yang berjalan. Nilai %CPU sama pada keduanya. Hal ini berarti kedua proses mengkonsumsi waktu proses yang sama dan berjalan cepat. PID dari kedua proses akan berbeda, misal 3148 dan 3149. Kemudian gunakan terminal 3 (yang tidak menjalankan primes maupun jendela Top) dan ketik renice 19 <PID terminal1> (contoh : renice 19 3148) lalu diikuti Enter. Hal ini berarti menggantikan penjadwalan prioritas pada proses ke 19.
10. Tunggu beberapa saat sampai program top berubah dan terlihat pada jendela Top. Pada kolom STAT memperlihatkan N untuk proses 3148. Hal ini berarti bahwa penjadwalan prioritas untuk proses 3148 lebih besar (lebih lambat) dari 0. Proses 3149 berjalan lebih cepat.
11. Program top juga mempunyai fungsi yang sama dengan renice. Pilih jendela Top dan tekan r. Program top terdapat prompt PID to renice : tekan 3148 (ingat bahwa anda harus mengganti 3148 dengan PID anda sendiri) lalu tekan Enter. Program Top memberikan prompt Renice PID 3148 to val: tekan 1-9 lalu tekan Enter.

12. Tunggu beberapa saat sampai Top berubah dan lihat nilai %CPU pada kedua proses. Sekarang proses 3148 lebih cepat dari proses 3149. Kolom status menunjukkan < pada proses 3148 yang menunjukkan penjadwalan prioritas lebih rendah (lebih cepat) dari nilai 0.
13. Pilih terminal 3 (yang sedang tidak menjalankan yes atau program top) dan ketik `nice -n -10` lalu tekan enter. Tunggu beberapa saat agar program Top berubah dan terlihat proses `primes` ketiga. Misalnya PID nya 4107. Opsi -10 berada pada kolom NI (penjadwalan prioritas).
14. Jangan menggunakan mouse dan keyboard selama 10 detik. Program Top menampilkan proses yang aktif selama proses `yes`. Maka akan terlihat proses Top terdaftar tetapi %CPU kecil (dibawah 10) dan konsisten. Juga terlihat proses berhubungan dengan desktop grafis seperti X, panel dll.
15. Pindahkan mouse sehingga kursor berubah pada screen dan lihat apa terjadi dengan tampilan Top. Proses tambahan akan muncul dan nilai bahwa proses 4107 berjalan pada penjadwalan prioritas tinggi. Pilih jendela Top, ketik `r`. PIO to renice : muncul prompt. Ketik 4107 (ubahlah 4107 dengan PID anda) lalu tekan Enter. Sekarang pindahkan mouse ke sekeliling screen. Lihat perubahannya.
16. Tutup semua terminal window.
17. Logout dan login kembali sebagai user.

Latihan :

1. Masukk ke tty2 dengan `Ctrl + Alt + F2` .Ketikkan `ps -au` dan tekan Enter. Kemudian perhatikan keluaran sebagai berikut :
 - a. Sebutkan nama-nama proses yang bukan root.
 - b. Tulis PID dan COMMAND dari proses yang paling banyak menggunakan CPU time
 - c. Sebutkan buyut proses dari PID dari proses tersebut.
 - d. Sebutkan beberapa proses daemon.

e. Pada prompt login lakukan hal-hal sebagai berikut :

\$ csh	\$ ls
\$ who	\$ sh
\$ bash	\$ ps

f. Sebutkan PID yang paling besar dan kemudian buat urutan-urutan proses sampai ke PPID = 1.

2. Cobalah format tampilan ps dengan opsi berikut dan perhatikan hasil tampilannya :

- f = daftar penuh	s = format sinyal
-j = format job	v = format virtual memori
j = format jobkontrol	X = format register i386
l = daftar memanjang	

3. Lakukan urutan pekerjaan berikut :

- Gunakan perintah find ke seluruh direktory pada sistem. Belokkan output sehingga daftar direktori dialihkan ke file directories.txt dan daftar pesan eror dialihkan ke file error.txt
- Gunakan perintah sleep 5. Apa yang terjadi dengan perintah ini ?
- Jalankan sleep 15 pada foreground, hentikan sementara dengan Ctrl + Z dan kemudian letakkan pada background dengan bg. Ketikkan jobs. Ketikkan ps. Kembalikan pada foreground dengan perintah fg.
- Jalankan sleep 15 pada background menggunakan & dan kemudian gunakan perintah kill untuk menghentikan proses diikuti job number.
- Jalankan sleep 15 pada background menggunakan % dan kemudian gunakan kill untuk menghentikan sementara proses. Gunakan bg untuk melanjutkan menjalankan proses.
- Jalankan sleep 60 pada background 5 kali dan terminasi semua pada dengan menggunakan perintah killall.
- Gunakan perintah ps, w dan top untuk menunjukkan semua proses yang sedang dieksekusi.

- Gunakan perintah `ps -aeH` untuk menampilkan hierarki proses. Carilah init proses. Identifikasi sistem daemon yang penting! Identifikasi shell dan subproses!
- Kombinasikan `ps -fae` dan `grep`, apa yang anda lihat?
- Jalankan proses `sleep 300` pada background. Log off komputer dan login kembali. Lihat daftar semua proses yang sedang berjalan! Apa yang terjadi pada proses `sleep`?

PERTEMUAN 4

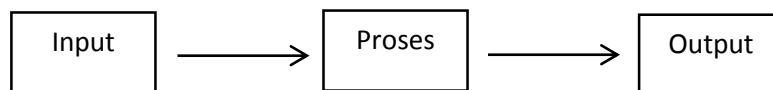
OPERASI INPUT OUTPUT

1. Tujuan

- Memahami konsep proses input dan output dan pembelokan.
- Memahami standar input, output, dan eror.
- Menggunakan notasi output, append dan here document.
- Memahami konsep PIPE dan filter

2. Dasar Teori

1. Proses Input dan Output



Interuksi (*command*) yang diberikan pada Linux melalui Shell disebut dengan eksekusi program yang selanjutnya disebut dengan proses. Setiap kali interuksi diberikan, maka Linux kernel akan menciptakan sebuah proses dengan memberikan nomor PID (*Process Identity*). Proses dalam Linux selalu membutuhkan Input dan akan menghasilkan Output

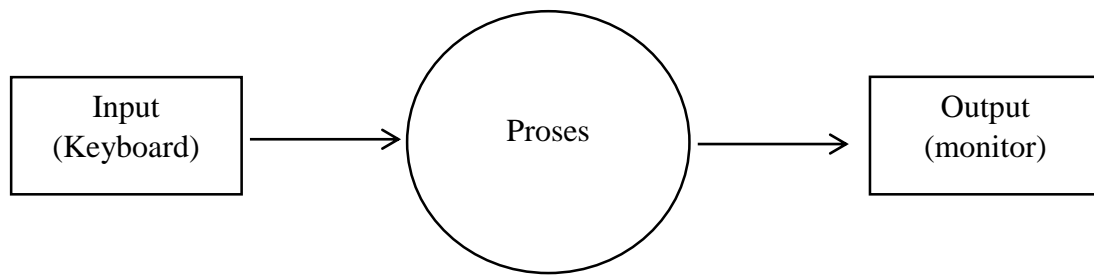
Dalam Linux konteks input / output meliputi :

- Keyboard (input)
- Layar (output)
- File
- Struktur data kernel
- Peralatan I/O lainnya (misal *Network*)

2. File Descriptor

Linux berkomunikasi dengan file melalui *file descriptor* yang direpresentasikan melalui angka yang dimulai dari 0, 1, 2 dan seterusnya.

Tiga buah *file descriptor* standar yang diciptakan oleh proses



adalah :

- 0 = keyboard (standar input)
- 1 = layar (standar output)
- 2 = layar (standar error)

1

0

2

Linux tidak membedakan antara peralatan hardware dan file, Linux memanipulasi peralatan hardware sama dengan file.

3. **Pembelokan (Redirection)**

Pembelokan dilakukan untuk standar input, output dan error. Yaitu untuk mengalihkan file descriptor dari 0, 1 dan 2. Simbol untuk pembelokan adalah :

0< atau < untuk standar input.

1> atau > untuk standar output.

2> untuk standar error.

4. **Pipa (Pipeline)**

Mekanisme pipa digunakan sebagai alat komunikasi antar proses.

Input \longrightarrow Proses1 \longrightarrow Output = Input \longrightarrow Proses2 \longrightarrow Output

Dalam gambar diatas dapat diartikan proses1 mendapatkan inputan yang nantinya menghasilkan output, selanjutnya output tersebut digunakan sebagai input oleh proses2. Hubungan output ini dinamakan pipa, yang menghubungkan proses1 dengan proses2 dan dinyatakan dengan simbol “|”. Proses1 | Proses2

5. **Filter**

Filter adalah utilitas Linux yang dapat memproses standar input

(dari keyboard) dan menampilkan hasilnya yang telah diproses pada standar output (layar). Contoh filter adalah :

- Perintah *grep*

Digunakan untuk menyaring masukkan dan menampilkan baris-baris yang

hanya mengandung pola yang ditentukan. Pola ini disebut regular expression.

- Perintah *wc*

Digunakan untuk menghitung jumlah baris, kata dan karakter dari baris-baris

masukkan yang diberikan kepadanya. Untuk mengetahui berapa baris gunakan

option *-l*, untuk mengetahui berapa kata gunakan option *-w* dan untuk mengetahui berapa karakter menggunakan option *-c*. Jika salah satu option

tidak digunakan maka tampilnya adalah jumlah baris, jumlah kata dan jumlah

karakter.

- Perintah *sort*

Digunakan untuk mengurutkan masukkannya berdasarkan urutan nomor ASCII

dari karakter.

- Perintah *cut*

Digunakan untuk mengambil kolom tertentu dari baris-baris masukkannya

yang ditentukan pada option *-c*.

- Perintah *uniq*

Digunakan untuk menghilangkan baris-baris berurutan yang mengalami

duplikasi, biasanya digabungkan dalam pipeline dengan sort.

