

# Pertemuan 12

## ADMINISTRASI SISTEM LINUX

# 1. Penanganan Pengguna (User dan Group)

Administrasi sistem pada LINUX hanya dapat dijalankan oleh super user atau administrator sistem tersebut.

## 1. User

User hanya dapat dibuat dan dihapus oleh administrator atau super user saja. Untuk membuat atau menambah user ada dua cara, yaitu dengan *shell script (mode text)* atau *melalui GUI (X-Window)*. Untuk *shell script* dengan perintah :

- **useradd so** : digunakan untuk membuat user so, atau
- **adduser so** : untuk menambah user. bila perintah useradd tidak ditemukan
- **rmuser -r so** : untuk menghapus user so beserta home direktorinya

Untuk GUI tergantung dari distro Linux dan desktop managernya. Biasanya ada pada menu **configuration**

# Penanganan Pengguna (User dan Group) Lanjut...

## 2. Group

Sama halnya dengan pembuatan user, pembuatan group dapat dilakukan dengan dua cara, yaitu dengan *shell script* dan *GUI*. Untuk *shell script* dengan perintah :

- **groupadd stmik :**  
digunakan untuk membuat group stmik, *atau*
- **addgroup stmik:**  
untuk menambah group. bila perintah groupadd tidak ditemukan
- **groupdel/rmgroup :**  
digunakan untuk menghapus group yang sudah ada

## 2. Batasan File (File Restriction)

- Di dalam linux setiap file mempunyai batasan file. Batasan file (hak akses) meliputi hak perijinan akses (permission), kepemilikan (owner), serta group pemilik file.
- Fungsinya adalah untuk mengatur serta membatasi akses-akses terhadap file yang bersangkutan. Untuk melihat hak akses file menggunakan perintah **ls-l**, sedangkan perintah untuk menentukan hak akses adalah **chmod**.

# Batasan File (File Restriction) Lanjut....

Dalam batasan file ada beberapa ketentuan pengaturan untuk perijinan dan kepemilikan suatu file, yaitu :

- Setiap file selalu dimiliki oleh pemilik yang berhak mengatur hak akses terhadap file tersebut, hal ini berlaku juga untuk **root /administrator** atau **super user / subtitude user**
- Hak akses file dibagi menjadi tiga :
  - a. read (baca) hak akses ini disimbolkan dengan huruf '**r**',
  - b. write (tuliskan) hak akses ini disimbolkan dengan huruf '**w**',
  - c. execute (eksekusi) hak akses ini disimbolkan dengan huruf '**x**'

# Batasan File (File Restriction) Lanjut....

Pemilik file dibagi menjadi tiga :

- user (pemilik file)
- group (group dari memiliki file), group di sini merupakan group dari pemilik file.
- other (selain user dan group), adalah **user** yang tidak berada dalam satu group dengan **user pemilik file**

contoh :

```
# ls -l /home/data  
-rwxr-xr-x 1      so  stmik  5  Mar 25 2004  /home/data
```

Pada kolom pertama terdapat hak perijinan akses dari file **/home/data**. File 'data' dimiliki oleh 'so' dan dimiliki oleh group 'stmik', dengan ukuran file 5 byte, serta tanggal pembuatan/editing terakhir.

# Batasan File (File Restriction) Lanjut....

Perhatikan untuk kolom pertamanya '-rwx-r-xr-x' (dibaca dari kiri ke kanan).

-	R	w	x	r	-	x	r	-	x
1	2	3	4	5	6	7	8	9	10

Terlihat ada 10 karakter untuk hak akses. Maksud dari tampilan tersebut adalah

- Nomor 1 menunjukkan bahwa data merupakan file (- → file biasa, d → direktori, l → link ).
- rwx (nomor 2,3,4) adalah hak akses untuk pemilik file / user (read/baca, write/tulis, execute/eksekusi).

# Batasan File (File Restriction) Lanjut....

- r-x (nomor 5,6,7) adalah hak akses untuk group dari pemilik file (read/baca, execute/eksekusi) **bukan group yang memiliki file (pada** contoh adalah group 'stmik').
- r-x (nomor 8,9,10) adalah hak akses untuk other (user-user selain pemilik file dan groupnya)(read/baca, execute/eksekusi).

Pemilik file atau super user / administrator dapat mengubah hak akses terhadap file-filenya.



