**Percobaan 1 : Izin Akses**

1. Melihat identitas melalui etc/passwd atau etc/group $id

$grep **[nama\_user]** /etc/passwd

$grep **[nomor\_group\_id]** /etc/group

1. Memeriksa direktori home

$ls -ld /home/nama\_user

1. Mengubah izin akses (chmod)

$touch f1 f2 f3

**$ls -l**

**Hasil :**

Menampilkan isi direktori home yang sudah ada + tiga file tambahan baru, yaitu f1, f2, dan f3 dimana ketiga file tersebut memiliki izin akses user = read-write; group = read-write; dan others = read.

$chmod u+x f1

$chmod g=w f1

$chmod o-r f1

**$ls -l**

**Hasil :**

Izin akses user, group, dan others pada file f1 berubah menjadi user = read-write-execute; group = write; dan others = – .

$chmod a=x f2

$chmod u+x,g-r,o=w f3

**$ls -l**

**Hasil :**

Izin akses user, group, dan others pada file f2 berubah menjadi execute sedangkan izin akses pada file f3 berubah menjadi user = read-write-execute; group = write; dan others = write.

$chmod 751 f1

$chmod 624 f2

$chmod 430 f3

$ls -l

**Hasil :**

* Izin akses pada file f1 :

Sebelum : rwx -x- ---

chmod 751 dimana 7 = 111; 5 = 101; 1 = 001

Sesudah : rwx r-x --x

* Izin akses pada file f2 :

Sebelum :--x --x --x

chmod 624 dimana 6 = 110; 2 = 010; 4 = 100

Sesudah : rw- -w- r--

* Izin akses pada file f3 :

Sebelum : rwx -w- -w-

chmod 430 dimana 4 = 100; 3 = 011; 0 = 000

Sesudah : r-- -wx ---

1. Menguji izin akses (chmod)

$echo "belajar chmod" >> f1

$echo "belajar chmod" >> f2

$echo "belajar chmod" >> f3

**1# Kesimpulan :**

Ketiga file, yaitu f1, f2, dan f3 berisi teks “belajar chmod” dimana izin akses user pada file f1 adalah dapat melakukan read, write, dan execute. Izin akses user pada file f2 adalah dapat melakukan read dan write, sedangkan izin akses user pada file f3 hanyalah read only. Apabila ketiga file dipanggil dalam terminal menggunakan perintah cat maka yang hanya muncul teksnya adalah file f1 dan f2.

1. Untuk mengganti kepemilikan digunakan perintah chown. Beberapa hal yang perlu diperhatikan adalah bahwa hanya root yang dapat menggunakan perintah chown, account biasa tidak diperbolehkan meskipun ia adalah user owner file yang ingin diubah.

$ls -l

$chown [user\_Owner\_Baru] f1

$chown [user\_Owner\_Baru:grup\_Owner\_Baru] f2

$chown [user\_Owner\_Baru:grup\_Owner\_Baru] f1 f3

$ls -l

**2# Kesimpulan :**

Setelah menambahkan user\_Owner\_Baru dan grup\_Owner\_Baru melalui sudo add[user/group] maka setelah menggunakan perintah chown kita dapat mengganti kepemilikan file f1, f2, f3 dimana kepemilikan user yang awalnya adalah sailor\_moon kini berubah menjadi milik nezuko\_chan, sedangkan kepemilikan group yang awalnya adalah sailor\_moon kini juga telah berubah menjadi milik kamado.

1. Mengubah group hanya dapat dilakukan oleh pemilik file/direktori tersebut dengan perintah chgrp.

Melihat nama grup yang terdaftar:

**$grep [nama\_group\_yang\_dicari] /etc/group**

#chgrp [group\_owner\_baru] f1

$ls -l

**Percobaan 2 : Umask *(User file-creation mode mask)***

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

$touch myfile

$ls -l myfile

1. Melihat nilai umask

$umask atau $umask -S

1. Modifikasi nilai umask

$umask 027

$umask

$touch newfile

$mkdir mydir

$ls -l

**Hasil :**

* Izin akses myfile

File biasa : 666

Umask : 002

Izin akses : 664 (110 | 110 | 100) atau rw- rw- r--

* Izin akses newfile

File biasa : 666

Umask : 027

Izin akses : 639 ≈ 640 (110 | 100 | 000) \*max biner 3 angka sampai 7

atau rw- r-- ---

Izin akses myfile, yaitu user = read-write; group = read-write; dan others = read only, sedangkan izin akes newfile, yaitu user = read-write; group = read only; dan others = - .

$umask 077

$touch xfile

$mkdir xdir

$ls -l

**Hasil :**

* Izin akses xfile

File biasa : 666

Umask : 077

Izin akses : 589 ≈ 600 (110 | 000 | 000) \*max biner 3 angka sampai 7

atau rw- --- ---

* Izin akses xdir

Direktori : 777

Umask : 077

Izin akses : 700 (111 | 000 | 000) atau rwx --- ---

Nilai izin akses xfile, yaitu user = read-write; group = –; dan others = – , sedangkan izin akses direktori xdir adalah user = read-write-execute; group = –, dan others = – .

**3#Kesimpulan :**

Umask digunakan untuk mengatur izin akses ketika suatu file atau direktori akan dibuat. Angka-angka di umask memiliki fungsi terbalik dari chmod. Untuk menghitung nilai izin akses suatu file digunakan angka 666 kemudian diminuskan dengan umask, sedangkan untuk mengitung nilai izin akses suatu direktori digunakan angka 777 yang kemudian diminuskan dengan umask.

**Percobaan 3 : Link**

1. Hard link

$echo "Hallo apa kabar?" > halo.txt

$ls -l

$ln halo.txt hardlink

$ls -l

$cat hardlink

$mkdir newdir

$ln hardlink newdir/halo.juga

$cat newdir/halo.juga

$ls -l newdir/halo.juga

**4# Analisa :**

Original file halo.txt dikenai perintah harlink sehingga menduplikatnya dalam file baru yang juga bernama ‘hardlink’. Dalam terminal diatas juga dibuat direktori baru, yaitu newdir untuk menampung file duplikat hardlink ‘halo.juga’ apabila file ‘hardlink’ dan file ‘halo.juga’ dalam direktori newdir dipanggil, keduanya sama-sama berisi teks “Halo apa kabar?” yang diambil dari original file halo.txt.

1. Soft Link (Symbolic link)

$ln -s halo.txt softlink

$ls -l

$cat softlink

**5# Analisa :**

Original file halo.txt dikenai perintah softlink sehingga menduplikasinya dalam file baru bernama ‘softlink’, apabila dipanggil softlink ini juga akan mengeluarkan isi teks “Halo apa kabar?” seperti original file halo.txt. File hasil dari perintah softlink memiliki tanda anak panah dibagian pojok kiri bawahnya.

1. Menghapus link

$rm halo.txt

$cat hardlink

$cat softlink

**6# Kesimpulan**

Pada hardlink apabila original file halo.txt dihapus maka file hardlink tidak hilang sehingga apabila dipanggil masih bisa mengeluarkan isi teks “Halo apa kabar?” dari file duplikat halo.txt, sedangkan untuk softlink apabila original filenya dihapus maka data yang ada juga ikut terhapus hal ini karena softlink merujuk pada original file, tidak seperti hardlink yang merujuk pada inode.