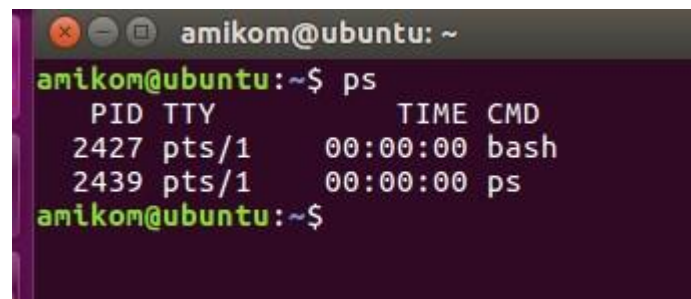
Percobaan :

1. Login sebagai user.
2. Bukakan Console Terminal dan lakukan percobaan-percobaan dibawah ini.
Perhatikan hasil setiap percobaan.
3. Selesaikan soal-soal latihan.

Percobaan 1 : File descriptor

1. Ouput ke layar (standar output), input dari sistem (karnel).

\$ ps



```
amikom@ubuntu: ~  
amikom@ubuntu:~$ ps  
  PID TTY          TIME CMD  
 2427 pts/1        00:00:00 bash  
 2439 pts/1        00:00:00 ps  
amikom@ubuntu:~$
```

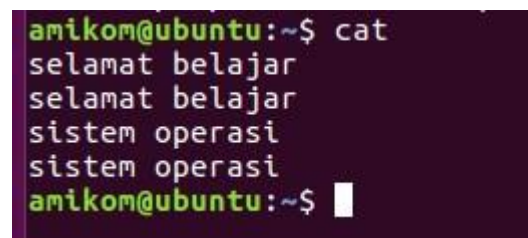
2. Output ke layar (standar output), input dari keyboard (standar input).

\$ cat

selamat belajar sistem operasi

selamat belajar sistem operasi

[Ctrl + d]



```
amikom@ubuntu:~$ cat  
selamat belajar  
selamat belajar  
sistem operasi  
sistem operasi  
amikom@ubuntu:~$
```

3. Input dari keyboard dan output ke alamat email.

\$ mail amikom@amikom.ac.id

contoh surat yang langsung dibuat

pada standar input (keyboard)

[Ctrl + d]

```
amikom@ubuntu:~$ mail amikom@amikom.ac.id
The program 'mail' is currently not installed. You can install it by typing:
sudo apt install mailutils
amikom@ubuntu:~$
```

packet mail harus dinstal dahulu agar bisa digunakan.

4. Input nama direktori, output tidak ada (karena direktori baru dan belum ada nama yang sama), membuat direktori baru lagi dengan nama yang sama maka akan tampil pesan error pada layar (standar error).

\$ mkdir direktoribaru

\$ mkdir direktoribaru (terdapat pesan error)

```
amikom@ubuntu:~$ mkdir amikom
amikom@ubuntu:~$ mkdir amikom
mkdir: cannot create directory 'amikom': File exists
amikom@ubuntu:~$
```

Percobaan 2 : Pembelokan (redirection)

1. Pembelokan standar output.

\$ cat 1> filebaru.txt

ini adalah isian dari filebaru.txt

yang menggunakan pembelokan standar output

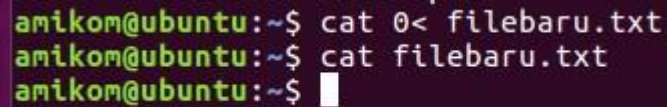
[Ctrl + d]

```
amikom@ubuntu:~$ cat 1> filebaru.txt
ini adalah isian dari filebaru.txt
dengan perintah pembelokan standar output
amikom@ubuntu:~$
```

2. Pembelokan standar input, yaitu input dibelokkan dari keyboard menjadi file.

\$ cat 0< filebaru.txt

\$ cat filebaru.txt



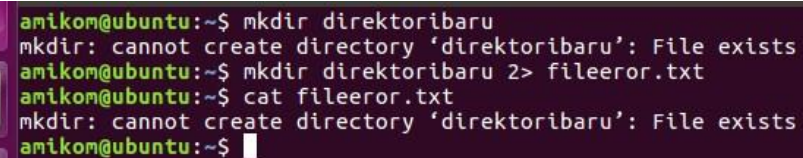
```
amikom@ubuntu:~$ cat 0< filebaru.txt
amikom@ubuntu:~$ cat filebaru.txt
amikom@ubuntu:~$
```

3. Pembelokan standar eror untuk disimpan di file

\$ mkdir direktoribaru (terdapat pesan eror)

\$ mkdir direktoribaru 2> fileerror.txt

\$ cat fileerror.txt



```
amikom@ubuntu:~$ mkdir direktoribaru
mkdir: cannot create directory 'direktoribaru': File exists
amikom@ubuntu:~$ mkdir direktoribaru 2> fileerror.txt
amikom@ubuntu:~$ cat fileerror.txt
mkdir: cannot create directory 'direktoribaru': File exists
amikom@ubuntu:~$
```

4. Notasi 2>&1 : pembelokan standar eror (2>) adalah identik dengan file descriptor 1.

\$ ls filesaya (terdapat pesan eror)

\$ ls filesaya 2> out.txt

\$ cat out.txt

\$ ls filebaru 2> out.txt 2>&1

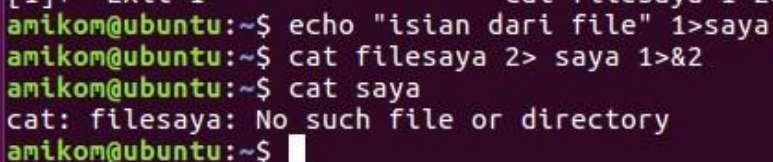
\$ cat out.txt

5. Notasi 1>&2 (atau >&2) : pembelokan standar output adalah sama dengan file descriptor 2 yaitu standar eror.

\$ echo "isian dari file" 1> saya

\$ cat filesaya 2> saya 1>&2

\$ cat saya



```
amikom@ubuntu:~$ echo "isian dari file" 1>saya
amikom@ubuntu:~$ cat filesaya 2> saya 1>&2
amikom@ubuntu:~$ cat saya
cat: filesaya: No such file or directory
amikom@ubuntu:~$
```

6. Notasi >> (append)

\$ echo "kalimat pertama" > surat

\$ echo "kalimat kedua" >> surat

```
$ echo "kalimat ketiga" >> surat
$ cat surat
$ echo "kalimat keempat" > surat
$ cat surat
```

A terminal window titled 'amikom@ubuntu: ~' showing a series of commands and their outputs. The user creates a file named 'surat' with 'echo' and appends content with '>>'. Then, they use 'cat' to display the file's contents. The process is repeated for a second file.

```
amikom@ubuntu:~$ echo "kalimat pertama" > surat
amikom@ubuntu:~$ echo "kalimat kedua" >> surat
amikom@ubuntu:~$ echo "kalimat ketiga" >> surat
amikom@ubuntu:~$ cat surat
kalimat pertama
kalimat kedua
kalimat ketiga
amikom@ubuntu:~$ echo "kalimat keempat" > surat
amikom@ubuntu:~$ cat surat
kalimat keempat
amikom@ubuntu:~$
```

7. Notasi here document (<<++++) digunakan sebagai pembatas input dari keyboard. Perhatikan bahwa tanda pembatas dapat digantikan dengan tanda apa saja, namun harus sama dan tanda penutup harus diberikan pada awal baris.

```
$ cat <<++
selamat belajar
sistem operasi
semangat
++
```

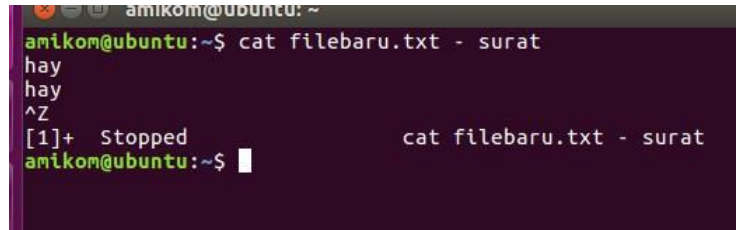
A terminal window titled 'amikom@ubuntu: ~' demonstrating the use of the 'here document' notation. The user enters 'cat <<++' followed by several lines of text, and then '++' on a new line to close the document. The text is then displayed on the screen.

```
amikom@ubuntu:~$ cat <<++
> selamat belajar
> sistem operasi
> semangat
> ++
selamat belajar
sistem operasi
semangat
amikom@ubuntu:~$
```

8. Notasi – (input keyboard) adalah representasi input dari keyboard. Artinya menampilkan file 1, kemudian menampilkan input dari keyboard dan menampilkan file 2. Perhatikan bahwa

notasi “-“ berarti menyelipkan input dari keyboard.

\$ cat filebaru.txt - surat



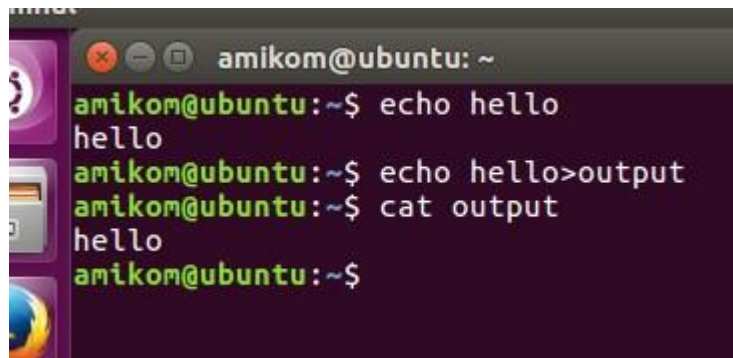
```
amikom@ubuntu:~$ cat filebaru.txt - surat
hay
hay
^Z
[1]+  Stopped                  cat filebaru.txt - surat
amikom@ubuntu:~$
```

9. Untuk membelokan standar output ke file, digunakan operator >

\$ echo hello

\$ echo hello > output

\$ cat output

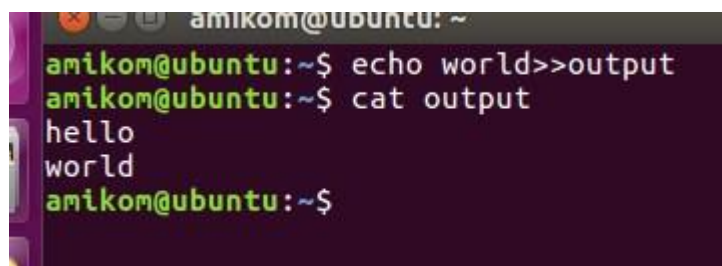


```
amikom@ubuntu:~$ echo hello
hello
amikom@ubuntu:~$ echo hello > output
amikom@ubuntu:~$ cat output
hello
amikom@ubuntu:~$
```

10. Untuk menambahkan output ke file digunakan operator >>

echo world >> output

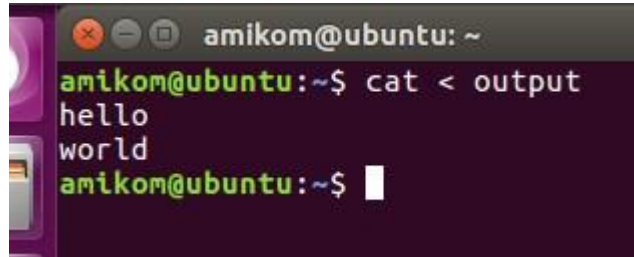
\$ cat output



```
amikom@ubuntu:~$ echo world >> output
amikom@ubuntu:~$ cat output
hello
world
amikom@ubuntu:~$
```

11. Untuk membelokkan standar input digunakan operator <

\$ cat < output

A terminal window titled 'amikom@ubuntu: ~' showing a command prompt. The user enters 'cat < output', and the terminal displays the contents of the 'output' file: 'hello' and 'world'. The prompt returns to '\$ '.