# Pengaturan hak akses terhadap suatu file

## Metode Huruf

- Pada metode huruf ini sintaknya diwakili oleh huruf-huruf yang menerangkan tentang hak akses dan pemilik file tersebut. Huruf-huruf yang dipakai adalah :

r = menyimbolkan read / baca

w = menyimbolkan write / tulis

x = menyimbolkan execute / eksekusi

u = menyimbolkan user / pemilik file

g = menyimbolkan group

o = menyimbolkan other / user lain selain pemilik file dan group

# Pengaturan hak akses terhadap suatu file (Lanjut.....)

## Sintak :

- **chmod [ugo]+/-[rwx] nama\_file**

Dengan tanda '+' digunakan untuk memberi hak akses. tanda '-' digunakan untuk menghilangkan hak akses.

## contoh :

- **chmod u+rw aku :**  
memberi hak akses kepada user / pemilik file dengan hak akses read / baca dan write / tulis pada file stmik.
- **chmod go-w aku :**  
menghilangkan hak akses write / tulis untuk group dan other pada file oman
- **chmod ugo+w aku :**  
menambahkan hak akses write / tulis untuk user, group dan other pada file oman,

# Pengaturan hak akses terhadap suatu file (Lanjut.....)

- **Metode Angka**

Selain menggunakan metode huruf, untuk mengatur hak akses dapat menggunakan metode angka.

Angka-angka yang digunakan merupakan hasil dari konversi dari bilangan biner ke desimal, yaitu :

- 4 mewakili hak akses read / baca ( r-- ) --> 100 (biner) = 4
- 2 mewakili hak akses write / tulis ( -w- ) --> 010 = 2
- 1 mewakili hak akses execute/ eksekusi( --x ) --> 001 = 1
- 0 menghilangkan semua hak akses ( --- ) --> 000 = 0

# Pengaturan hak akses terhadap suatu file (Lanjut.....)

## Contoh :

- **chmod 651 saya**

Hak akses untuk user adalah read dan execute (6) hak akses untuk group adalah write dan execute (5) hak akses untuk other adalah execute (1)

- **chmod 740 kamu**

hak akses untuk user adalah read, write dan execute (7), (*full akses*).hak akses untuk group adalah write. hak akses untuk other tidak ada (0)

### 3. Kepemilikan File (File Ownership)

Hak kepemilikan dari sebuah file atau direktori dapat diubah-ubah. Baik mengubah kepemilikan dari seorang user ke user lain atau dari suatu group ke group lain.

Ada yang perlu diperhatikan dalam merubah kepemilikan file:

- Hanya super user / administrator yang dapat merubah hak kepemilikan sebuah file
- Dalam hal pemindahan kepemilikan dari suatu group ke group lain yang perlu diperhatikan adalah group yang memiliki file, **bukan *group dari user si pemilik file*** .

# Kepemilikan File (File Ownership)

## Lanjut.....

Untuk mengubah kepemilikan file menggunakan perintah **chown** (untuk merubah kepemilikan file dari seorang user ke user lain) dan **chgrp** (untuk merubah kepemilikan file dari satu group ke group lain), sintaknya adalah :

**chown owner nama\_file**

Dengan owner adalah nama user lain yang akan diberi file

**contoh :**

- **chown mahasiswa data** --> akan menjadikan file data menjadi milik user mahasiswa
- **chgrp group nama\_file**

Dengan group adalah nama group lain yang akan diberi file

**contoh :**

- **chgrp amikom data** --> akan menjadikan file data menjadi milik dari group amikom

## 4. Manajemen Proses

Dalam Linux/UNIX Untuk melihat proses dan penanganan proses-proses yaitu program konsole '**top**' dan '**ps**' dan untuk memamatkannya atau memanipulasinya menggunakan perintah seperti '**kill**' dan '**nice**'.

Semua informasi mengenai proses yang sedang berlangsung / berjalan disimpan pada direktori '**/proc**' yang berubah secara real time. Dengan membaca file-file yang ada di direktori '**/proc**' akan sangat merepotkan, ada dua perintah yang dapat melihat proses yang sedang berlangsung .

# PS

Perintah **ps** digunakan untuk melihat proses yang sedang berlangsung, dengan menggunakan **ps** dapat dilihat informasi proses yang sedang berlangsung.

contoh tampilan **ps**

- **PID TTY TIME CMD**
- **235 tty1 00:00:00 bash**
- **780 tty1 00:00:00 ps**

Tampilan di atas adalah untuk menampilkan informasi proses itu sendiri. karena menjalankan **ps** di baris perintah yang menginduk pada shell command line (program yang menerjemahkan perintah, biasanya **bash**).