12. Pembelokkan standar input dan standar output dapat

dikombinasikan tetapi tidak boleh menggunakan nama file yang sama sebagai standar input dan output.

\$ cat < output > out

\$ cat out

\$ cat < output >> out

\$ cat out

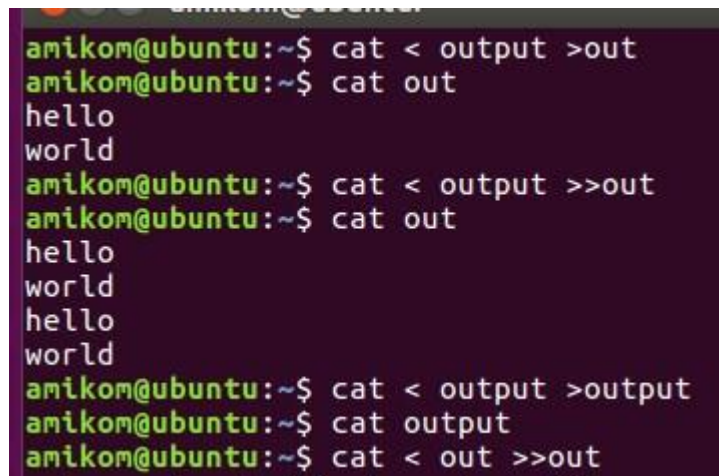
\$ cat < output > output

\$ cat output

\$ cat < out >> out (proses tidak berhenti)

[Ctrl + c]

\$ cat out

A terminal window titled 'amikom@ubuntu: ~' showing a series of commands and their outputs. The commands are: 'cat < output > out' (output: 'hello', 'world'), 'cat out' (output: 'hello', 'world'), 'cat < output >> out' (output: 'hello', 'world'), 'cat out' (output: 'hello', 'world'), 'cat < output > output' (output: 'hello', 'world'), 'cat output' (output: 'hello', 'world'), and 'cat < out >> out' (output: 'hello', 'world'). The prompt returns to '\$ ' after each command.

Percobaan 3 : Pipa (Pipeline)

1. Operator pipa() digunakan untuk membuat eksekusi proses dengan melewati data langsung ke data lainnya.

\$ who

\$ who | sort

\$ who | sort -r

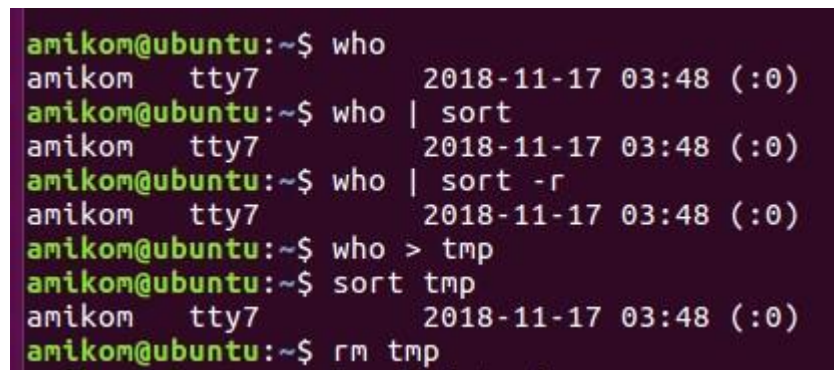
\$ who > tmp

\$ sort tmp

\$ rm tmp

\$ ls -l /etc | more

\$ ls -l /etc | sort | more



```
amikom@ubuntu:~$ who
amikom  tty7      2018-11-17 03:48 (:0)
amikom@ubuntu:~$ who | sort
amikom  tty7      2018-11-17 03:48 (:0)
amikom@ubuntu:~$ who | sort -r
amikom  tty7      2018-11-17 03:48 (:0)
amikom@ubuntu:~$ who > tmp
amikom@ubuntu:~$ sort tmp
amikom  tty7      2018-11-17 03:48 (:0)
amikom@ubuntu:~$ rm tmp
```

Percobaan 4 : Filter

2. Pipa juga digunakan untuk mengkombinasikan utilitas sistem untuk membentuk fungsi yang lebih kompleks.

\$ w -h | grep <user>

\$ grep <user> /etc/passwd

\$ ls /etc | wc

\$ ls /etc | wc -l

\$ cat > kelas1.txt

Agus

Latifah

Galih

Siti

[Ctrl + d]

\$ cat > kelas2.txt

Raha

Aziz

Asep

[Ctrl + d]

```
amikom@ubuntu:~$ w -h | grep amikom
amikom  tty7      :0                03:48   36:46   1:57   1.27s /sbin/upstart --
user
amikom@ubuntu:~$ grep amikom /etc/passwd
amikom:x:1000:1000:amikom,,,:/home/amikom:/bin/bash
amikom@ubuntu:~$ ls /etc | wc
      230      230      2216
amikom@ubuntu:~$ ls /etc | wc -l
230
amikom@ubuntu:~$ cat > kelas1.txt
Agus
Latifah
Galih
Siti
amikom@ubuntu:~$ cat > kelas2.txt
Raha
Aziz
Asep
amikom@ubuntu:~$ cat kelas1.txt | sort
Agus
Galih
Latifah
Siti
```

\$ cat kelas1.txt kelas2.txt | sort

\$ cat kelas1.txt kelas2.txt > kelas.txt

\$ cat kelas1.txt | sort | uniq

PERTEMUAN 5

SISTEM FILE (HAK AKSES)

1. Tujuan

Setelah Mempelajari dan menerapkan modul ini mahasiswa di harapkan mampu

1. Memahami atribut file dan izin akses
2. Memahami perintah untuk mengubah izin akses suatu file
3. menggunakan perintah perintah

2. Dasar Teori

1. Atribut File

File Mempunyai atribut antara lain :

- Tipe file

Karakter	Arti
-	File Biasa
d	Direktori
l	Symbolic link
b	Block Special File
c	Character Special File
s	Socket Link
p	FIFO

- Ijin akses : menentukan hak user terhadap file ini.
- Jumlah link : jumlah link untuk file ini.
- Pemilik (Owner) : menentukan siapa pemilik file ini
- Group : menentukan group yang memiliki file ini
- Jumlah karakter : menentukan ukuran file dalam byte
- Waktu pembuatan : menentukan kapan file terakhir dimodifikasi
- Nama file : menentukan nama file yang dimaksud

Contoh :

-rw-rw-r--	1	bin	auth	1639	Oct 31 20:19	/etc/passwd
Ijin akses	Jml link	Pemilik	Group	Jml karakter	Waktu	Nama file
tipe						

2. Ijin Akses

Setiap obyek pada Linux harus mempunyai pemilik, yaitu nama pemakai

Linux (account) yang terdaftar pada /etc/passwd .

Ijin akses dibagi menjadi 3 peran yaitu :

- A. Pemilik (Owner)
- B. Kelompok (Group)
- C. Lainnya (Others)

Setiap peran dapat melakukan 3 bentuk operasi yaitu :

A. Pada File

- R (Read) Ijin untuk membaca
- W (Write) Ijin untuk mengubah / membuat
- X (Execute) Ijin untuk menjalankan program

B. Pada Direktori

- R (Read) Ijin untuk membaca daftar file dalam direktori
- W (Write) Ijin untuk mengubah/membuat file di direktori
- X (Execute) Ijin untuk masuk ke direktori (cd)

Pemilik File/Direktori dapat mengubah ijin akses sebagai berikut :

```
-rwxrwxrwx 1 student test 1639 Oct 31 20:19 file
```

Diagram illustrating the mapping of permissions to user categories:

- user (first three characters)
- group (next three characters)
- other (last three characters)

Format untuk mengubah ijin akses

chmod [ugoa] [= + -] [rwx] File(s)

chmod [ugoa] [= + -] [rwx] Dir(s)

dimana u = user (pemilik)

g = group (kelompok)

o = others (lainnya)

a = all

Format lain dari chmod adalah menggunakan bilangan octal sebagai berikut

r w x

4 2 1 = 7

3. User Mask

Untuk menentukan ijin akses awal pada saat file atau direktori dibuat digunakan perintah umask. Untuk menghitung nilai default melalui umask pada file,

maka dapat dilakukan kalkulasi sebagai berikut :

Kreasi file (biasa) 6 6 6

Nilai umask 0 2 2

6 4 4

Kreasi direktori 7 7 7

Nilai umask 0 2 2

7 5 5

Percobaan1 :

1. Login sebagai user.
2. Bukalah Console Terminal dan lakukan percobaan-percobaan di bawah ini kemudian analisa hasil percobaan.
3. Selesaikan soal-soal latihan.

1. Melihat identitas diri melalui *etc/password* atau *etc/group*, informasi apa di tampilkan?

```
$ id
```

```
$ grep <user> /etc/passwd
```

```
$ grep [Nomor group id] /etc/group
```

2. Memeriksa direktori home

```
$ ls -ld /home/<user>
```

3. Mengubah Ijin akses (*chmod*). Perhatikan dan analisis setiap perubahannya

```
$ touch file1 file2 file2
```

```
$ ls -l
```

```
$ chmod u+x file1
```

```
$ ls -l
```

```
$ chmod g=w file1
```

```
$ ls -l file1
```

```
$ chmod a=x file2
```

```
$ ls -l file2
```

```
$ chmod u+x,g-r,o=w file3
```

```
$ ls -l file3
```

```
$ chmod 751 file1
```

```
$ chmod 624 file2
$ chmod 430 file3
$ ls -l file1 file2 file3
```

4. Mengganti kepemilikan digunakan perintah *chown*. Masuk ke root untuk mengganti kepemilikan tersebut

```
$ su root
$ echo selamat pagi > file1
$ ls -l file1
$ chown <user-baru> file1 contoh :chown siswa file1
$ ls -l file1
```

5. Ubahlah izin akses home directory <user> (student) pada root sehingga <userbaru>

(student1) pada satu group dapat mengakses home directory <user>. Hal ini dimaksudkan agar file file1 yang sudah diubah kepemilikannya dapat diakses <user-baru>. Perubahan izin akses home directory <user> hanya dapat dilakukan pada root.

```
$ chmod g+rwx /home/<user> contoh : chmod g+rwx /home/mahasiswa
$ ls -l /home
$ exit
```

6. Sekarang cobalah untuk substitute user ke <user-baru> (student1). Cobalah untuk mengakses file file1

```
$ su <user-baru>
$ ls -l f1
$ cat f1
$ exit
```

7. Mengubah group dengan perintah *chgrp*

```
$ grep root /etc/group
```

```
$ grep other /etc/group
$ su
$ chgrp root f1
$ ls -l f1
$ chgrp <group-baru> f3
$ ls -l f3
$ exit
```

Percobaan 2:

1. Menentukan ijin akses awal pada saat file atau direktori dibuat

```
$ touch myfile
$ ls -l myfile
```

2. Melihat nilai umask

```
$ umask
```

3. Modifikasi nilai umask

```
$ umask 027
$ umask
$ touch file_baru
$ mkdir mydir
$ ls -l
$ umask 077
$ touch xfiles
$ mkdir xdir selanjutnya $ ls -l
```

PERTEMUAN 6

BEKERJA DENGAN BASH SHELL

1. Tujuan

Setelah mempelajari dan menerapkan modul ini, siswa di harapkan mampu :

1. Memahami shell pada sistem operasi Linux.
2. Menggunakan feature history pada Bash Shell.
3. Mengubah feature history pada Bash Shell.
4. Mengubah prompt shell.
5. Melakukan konfigurasi Bash Shell untuk menjalankan skrip secara otomatis.
6. Membuat dan mengeksekusi shell script sederhana melalui editor vi.
7. Memahami job control.
8. Memahami stack.
9. Menggunakan alias.

2. Dasar Teori

1. Shell

Shell adalah Command executive, artinya program yang menunggu instruksi

dari pemakai, memeriksa sintak dari instruksi yang diberikan, kemudian mengeksekusi

perintah tersebut. Shell ditandai dengan prompt. Untuk pemakai menggunakan prompt

\$ dan untuk superuser menggunakan prompt #.

Beberapa Macam Shell

- /bin/sh

Bourne shell, dirancang oleh steve Bourne dari AT&T

- /bin/csh

Dikembangkan oleh UNIX Berkeley yang di kenal dengan C-Shell

- /bin/bash

Kompatibel dengan Bourne Shell dan juga mengadaptasi kemampuan Korn-Shell. Perbedaan mendasar antara Shell di atas hampir tidak ada, kecuali pada fasilitas pemrograman dan editing.

2. Profile

Pada saat login, program akan menjalankan beberapa program yaitu

1. /etc/profile

Berisi shell script yang berlaku untuk seluruh pengguna Linux.

2. Profil untuk setiap pemakai

Pada home directory, login pertama kali akan memeriksa file `.bash_profile`. Bila tidak ada, maka file `.bash_login` akan dicari. Bila `.bash_login` tidak ada, maka dicari file bernama `.profile`.

3. .bashrc

File ini akan dieksekusi untuk perpindahan dari satu shell ke shell yang lain melalui instruksi `su`.

4. .bash_logout

Pada saat logout, maka bash akan mencari file `.bash_logout`. Bila ada, file tersebut akan dieksekusi sebelum logout

isi dari /etc/profile

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1" ]; then
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi
```

keterangan :

1. Path

merupakan daftar nama direktori. Bila sebuah instruksi diberikan dari prompt shell, maka instruksi tersebut akan dicari pada daftar tersebut.

2. PS1

adalah prompt dimana

\u = Nama User

\h = Nama Host

\W = Nama working directory

3. History

History diadaptasi dari C-Shell, yaitu catatan dari semua instruksi yang sejauh

ini telah dilakukan. Catatan ini dapat dilihat sebagai history, kemudian dapat dipilih kembali, diedit dan dieksekusi. History memudahkan pemakai untuk mengedit kembali instruksi kompleks dan panjang, terutama bila terjadi kesalahan pada penulisan instruksi maupun parameter.

Navigasi pada daftar history menggunakan karakter kontrol sebagai berikut :

^P (Ctrl-P) melihat instruksi sebelumnya

^N (Ctrl-N) melihat instruksi berikutnya

!! eksekusi kembali instruksi sebelumnya

!! -3 3 instruksi sebelumnya akan diulang

!!88 ulangi instruksi no 88

4. Bash Script

Bash-script adalah file yang berisi koleksi program yang dapat dieksekusi.

Untuk eksekusi bash script gunakan . sebelum file bash-script yang berarti eksekusi shell dan tanda ./ berarti file bash-script berada pada direktori actual.

5. Job Kontrol

Job adalah sebuah eksekusi program yang di berikan pada kernel. sebuah job di anggap selesai, bila eksekusi program tersebut berakhir,. Eksekusi Job adalah sama dengan eksekusi program, baik background maupun foreground.

6. Editor Vi

Vi adalah full screen editor, artinya editor tersebut dapat memanfaatkan fasilitas satu layar penuh. Vi mempunyai 2 buah modus, yaitu:

- Command Line

Editor vi menginterpretasikan input sebagai instruksi untuk dieksekusi oleh editor, contoh seperti mencari teks, mengganti teks secara otomatis dan lainnya

- Editing

Editor vi menginterpretasikan input sebagai teks yang akan dimasukkan ke dalam buffer editor. Pada bagian bawah layar akan tampil teks “INSERTING”.

Pada awal vi dijalankan, maka program memasuki command mode. Dengan menekan tombol “i” maka akan memasuki editing. Untuk kembali ke command mode, tekan tombol Esc.

Percobaan 1:

Membuat bash script dan menjalankannya

1. `.$ vi p1.sh`

Isi dengan keterangan seperti di bawah ini

```
Membuat Program bash Script
```

<code>^G</code> Get Help	<code>^O</code> WriteOut	<code>^R</code> Read File	<code>^Y</code> Prev Page	<code>^K</code> Cut Text	<code>^C</code> Cur Pos
<code>^X</code> Exit	<code>^J</code> Justify	<code>^W</code> Where Is	<code>^V</code> Next Page	<code>^U</code> UnCut Text	<code>^T</code> To Spell

2. Mengubah program menjadi executable

```
$ chmod +x p1.sh
amikom@amikom-VirtualBox:~$ ls -l p1.sh
-rw-rw-r-- 1 amikom amikom 28 Nov 14 19:07 p1.sh
$ ls -l p1.sh
-rwxrwxr-x 1 amikom amikom 28 Nov 14 19:07 p1.sh
```

3. Menjalankan bash script

```
$ bash p1.sh
```

Hasilnya :

```
"Membuat Program bash Script"  
$ sh p1.sh  
p1.sh: 1: p1.sh: Membuat Program bash Script
```

4. 4. Konvensi dalam pembuatan script shell dinyatakan sebagai `#!/bin/bash`. Tambahkan pada file `p1.sh` konvensi tersebut

```
"Membuat Program bash Script"  
#!/bin/bash
```

5. Membuat file `p2.sh`

```
$ vi p2.sh  
"Program Bash Script 2"  
#!/bin/bash
```

6. Jalankan file 1 dan file 2 dengan 1 baris script menggunakan tanda;

PERTEMUAN 7

RESPONSI

“SOAL RESPONSI SISTEM OPERASI”

1. Lihatlah identitas user (nomer id dan group idnya)
2. Buatlah Folder Dengan Nama **Responsi xxxx** (xxxx= 4 digit nim terakhir & folder menggunakan spasi)

Contoh : Responsi 9424

3. Didalam folder responsi xxxx
Buatlah nama file
 1. Laporan.txt
 2. Penjualan.sh
 3. Pembelian.sh
 4. Kategoriproduk.txt
4. Di dalam laporan.txt di isi dengan
“laporan penjualan 20 agustus – 20 september 2018”
 1. 25.000
 2. 30.000
 3. 15.000
 4. 80.000
 5. 35.000
 6. 80.000
 7. 90.000
 8. 27.000
 9. 35.000
 10. 24.000
5. Buat folder dengan nama kategori di dalam folder responsi
6. Carilah kata 20 agustus pada file laporan.txt
7. Belokkan standar output dari laporan.txt ke laporanfix.txt
8. Bukalah aplikasi libre & firefox pada system operasi Ubuntu, kemudian lihat status prosesnya dan berhentikan aplikasi tersebut
9. Lihatlah hak akses laporan.txt dan jelaskan maksud dari hak akses tersebut
10. Ubahlah hak akses dari pembelian.sh dengan ketentuan sebagai berikut
 - Owner = baca, eksekusi
 - Grup = tulis
 - Other = baca
11. Pindahkan file kategoriproduk.txt ke dalam folder kategori

PERTEMUAN 8

MANAJEMEN APLIKASI

1. Tujuan

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- Mengerti konsep RPM, TAR dan GZIP.
- Menggunakan RPM.
- Menggunakan TAR dan GZIP untuk instalasi software.

2. Dasar Teori

1. Manajemen Paket Software

Setiap system Linux mempunyai manajemen paket software, yang paling populer adalah RPM (RedHat Package Management). RPM mengatur instalasi paket software, maintenance/upgrade dan menghapus paket software dari system, atau lebih dikenal dengan install dan uninstall (install / remove).

RPM menyimpan informasi tentang paket yang diinstalasi dalam sebuah database. Penghapusan paket berarti juga menghapus semua files dan direktori yang terdaftar pada database tersebut, lengkap dengan nama PATH (lokasi dimana file dan direktori tersebut berada).

RPM menyimpan paket dalam bentuk file yang telah dikompres dan ditulis sebagai file dengan ekstensi *.rpm.

2. Fungsi Manajer Paket Software

- Menghitung besar paket yang disesuaikan dengan kapasitas penyimpanan disk yang masih tersedia, apakah cukup atau tidak.
- Memeriksa apakah ada library atau file-file lain yang dibutuhkan untuk software tersebut.
- Menghindari konflik dengan software yang telah terpasang di system.
- Proses instalasi tidak mengacaukan system (membuat system file menjadi terganggu / korup).
- Upgrade ke versi yang baru tanpa mengganggu konfigurasi yang sudah ada.
- Verifikasi files dalam paket tersebut.

3. Paket Software

Terdiri dari 2 jenis

1. Paket binary (biner)

Terdiri atas kumpulan program executable.paket ini berekstensi *.rpm

2. Paket source

Berisi teks dari program yang kemudian dapat di kompilasi menjadi executable dan mempunyai ekstensi *.src.rpm

4. Nama Paket

Penamaan paket di atur dengan konvensi sebagai berikut

- Nama
- Versi
- Release
- Platform arsitektur



Platform :

i386	Intel
alpha	DEC
sparc	SUN
mips	MIPS
ppc	PowerPC
m68k	Motorola 680x0
noarch	Bebas, tanpa platform

5. RPM Query

RPM dengan opsi `-q` memberikan informasi tentang paket sebagai berikut :