# PS (Lanjut....)

**Ps** sendiri mempunyai banyak option contoh :

- **ps aux**

Keterangan tentang field/kolom untuk perintah di atas

**USER:** adalah nama user yang menjalankan proses yang bersangkutan atau atas ijin siapa proses tersebut berjalan. sebagian proses yang dijalankan oleh root merupakan proses boot yang dijalankan oleh init (central dari semua proses, dapat dilihat dengan perintah 'pstree', init mengendalikan proses mana yang akan dijalankan dan dihentikan)

**PID:** adalah id dari proses

**TTY:** adalah terminal/konsol (teletype) dari mana program tersebut dijalankan, semua proses yang dijalankan oleh init tidak mempunyai aturan terminal ini.

# PS (Lanjut....)

**STAT:** status proses sat ini. keterangannya adalah

S (sleeping) → proses dalam keadaan sleep //kurang lebih 20 menit

R (running) → proses sedang berjalan

D (defunct) → proses tidak dipakai, merupakan proses yang berjalan sembarangan, kill secepat mungkin

I (idle) → proses sedang tidak dijalankan/tidak dipakai

Z (zombie) → proses yang sudah mati, akan hilang saat di shutdown berikutnya, tidak berpengaruh pada sistem

T (terminate) → proses dihentikan

# PS (Lanjut....)

- TIME** menunjukkan cpu-time yang dihabiskan (akumulasi waktu yang digunakan proses), bukan waktu proses dijalankan
- %CPU** persentasi dari waktu CPU yang sudah digunakan, untuk prosesor modern proses akan berisi nol, kecuali untuk keperluan X-Window
- %MEM** persentasi memori sistem yang digunakan oleh proses, nilai berhubungan dengan RSS, bukan VSZ
- VSZ** ukuran memori virtual proses. besar memori proses termasuk shared libraries
- RSS** (resident set size) ukuran sebenarnya. biasanya nilainya lebih kecil dari nilai yang diberikan 'top'.karena berbeda dalam hitungannya
- CMD/COMMAND**  
adalah nama program / perintah yang digunakan

# TOP

- **top**

Perintah **top** akan memberikan tampilan secara dinamis mengenai status proses yang sedang berlangsung / berjalan. tampilan tabel proses yang ada akan di *update pada interval waktu tertentu (default 5 detik)*. *option penting* pada '**top**' adalah **-d [detik]** untuk menentukan interval update dalam detik.

contoh :

- **top -d 2**

ada beberapa field/kolom baru diantaranya

PRI    Prioritas dari proses. waktu penghitungan maksimum dalam milidetik untuk proses ini

NI     Nilai 'nice' nilai prioritas yang diberikan secara manual.

# NICE

- **nice**

Semua proses memiliki hak yang sama dalam pembagian sumber daya. Namun dapat diubah dengan perintah **nice** untuk memulai proses dengan prioritas yang diberikan. sintak :

**nice -n [value] [process]**

keterangan :

- Nilai negatif menambah prioritas hanya bisa dilakukan oleh 'root' namun nilai positif dapat dilakukan oleh siapa saja. Untuk merubah prioritasnya lagi dengan perintah **renice**.

sintak :

**renice [prioritas] [pid]**

# KILL

- **kill**

Kadang ada proses yang tidak diperlukan kehadirannya dalam sistem atau terlalu banyak memakan resource komputer, sehingga akan menurunkan kinerjanya. cara untuk menghentikan proses tersebut adalah dengan menggunakan perintah kill.

sintak :

**kill -[signal] PID**

- Menghentikan proses berdasarkan nomor ID proses, signal adalah nomorsignal yang dapat digunakan, selain nomor signal dapat juga dengan menggunakan nama signalnya, misal untuk nomor signal **9** dapat diganti dengan **KILL**

# KILL (Lanjut...)

- **killall -[signal] nama\_proses**

Menghentikan proses berdasarkan nama prosesnya. contoh:

**kill -9 253** → menghentikan proses dengan PID 253

**kill -KILL 253** → mematikan proses dengan PID 253

**killall httpd** → menghentikan proses dengan nama httpd

## Keterangan

- Tanpa menyebutkan signal secara default akan diberikan signal **15 (sigterm)** yang akan menutup program “menunjukkan jalan keluar bagi program”.