```
# rpm -q samba
samba -2.0.5-1S
```

Informasi tentang versi paket samba adalah versi 2.0.5.

Beberapa sub-opsi dapat diberikan, antara lain :

`i` menampilkan informasi yang lebih rinci

`l` list (daftar) semua file(s)

`d` tampilkan hanya file dokumentasi saja

`c` tampilkan hanya konfigurasi file

f info tentang paket memiliki file apa saja
p berfungsi pada paket yang belum diinstalasi
--scripts menampilkan script untuk instalasi

6. TAR

Tar singkatan dari Tape ARchive. Tar mula-mula didesain untuk backup tape, tetapi digunakan untuk membuat file tar pada semua sistem file. tar membuat satu "tar file" (yang disebut dengan "tarball") pada beberapa file dan direktori. File tar tidak dikompresi, hanya sebuah file heap yang dibentuk bersama dalam satu kontainer. Sehingga file tar akan mempunyai jumlah byte yang sama dengan semua file individu yang dikombinasikan ditambah sedikit file ekstra. File tar dapat dikompresi dengan menggunakan gzip atau bzip2.

Percobaan 1:

1. Pada percobaan ini paket samba dapat didownload pada website

<http://www.samba.org>. Tetapi pada percobaan ini sudah disediakan pada halaman website ini.

2. Ekstrak file samba menggunakan perintah tar

```
# tar -xvf samba
```

3. Informasi tentang paket software

```
# rpm -qi samba | more
```

4. Instal dari CDROM, bukan dari paket yang sudah diinstalasi. Untuk itu pindah direktori ke tempat dimana file rpm tersebut berada

```
# cd /mnt/cdrom/Packages/RPMS
```

```
# rpm -qpi samba-2.0.5-1S.i386.rpm | more
```

5. Melihat files apa saja yang sudah diinstalasi, gunakan sub-opsi -l.

```
# rpm -ql samba | more
```

6. Melihat file konfigurasi

```
# rpm -qc samba | more
```

7. Melihat dokumentasi file

```
# rpm -qd samba | more
```

8. Melihat script file yang digunakan untuk instalasi. Perhatikan bahwa hasil output tergantung atas versi Linux yang digunakan

```
# rpm -q --scripts samba
```


Percobaan 2: menghapus paket

1. Opsi `-e` digunakan untuk menghapus paket yang sudah instalasi. Bila ada ketergantungan paket lain atas paket yang akan dihapus, maka RPM membatalkan penghapusan tersebut

```
# rpm -q apache
```

```
# rpm -e apache
```

2. Paket apache tidak bisa dihapus karena masih ada paket lain yang tergantung atas paket tersebut, yaitu modul PHP. Maka paket `mod_php3` harus dihapus lebih dahulu

```
# rpm -e mod_php3
```

3. Setelah itu paket apache baru bisa dihapus

```
# rpm -e mod_php3
```

```
# rpm -q --scripts samba
```

Percobaan 3: Menggunakan Tar

1. Logout dari root dan login sebagai `<user>`.

2. Pastikan berada pada home directory. Dan lihatlah isi home direktori Pindah ke direktori Desktop. Setelah itu kembali ke home direktori

```
$ cd
```

```
$ ls -al
```

```
$ cd Desktop
```

```
$ cd
```

3. Buatlah file tar pada direktori Desktop

```
$ tar -cvf desktop.tar Desktop
```

```
$ ls -al
```

4. Rename direktori Desktop menjadi Desktop.old. Lihatlah file desktop.tar pada direktori Desktop.old

```
$ mv Desktop Desktop.old
```

```
$ ls -l
```

```
$ cd Desktop.old
```

```
$ ls -l
```

```
$ cd ..
```

5. Ekstraksi isi dari file tar. Lihatlah isi direktori dan seharusnya akan terlihat direktori Desktop yang original

```
$ tar -xvf desktop.tar
```

```
$ ls -al
```

```
$ cd Desktop
```

6. Hapus direktori Desktop

```
$ rm -rf Desktop
```

Percobaan 4 : Menggunakan GZIP

1. Buatlah file gzip dari desktop.tar, pastikan terbentuk file desktop.tar.gz

```
$ gzip desktop.tar
```

```
$ ls -al
```

2. Dekompresi file gzip desktop.tar.gz dan cobalah untuk ekstraksi isi file tar

```
$ gzip -d desktop.tar.gz
```

```
$ ls -al
```

```
$ tar -xvf desktop.tar
```

```
$ ls -al
```

```
$ cd Desktop
```

3. Hapus file tar dan direktori Desktop.old

```
$ rm desktop.tar
```

```
$ rm -rf Desktop.old
```

PERTEMUAN 9

MANAJEMEN PERANGKAT KERAS

1. Tujuan

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- Mengetahui bagaimana melihat perangkat keras yang terpasang pada system komputer.
- Menggunakan perintah mount dan umount pada sistem file
- Menggunakan perintah-perintah untuk manajemen perangkat keras.

2. Dasar Teori

1. File Perangkat Keras

/dev berisi file device (perangkat) yang merupakan aspek penting pada system file Linux. /dev/cdrom dan /dev/fd0 merupakan drive CD-ROM dan floppy pada komputer Anda. Kita dapat melakukan akses read dan write pada perangkat. Sebagai contoh /dev/dsp merupakan perangkat speaker. Sembarang data yang ditulis ke file ini akan dialihkan ke speaker. 'cat /boot/vmlinuz > /dev/dsp' menyebabkan kita dapat mendengarkan suara dari speaker. Untuk mencetak file dapat dikirim ke perangkat /dev/lp0. Mengirim data ke dan membaca data dari /dev/ttyS0 akan menyebabkan komunikasi dengan perangkat modem. Mayoritas device berupa block device atau character device. Block device adalah device yang menyimpan atau membawa data, character device adalah device yang mengirim atau transfer data. Sebagai contoh, diskette drive, hard drive dan CDROM drive adalah block device, sedangkan serial port, mouse dan paralel printer adalah character device.

Beberapa file perangkat yang umum digunakan yang perlu diingat adalah :

/dev/ttyS0 (First communication port, COM1) : First serial port (mouse, modem)

/dev/psaux (PS/2) : PS/2 mouse connection (mouse, keyboard)

/dev/lp0 (First printer port, LPT1) : First parallel port (printer, scanner dsb)

/dev/dsp (First audio device) : sound card, digitized voice dan PCM

/dev/usb (USB Device) : node USB device

/dev/sda (C:/SCSI device) : First SCSI device (HDD, Memory stick, external mass storage device seperti CD-ROM pada laptop)

/dev/scd (D:\, SCSI CD-ROM device) : First SCSI CD-ROM device

/dev/js0 (Standard gameport joystick) : First joystick device

Device didefinisikan sebagai tipe seperti block atau character dan nomor mayor

dan minor. Nomor mayor digunakan untuk melakukan kategori device dan nomor minor untuk mengidentifikasi tipe device khusus. Sebagai contoh, semua IDE device dihubungkan dengan primary controller mempunyai nomor mayor 3. Perangkat master dan slave, didefinisikan lebih jauh dengan nomor minor. Terdapat dua nomor sebelum tanggal yang tercetak. Jika kita lakukan perintah `ls -l /hd*` maka akan terlihat nomor mayor untuk perangkat hda dan hdb adalah 3. Nomor minor berubah untuk setiap partisi tertentu. Kita dapat selalu membuat perangkat menggunakan skrip MAKEDEV dimana akan diletakkan pada directory /dev.

2. Perintah Mount Dan Umount

Sebelum menggunakan sistem file, harus di-mount terlebih dahulu. Kemudian sistem operasi dapat mengerjakan penyimpanan file. Karena semua file UNIX berada pada satu pohon direktori, operasi mount akan terlihat seperti isi dari sub direktori yang ada pada sistem file yang sudah dilakukan mounting. Contoh perintah mount

```
$ mount /dev/hda2 /home|
$ mount /dev/hda3 /usr
```

Perintah mount mempunyai 2 argumen, argumen pertama adalah file device yang berhubungan dengan disk atau partisi dari sistem file. Argumen kedua adalah directory yang dimounting. Perintah diatas berarti bahwa "/dev/hda2 dilakukan mounting ke /home" begitu juga dengan /usr. Perbedaan antara file device /dev/hda2 dan direktory mount /home adalah file device memberikan akses ke isi disk mentah, direktory mount memberikan akses ke file dari disk. Direktory mount disebut mount point. Linux mendukung beberapa tipe sistem file. Mount akan menebak tipe dari sistem file. Opsi -t fstype akan memberikan spesifikasi tipe sistem flie. Sebagai contoh, untuk mount floppy MS-DOS, dapat menggunakan perintah berikut :

```
$ mount -t msdos /dev/fd0 /floppy
```

Sistem file root dilakukan mounting pada waktu booting. Jika sistem file root tidak dapat dimounting, sistem tidak dapat melakukan booting. Nama sistem file dimounting sebagai root. Sistem file root mula-mula bersifat read-only. Skrip startup kemudian menjalankan fsck untuk melakukan verifikasi validitas dan jika tidak ada permasalahan, dilakukan mounting lagi sehingga write diperbolehkan. Fsck tidak boleh dijalankan pada saat sistem file dimounting, karena setiap perubahan ke sistem file saat fsck berjalan mengakibatkan kesalahan. Bila sistem file root dimounting read-only saat dilakukan pengecekan, fsck dapat memperbaiki permasalahan.

Jika sistem file tidak diperlukan untuk dimounting, dapat dilakukan unmounting dengan perintah umount. Perintah umount mempunyai satu argumen berupa file device atau mount point. Sebagai contoh untuk unmount direktory pada contoh diatas dapat digunakan perintah

```
$ umount /dev/hda2
```

```
$ umount /usr
```

Kita dapat melihat perangkat floppy dan mount point yang diijinkan pada /etc/fstab. Kita dapat melihat perangkat floppy dan mount point yang diijinkan pada /etc/fstab.

```
$ cat /etc/fstab  
/dev/fd0 /mnt/floppy auto rw,user,noauto 0 0  
/dev/hdc /mnt/cdrom iso9660 ro,user,noauto 0 0  
/dev/hdc /mnt/cdrom iso9660 0 0 0
```

Kolom terdiri dari file device, directory mounting, tipe sistem file, opsi, frekuensi backup, fsck pass number (0 berarti tanpa cek). Opsi noauto menghentikan mounting yang dilakukan secara otomatis jika sistem dimulai (misalnya menghentikan mount -a). Opsi user mengijinkan sembarang user melakukan mounting sistem file dan karena alasan keamanan, eksekusi program tidak diijinkan (normal atau setuid) Jika ingin menyediakan akses ke beberapa tipe floppy, perlu diberikan beberapa mount point. Setting berbeda untuk setiap mount point. Sebagai contoh untuk memberikan akses ke floppy MS-DOS dan ext2, dilakukan perubahan baris pada /etc/fstab :

```
/dev/fd0 /dosfloppy msdos user,noauto 0 0 /dev/fd0  
/ext2floppy ext user,noauto 0 0
```

Percobaan :

1. Pada percobaan ini setiap mahasiswa harus membawa sebuah floppy disk dan atau CDROM
2. Login sebagai user.
3. Bukalah Console Terminal dan lakukan percobaan-percobaan di bawah ini kemudian analisa hasil percobaan.
4. Selesaikan soal-soal latihan.

Percobaan 1 : Melihat perangkat pada sistem komputer

1. Melihat daftar perangkat. Perhatikan apakah perangkat-perangkat yang disebutkan pada dasar teori terdapat pada komputer anda. Perhatikan tipe perangkat berupa block device atau character device. Apa yang membedakan suatu perangkat merupakan block device atau character device?

```
$ ls -l /dev
```

2. Perhatikan nomor mayor dan minor pada perangkat hard disk Anda. Apa maksudnya ?

```
$ ls -l /dev/hd*
```

Percobaan 2: Menangani Removable Media

1. Melihat daftar perangkat yang ada pada sistem file utama. Perhatikan titik mount untuk perangkat floppy dan CDROM. Perhatikan opsi yang ada jelaskan maksudnya.

```
$ cat /etc/fstab
```

2. Cobalah melakukan mounting pada floppy disk

```
$ mount /dev/fd0 /mnt/floppy
```

```
$ cd /mnt/floppy
```

```
$ ls -l
```

3. Agar semua perubahan data tertulis pada floppy dan mengambil floppy disk dari sistem file gunakan perintah umount.

```
$ cd
```

```
$ umount /mnt/floppy
```

4. Lakukan hal yang sama untuk perangkat CDROM.

Percobaan 3: Melakukan Format MSDOS pada Floppy

1. Linux dapat membaca dan menulis dengan format MSDOS maupun Linux. Untuk menggunakan floppy MS, dapat digunakan perintah MS -DOS dengan didahului huruf "m". Misalnya, "mdir a:" akan melihat daftar file pada drive a, "mcopy" melakukan copy file, "mdel" melakukan penghapusan file. Lakukan format floppy dengan perintah \$ fdformat /dev/fd0H1440

```
$ mformat a:
```


2. Cobalah melakukan list directory, copy dan delete file

\$ mdir a:

\$ mcopy <namafile> a:

\$ mdel a:<namafile>

3. Lakukan pembuatan direktory pada floppy dengan perintah mmd, copy file dengan mcopy, delete file dengan mdel, pindah directory dengan mcd dan melihat isi directory dengan mdir.

4. Lakukan format floppy disk menggunakan perintah mkfs

\$ mkfs -t msdos /dev/fd0

5. Sebelum menggunakan floppy yang sudah terformat lakukan mounting sistem file

\$ mount /mnt/floppy

6. Untuk melihat apakah floppy sedang digunakan ketikkan

\$ df

7. Lakukan unmount terhadap floppy disk.

\$ umount /mnt/floppy

PERTEMUAN 10

MANAJEMEN USER DAN GRUB

1. Tujuan

- Mahasiswa bisa menambah, memodifikasi dan menghapus user.
- Mahasiswa bisa menambah dan menghapus group.

2. Dasar Teori

Pada Linux Red Hat, informasi tentang user account dan groups disimpan dalam beberapa file teks dalam direktori `/etc/`. Jika sistem administrator membuat user baru ile tersebut harus diedit atau aplikasi harus digunakan untuk perubahan tersebut.

1. File `/etc/passwd`

file `/etc/passwd` adalah file yang berisi daftar user yang dipisahkan dengan baris. Setiap baris berisi informasi mengenai :

- Username = nama user yang diketik saat login.
- Password = berisi password yang dienkripsi (atau x bila shadow password digunakan).
- User ID (UID) = bilangan numerik yang *ekuivalen* dengan username yang menjadi acuan sistem.
- Group ID (GID) = bilangan numerik yang *ekuivalen* dengan nama group primer yang menjadi acuan sistem.
- GECOS = nama histori, kolom GECOS[1] bersifat opsional dan digunakan untuk menyimpan informasi tambahan (seperti nama lengkap user).
- Home directory = *path absolut* untuk home direktori dari user.
- Shell = program yang otomatis dijalankan bila user login. Beberapa *command interpreter* (biasanya disebut shell).

Contoh isi file `/etc/passwd` :

```
root : x :0: 0 :root : / root : /bin/bash
```

Baris diatas menunjukkan root user mempunyai shadow password, UID dan GID nya 0. User root mempunyai home direktori `/root/` dan menggunakan shell `/bin/bash`

2. File `/etc/group`

File `/etc/group` adalah file yang berisi daftar group yang dipisahkan per baris. Setiap baris terdiri dari 4 kolom, yang berisi informasi mengenai

- Group name = nama group.
- Group password = bila di set, mengijinkan user yang bukan bagian dari group bergabung ke dalam group dengan menggunakan perintah `nergrp` dan mengetikkan password. Jika lebih kecil dari x, maka shadow group password digunakan.
- Group ID (GID) = bilangan numerik yang ekuivalen dengan group name.
- Member list = daftar user yang menjadi milik group.

Contoh baris pada file `/etc/group`

```
general : x :502 : juan, shelly, augustha
```

Baris diatas menunjukkan, group `general` menggunakan password *shadow*, mempunyai GID 502 dan anggota `juan`, `shelly`, `augustha`.

3. Aplikasi User Account dan Group

Terdapat dua tipe dasar aplikasi yang digunakan untuk mengatur *user account* dan group pada sistem Linux Red Hat, yaitu :

- Aplikasi Graphical User Manager
- Perintah pada virtual console.

Baik aplikasi User Manager dan utilitas perintah membentuk task yang sama, perintah mempunyai kelebihan dalam skrip lebih mudah diotomatisasi.

Tabel berikut berisi beberapa perintah yang umum untuk membuat dan mengatur user command dan group.

Aplikasi	Fungsi
/usr/sbin/useradd	Menambahkan <i>user account</i> . Digunakan untuk keanggotaan <i>primer</i> dan <i>sekunder</i> .
/usr/sbin/userdel	Menghapus <i>user account</i> .
/usr/sbin/usermod	Mengedit attribut account termasuk beberapa fungsi yang berhubungan dengan masa berlaku password. Juga digunakan untuk menentukan keanggotaan <i>primer</i> dan <i>sekunder</i> .
Passwd	Melakukan setting password. Selain untuk mengubah password user juga untuk mengontrol semua aspek tentang masa berlaku password.
/usr/sbin/chpasswd	Membaca file yang berisi username dan password dan update setiap password user.
change	Mengubah masa berlaku password user. Perintah passwd juga digunakan untuk tujuan ini.
chfn	Mengubah informasi GECOS user.
Chsh	Mengubah shell default user.

Tabel berikut berisi beberapa perintah untuk membuat dan mengatur group:

Aplikasi	Fungsi
/usr/sbin/groupadd	Menambahkan group tetapi tidak menentukan user pada group tersebut. Perintah useradd dan usermod digunakan untuk menentukan user

/usr/sbin/groupdel	pada group yang ada. Menghapus group.
/usr/sbin/groupmod	Memodifikasi nama group atau GID, tetapi tidak mengubah keanggotaan group. Perintah useradd dan usermod menentukan user pada group yang ada.
Gpasswd	Mengubah keanggotaan group dan melakukan setting password untuk mengizinkan anggota selain group tersebut, yang mengetahui password group untuk bergabung. Juga digunakan untuk menentukan administrator group.
/usr/sbin/grpck	Memeriksa integritas file /etc/group dan /etc/gshadow.

Tugas Pendahuluan :

Sebagai tugas pendahuluan, kerjakan perintah dan pertanyaan dibawah ini :

1. Jelaskan isi dari satu baris pada file didalam /etc/passwd!
2. Jelaskan isi dari satu baris pada file didalam /etc/group!
3. Sebutkan perintah yang digunakan untuk membuat user baru dan berikan contohnya!
4. Sebutkan perintah yang digunakan untuk membuat group baru dan berikan contohnya!
5. Sebutkan perintah yang digunakan untuk menghapus user dan group. Berikan contohnya!
6. Apa yang dimaksud dengan perintah usermod dan groupmod?

Percobaan :

1. Login sebagai root.
2. Bukalah Console Terminal dan lakukan percobaan di bawah ini kemudian analisa hasil percobaannya.
3. Selesaikan soal-soal latihan.

Percobaan 1: Melihat file /etc/passwd dan/etc/group.

1. Lihatlah isi file /etc/passwd dan sebutkan kolom apa saja yang terdapat pada setiap baris.
`# cat /etc/passwd | more`
2. Lihatlah isi file /etc/group dan sebutkan kolom apa saja yang terdapat pada setiap baris.
`# cat /etc/group | more`

Percobaan 2 : Menambah Group dan User.

1. Buatlah 3 group baru dengan perintah groupadd.Perhatikan informasi group user baru pada file /etc/group.
`# groupadd dosen`
`# groupadd mahasiswa`
`# groupadd tamu`
`# cat /etc/group`

Percobaan 3 : Menambah User.

1. Buatlah user baru dengan perintah useradd. Perhatikan perubahan isi file /etc/passwd setelah pembuatan user baru. Juga perhatikan apakah home direktori setiap user juga dibuat saat pembuatan user baru.