# KILL (Lanjut...)

**9 (sigkill)** akan mematikan program / mengeluarkan program dari sistem secara paksa (membasmi program) beberapa signal yang digunakan adalah:

- 1 HUP Hangup : menggantungkan proses
- 2 INT Interrupt : mereboot program / proses
- 3 QUIT Quit : menutup program
- 6 ABRT abort : membatalkan proses
- 9 KILL Kill : mematikan proses (dengan paksa)
- 14 ALRM Alarm clock
- 15 TERM terminate : mengakhiri program



# 5. Manajemen Disk

- Pada sistem operasi *under UNIX* seperti *LINUX* tidak mengenal adanya drive **C:\** atau **drive A:\**, karena pada *LINUX* menggunakan sistem hirarki dan penyatuan (direktori di dalam direktori) dan memperlakukan file, direktori dan device driver (termasuk floppy dan cdrom) sebagai file.
- Sedangkan device driver untuk disk drive ditempatkan pada direktori **/dev** (direktori tempat semua device). Dan disk drive yang belum dikenal tersebut harus dikenalkan / dimuatkan / dipasangkan pada suatu direktori agar bisa dibaca / digunakan.

# Manajemen Disk (Lanjut..)

Ada beberapa tipe file sistem yang sering digunakan dan perlu anda perhatikan untuk mengenalkan device yang akan digunakan :

- msdos** → untuk mengenalkan file-file dari DOS dan Windows
- vfat** → juga untuk mengenalkan file-file dari DOS dan Windows (FAT 16 dan FAT 32)
- ext2** → untuk mengenalkan file-file dari Linux
- ntfs** → untuk file sistem dari NT
- ufs** → untuk mengenalkan file-file dari UNIX

# Manajemen Disk (Lanjut..)

untuk mengenalkannya menggunakan perintah **mount**. sintak standarnya adalah :

- **mount -t [fstype] [device] [dir]**

keterangan :

- mount -t** ➔ adalah perintah untuk mengenalkan sebuah file sistem kedalam Linux
- fstype** ➔ adalah tipe dari file sistem pada device yang akan dipakai
- device** ➔ adalah direktori tempat device driver (driver dari alat yang digunakan) berada, terletak pada direktori /dev
- dir** ➔ adalah direktori yang digunakan untuk menampung file-file pada device yang akan digunakan

Setelah melakukan **mount** (apabila sudah selesai dipakai) sebaiknya device yang telah di **mount** harus dinonaktifkan (**di umount**).

sintaknya :

- **umount [device]**

# Harddisk

Untuk mengenalkan media harddisk prinsipnya sama dengan floppy. Untuk mengetahui partisi-partisi yang ada dalam harddisk dapat menggunakan perintah **df**. Untuk mengenalkan media harddisk

contoh :

- **mount -t vfat /dev/hda1 /mnt/windows**

keterangan :

- File sistem yang akan dikenalkan berasal dari DOS atau Windows, dan device/alat yang digunakan adalah harddisk primary . Dan hasil dari pengenalan file sistem tersebut akan ditempatkan pada direktori **/mnt/windows**. Selain menggunakan tipe file sistem **msdos** bisa juga menggunakan tipe **vfat** untuk tipe file sistem dari Windows.

# Harddisk (Lanjut...)

- Beberapa tipe file system seperti NTFS, untuk beberapa distro terkadang tidak dapat dikenali (seperti distro Redhat dan Fedora). Agar tetap dapat dikenali, user harus menambahkan modul untuk mendeteksi file system NTFS pada distro tersebut.
- Beberapa distro lain yang dapat mengenali file system NTFS terkadang hanya dapat membaca file-file dan direktori pada partisi hardisk dengan format file system NTFS, tetapi tidak dapat diedit, copy atau modifikasi lainnya..

# CDROM

Untuk mengenalkan CDROM prinsip sama saja dengan dua media di atas, hanya saja untuk tipe file sistemnya tidak perlu dituliskan.

contoh :

- **mount /dev/cdrom /mnt/cdrom**

keterangan :

**/cdrom** adalah device untuk cdrom yang akan dikenalkan.

**/mnt/cdrom** adalah direktori tempat menampung file dari cdrom

# Flash Disk

Untuk mengenalkan flashdisk prinsip sama saja dengan media CDROM di atas, untuk tipe file sistemnya tidak perlu dituliskan

contoh :

- **mount /dev/sda1 /mnt/flash**

keterangan :

- **/sda1** adalah device untuk flashdisk dari **usb1** yang akan dikenali oleh linux Slackware, distro lain mungkin saja berbeda, seperti **/dev/sd0** dan **/mnt/flash** adalah direktori tempat menampung file dari cdrom.
- Ingat direktori flash sebelumnya harus dibuat terlebih dahulu, baru proses mounting device dilakukan