```
# useradd -g dosen agung
```

```
# grep agung /etc/passwd
```

```
# useradd agus
```

```
# passwd agus
```

```
# grep agus /etc/passwd
```

```
# ls -l /home
```

2. Opsi -g pada perintah useradd untuk menentukan group dari user yang dibuat.

```
# useradd -g tamu asep
```

```
# ls -l /home
```

Percobaan 4 : Memodifikasi Group Dari User.

1. Dengan perintah usermod, modifikasi group dari setiap user merupakan milik suatu group primer dan kemungkinan juga bagian dari group lain (*supplementary group*). Untuk memodifikasi group dari suatu user dapat digunakan perintah usermod.

```
# usermod -g mahasiswa -G dosen,tamu agung
```

```
# usermod -g dosen -G mahasiswa agus
```

Percobaan 5 : Melihat Group Dari User.

1. Lihat group dari seorang user dengan perintah groups.

```
# groups agung
```

```
# groups agus
```

```
# groups asep
```

Percobaan 6 : Mengubah Password User

1. Root dapat mengubah password dan user.
passwd agung
2. Password yang diubah dengan perintah usermod merupakan file *enkripsi* sehingga tidak dapat digunakan sebagai password pada saat login.
useradd -g dosen agatha
useradd -p agatha agatha
3. Cobalah login sebagai user agatha, apakah anda dapat login?
4. Cobalah mengubah password user dengan login pada user yang bersangkutan .Login sebagai user, dan ubahlah password user.
\$ passwd

Percobaan 7 : Menghapus User.

1. Hapus user dengan menggunakan perintah userdel.Opsi -r untuk menghapus seluruh isi home direktori.
userdel -r agung
userdel -r agus
userdel -r asep
userdel -r agatha

Percobaan 8 : Menghapus Group.

1. Hapus group dengan menggunakan perintah groupdel.
groupdel dosen
groupdel mahasiswa
groupdel tamu

Percobaan 9 : Menghapus Home Direktori.

1. Hapus home direktori.

```
# rmdir /home/agung
```

```
# rmdir /home/agus
```

```
# rmdir /home/asep
```

```
# rmdir /home/agatha
```

Latihan :

1. Buatlah tiga group “parent”, “children” dan “soho”. Perhatikan anggota dari setiap group berikut :

parent	children	soho
paul	alice	accounts
jane	derek	sales

2. Buatlah user account untuk setiap anggota group sesuai tabel diatas!
3. Cek apakah home direktori yang terbentuk sesuai dengan tabel diatas!
4. Ubahlah password paul dan derek melalui root!
5. Cobalah mengubah password alice dengan login sebagai alice!
6. Lihat keanggotaan setiap user!
7. Hapuslah user account dan sales!

Laporan Resmi :

1. Analisa hasil percobaan yang anda lakukan!
2. Kerjakan latihan diatas dan analisa hasil tampilannya!
3. Berikan kesimpulan dari praktikum ini!

PERTEMUAN 11

ORGANISASI FILE DAN STRUKTUR DIREKTORI.

1. Tujuan

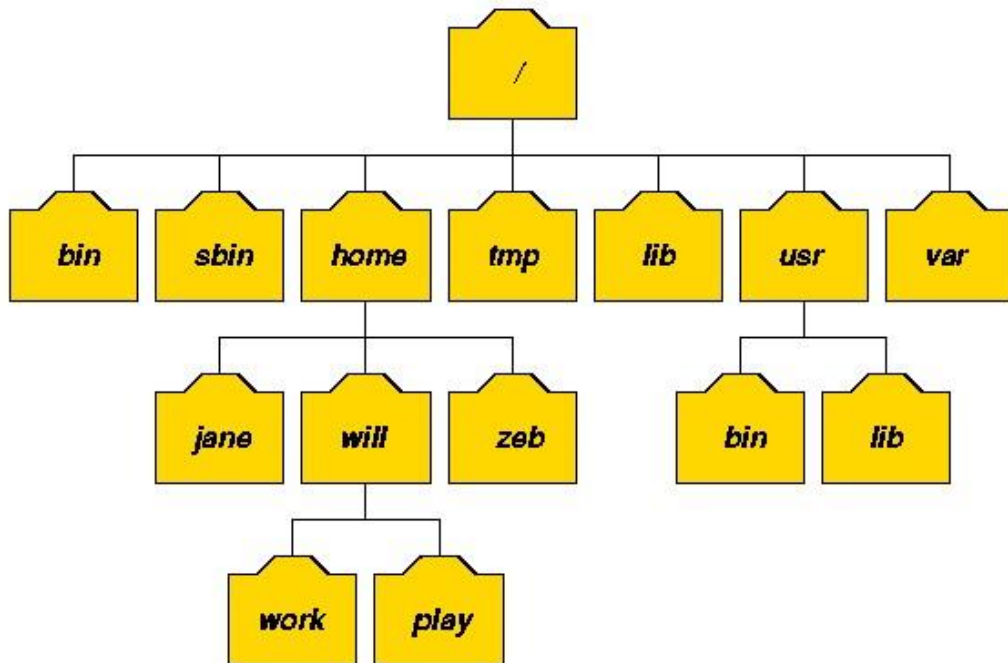
Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu :

- ✓ Membuat direktori.
- ✓ Memanipulasi direktori.
- ✓ Mengetahui operasi file atau organisasi file pada sistem operasi Linux.
- ✓ Mengetahui konsep link dan symbolic link

2. Dasar Teori

1. Organisasi File

Sistem file pada Linux menyerupai pohon (tree), yaitu dimulai dari root, kemudian direktori dan sub direktori. Sistem file pada Linux diatur secara hirarkhikal, yaitu dimulai dari root dengan simbol “/” seperti pada gambar 8.1. Kita dapat menciptakan file dan direktori mulai dari root ke bawah. Direktori adalah file khusus, yang berisi nama file dan INODE (pointer yang menunjuk ke data / isi file tersebut). Secara logika, direktori dapat berisi file dan direktori lagi disebut sub direktori.



Gambar 8.1

2. Direktori Standard pada Sistem Operasi Linux

Setelah proses instalasi sistem operasi selesai tanpa adanya corrupt, maka sistem operasi linux akan menciptakan sistem file yang baku, yang terdiri dari /bin, /dev, /bin, /sbin dan lain sebagainya seperti terlihat pada tabel 8.1 berikut ini.

Table 8.1

Direktori	Deskripsi
/etc	Berisi file administratif (konfigurasi dll) dan file executable atau script yang berguna untuk administrasi system.
/dev	Berisi file khusus yang merepresentasikan peralatan hardware seperti memori, disk, printer, tape, floppy dll.
/bin	Berisi utilitas sistem level rendah (binary).
/sbin	Berisi utilitas sistem untuk superuser (untuk membentuk administrasi sistem).

/usr/sbin /usr/bin	Berisi utulitas sistem dan program aplikasi level tinggi.
/usr/lib	Berisi program library yang diperlukan untuk kompilasi.
/tmp	Berisi file sementara, yang pada saat Bootstrap akan dihapus (dapat digunakan oleh sembarang user).
/boot	Berisi file yang sangat penting untuk proses Bootstrap. Kernel vmlinuz disimpan di direktori ini.
/proc	Berisi informasi tentang kernel Linux, proses dan virtual sistem file.
/var	Direktori variable, artinya tempat penyimpanan LOG (catatan hasil output program), file ini dapat membengkak dan perlu dimonitor perkembangannya.s
/home	Berisi direktori untuk pemakai Linux 9pada SCO diletakkan pada /usr).
/mnt	Direktori untuk mounting sistem file.
/root	Home direktori untuk superuser (root).
/usr/bin/X11	Symbolic link ke /usr/X11R6/bin, program untuk X-Window.
/usr/src	Source code untuk Linux.
/opt	Option, direktori ini biasanya berisi aplikasi tambahan (“add-on”) seperti Netscape Navigator, kde, gnome,applix dll.

3. Direktori /etc

Berisi file yang berhubungan dengan administrasi sistem, maintenance script, konfigurasi, security dll. Hanya superuser (root) yang boleh memodifikasi file yang berada di direktori ini. Sub direktori yang sering diakses pada direktori /etc, antara lain :

- ppp, point to point protocol (untuk koneksi ke internet).
- rc.d / init.d (inisialisasi (startup) dan terminasi (shutdown) proses di Linux dengan konsep runlevel.

- cron.d (rincian proses yang dieksekusi dengan menggunakan jadwal (timedependent proses)).
- FILES (file security dan konfigurasi meliputi : password, host, shadow, ftpaccess, inetd.conf, lilo.conf, motd, printcap, profile, resolv.conf, sendmail.cf, syslog.conf, dhcp.conf, smb.conf, fstab).

4. Direktori /dev

Konsep Unix dan Linux adalah memperlakukan peralatan hardware sama seperti penanganan file. Setiap alat mempunyai nama file yang disimpan pada direktori /dev.

Table 8.2

Peralatan	Direktori
Floppy	/dev/fd0
Harddisk	IDE : /dev/had, /dev/hdb, /dev/hdc, dev/hdd SCSI : /dev/sda, /dev/sdb, /dev/sdc
CDROM	SCSI : /dev/scd0, /dev/scd1 IDE : /dev/gscd, /dev/sonycd Universal : /dev/cdrom (link dari actual cdrom ide atau scsi)
Mouse	PS2 : /dev/lp0 Universal : /dev/mouse
Parallel Port	LPT1 : /dev/lp0 LPT2 : /dev/lp1
Serial Port	COM1 : /dev/ttyS0 COM2 : /dev/ttyS1 Universal : /dev/modem (link dari S0 atau S1)

5. Direktori /proc

Direktori /proc adalah direktori yang dibuat diatas Random Access Memory (RAM) dengan sistem file yang diatur oleh kernel. /proc berisi nomor proses dari sistem dan nama driver yang aktif di sistem. Semua direktori berukuran 0 (kosong)

kecuali file kcore dan self. Setiap nomor yang ada pada direktori tersebut merepresentasikan process ID (PID).

6. Tipe File pada Sistem Operasi Linux

Pada sistem operasi Linux terdapat beberapa tipe file antara lain :

- Ordinary file
- Direktori
- Block Device (Peralatan I/O) merupakan representasi dari peralatan hardware yang menggunakan transmisi data per block (misalnya 1 KB block), seperti disk, floppy tape.
- Character Device (Peralatan I/O) merupakan representasi dari peralatan hardware yang menggunakan transmisi data karakter per karakter, seperti terminal, modem, plotter dll.
- Named Pipe (FIFO) file yang digunakan secara intern oleh sistem operasi untuk berkomunikasi antar proses.
- Link file

7. Properti File pada Sistem Operasi Linux

Pada sistem operasi Linux mempunyai beberapa atribut, antara lain berupa tipe file. Tipe file akan menentukan tipe dari suatu file, seperti pada tabel berikut ini.

Table 8.3

Karakter	Arti
-	File biasa.
D	Direktori.
	Symbolic link.
B	Block special file.
C	Character special file.
S	Socket link.
P	FIFO

8. Ijin Akses

Ijin akses akan menentukan hak user terhadap file yang bersangkutan.

- Jumlah link (jumlah link akan menentukan jumlah link yang terdapat pada suatu file).
- Pemilik (menunjukkan siapa pemilik suatu file).
- Jumlah karakter / ukuran file (menentukan ukuran file dalam ukuran byte).
- Waktu pembuatan terakhir (menunjukkan bahwa kapan file tersebut dibuat atau dimodifikasi).
- Nama file (menunjukkan nama suatu file).

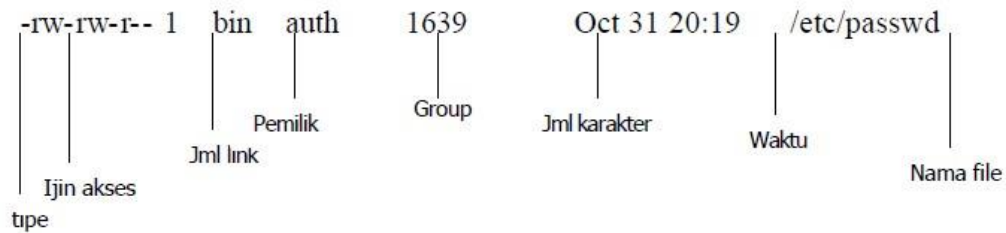
Contoh file dalam sistem operasi Linux dengan mode text ditunjukkan seperti gambar 8.2 dan gambar 8.3 berikut ini.

```
drwxr-xr-x 2 root root 4096 Dec 14 19:40 azka
drwxr-xr-x 2 root root 4096 Dec 14 19:40 azzam
-rw-r--r-- 1 root root    0 Dec 14 11:20 belajar_linux
drwxr-xr-x 2 root root 4096 Dec 10 15:55 yamta
```

Gambar 13.2. Contoh file dalam sistem operasi Linux mode Text

```
drwxr-xr-x 2 yamta yamta 4096 Dec 11 15:47 azzam
-rw-r--r-- 1 root root    0 Dec 10 22:04 belajar_linux
drwxr-xr-x 2 yamta yamta 4096 Dec 10 19:29 Desktop
drwxr-xr-x 2 yamta yamta 4096 Dec 10 19:29 Documents
drwxr-xr-x 2 yamta yamta 4096 Dec 10 19:29 Downloads
drwxr-xr-x 2 yamta yamta 4096 Dec 10 19:29 Music
drwxr-xr-x 2 yamta yamta 4096 Dec 10 19:29 Pictures
-rw-r--r-- 1 root root  121 Dec 10 22:38 ping.sh
drwxr-xr-x 2 yamta yamta 4096 Dec 10 19:29 Public
d-wx--x--x 2 root root 4096 Dec 11 12:20 samba
drwxrwxrwx 2 root root 4096 Dec 11 13:09 share
drwxr-xr-x 2 yamta yamta 4096 Dec 10 19:29 Templates
drwxr-xr-x 2 yamta yamta 4096 Dec 10 19:29 Videos
```

Gambar 8.2



Gambar 8.3

9. Nama File

Nama file maksimal terdiri dari 255 karakter berupa alfanumerik dan beberapa karakter spesial yaitu garis bawah, titik, koma dan lainnya kecuali spasi dan karakter "&", ";", "|", "?", "'", "\"", ":", "[", "]", "(", ")", "\$", "<", ">", "{", "}", "^", "#", "\", "/". Linux membedakan huruf kecil dengan huruf besar (casesensitive). Contoh nama file yang benar :

Abcde5434

3

prog.txt

PROG.txt

Prog.txt,old

report_101,v2.0.1

5-01.web.html

10. Symbolic Link

Link adalah sebuah teknik untuk memberikan lebih dari satu nama file dengan data yang sama. Bila file asli dihapus, maka data yang baru juga terhapus .
Format dari Link :

ln fileAsli fileDuplikat

fileDuplikat disebut *hard link* dimana kedua file akan muncul identik (*link count* = 2) Bila fileAsli atau ileDuplikat diubah perubahan akan terjadi pada file lainna.

Symbolic Link diperlukan bila file tersebut di “link” dengan direktori /file yang berada pada partisi yang berbeda. Tipe file menjadi l (link) dan file tersebut menunjuk ke tempat asal. Format :

`ln -s /FULLPATH/fileAsli /FULLPATH/fileDuplikat`

Pilihan `-s` (*shortcut*) merupakan bentuk *soft link* dimana jumlah *link count* pada file asal tidak akan berubah. Pada bentuk *soft link*, *symbolic link* dapat dilakukan pada file yang tidak ada, sedangkan pada *hard link* tidak dimungkinkan. Perbedaan lain, *symbolic link* dapat dibentuk melalui media disk atau partisi yang berbeda dengan *soft link*, tetapi pada *hard link* terbatas pada partisi disk yang sama.

11. Melihat Isi File

Untuk melihat jenis file menggunakan format :

`file filename(s)`

Isi file akan dilaporkan dengan deskripsi level tingginya seperti contoh berikut

- `$ file myprog.c letter.txt webpage.html`
- `myprog.c: C program text`
- `letter.txt: ASCII text`
- `webpage.html: HTML document text`

Perintah ini dapat digunakan secara luas untuk file yang kadang membingungkan, misalnya antara kode C++ dan Java.

12. Mencari File

Jika ingin melihat bagaimana pohon direktori dapat digunakan perintah

- `find`

Format : `find directory -name targetfile -print`

Akan melihat file yang bernama targetfile (bisa berupa karakter wildcard)

- `which`

Format : `which command`

Untuk mengetahui letak system utility

- `locate`

Format : locate string

Akan mencari file pada semua direktori dengan lebih cepat dan ditampilkan dengan path yang penuh.

13. Mencari Text pada File

Untuk mencari text pada file digunakan perintah grep (General Regular Expression Print) dengan format perintah

grep option pattern files

Grep akan mencari file yang bernama sesuai pattern yang diberikan dan akan menampilkan baris yang sesuai.

PERCOBAAN 1

1. Login sebagai user.
2. Bukalah terminal dan lakukan percobaan dibawah ini.

Perintah	Tujuan
\$ pwd	Melihat direktori Home.
\$ echo \$ Home	
\$ pwd	Melihat direktori aktual dan parent direktori.
\$ cd	
\$ mkdir	Membuat suatu direktori, lebih dari satu direktori atau sub direktori.
\$ ls -l	
\$ ls -l A/D	
\$ rmdir B	Menghapus satu atau lebih direktori hanya dapat dilakukan pada direktori kosong dan hanya dapat dihapus oleh pemiliknya kecuali bila diberikan ijin aksesnya.
\$ ls -l B	
\$ rmdir B/F B	
\$ ls -l B	Navigasi direktori dengan instruksi cd untuk pindah dari satu direktori ke direktori lain.
\$ pwd	
\$ ls -l	
\$ cd A	

\$ pwd	
\$ cd . .	
\$ pwd	
\$ cd/home/<user>/C	
\$ pwd	
\$ cd / <user>/C	
\$ pwd	

PERCOBAAN 2

Perintah	Tujuan
\$ cat > teks1 Membuat sebuah file [Ctrl -d]	Perintah cp untuk mengkopi file atau seluruh direktori.
\$ cp teks teks1	
\$ ls-l	
\$ cp contoh A	
\$ ls-l A	
\$ cp teks teks1 A/D	
\$ ls -l A/D	
\$ mv teks teks2	Perintah mv untuk memindah file.
\$ ls -l	
\$ mv teks1 teks2 A/D	
\$ ls -l A/D	
\$ mv teks teks1 C	
\$ ls -l C	
\$ rm teks2	Perintah rm untuk menghapus file.
\$ ls -l	
\$ rm -i teks	
\$ rm -rf A C	

\$ ls -l	
----------	--

PERCOBAAN 3

Perintah	Tujuan
\$ echo "Hallo apa kabar" > hallo.txt	Membuat shortcut (file link).
\$ ls -l	
\$ ln hallo.txt z	
\$ ls -l	
\$ Cat z	
\$ mkdir mydir	
\$ ln z mydir/hallo.juga	
\$ cat mydir/hallo.juga	
\$ ln -s z bye.txt	
\$ ls -l bye.txt	
\$ cat bye.txt	

PERCOBAAN 4

Perintah	Tujuan
\$ ls-l	Melihat isi file.
\$ file hallo.txt	
\$ file bye.txt	

PERCOBAAN 5

Perintah	Tujuan
\$ find /home -name "*.txt" -print > myerror.txt	Mencari file dengan perintah Find.
\$ cat myerror.txt	
\$ find . -name "*.txt" -exec wc -l '{} ';	
\$ which ls	Mencari file dengan perintah which.
\$ locate "*.txt"	Mencari file dengan perintah locate

PERCOBAAN 6

Perintah	Tujuan
\$ grep Hallo *.txt	Mencari text pada file.