



VERSI 1.0

MARET, 2023

PRAKTIKUM SISTEM OPERASI

MODUL 2 - NAVIGATING FILE SYSTEM AND CONTROLLING ACCESS TO FILE

TIM PENYUSUN: - MAHAR FAIQURAHMAN, S.KOM., M.T.  
- SYAHRUL PANGESTU  
- MUH. RIDHA AGAM

PRESENTED BY: LAB. INFORMATIKA

UNIVERSITAS MUHAMMADIYAH MALANG

## SISTEM OPERASI

## CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu menjelaskan konsep Hierarki Sistem file di Linux
2. Mahasiswa mampu mengelola baris perintah di linux
3. Mahasiswa mampu membuat tautan antara file di Linux
4. Mahasiswa mampu mengelola izin default dan akses pada file dan direktori linux

## SUB CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu memberikan contoh langsung Hierarki sistem file di Linux
2. Mahasiswa mampu mempraktekkan langsung pengelolaan baris perintah di Linux
3. Mahasiswa mampu mempraktekkan izin default dan akses pada file dan direktori di linux

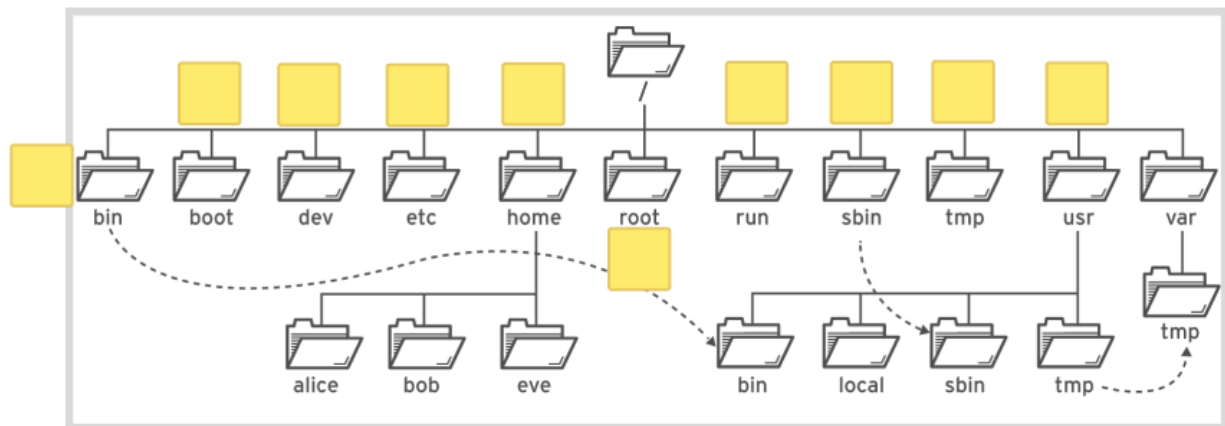
## KEBUTUHAN HARDWARE &amp; SOFTWARE

- Komputer/Laptop
- Sistem Operasi CentOS [\[link\]](#)

## MATERI POKOK

## Hierarki Sistem file

Apa itu Hierarki Sistem?



(Gambar 1. Struktur Hierarki file sistem pada CentOS)

Pada sebuah sistem, terdapat hierarki yang mana terbagi menjadi beberapa bagian dalam sebuah file sistem. Semua file di sistem Linux disimpan di sistem file, yang disusun dalam *inverted tree* direktori, yang dikenal sebagai hierarki sistem file. Pohon ini terbalik karena akar dari pohon tersebut dikatakan berada di puncak hierarki, dan cabang direktori dan subdirektori membentang di bawah akar. Sistem file sangatlah penting untuk dapat mengetahui berbagai macam pembagian file sehingga modifikasi kedepannya dapat lebih mudah.

Berikut ini adalah beberapa contoh beserta penjelasan direktori penting dalam direktori Linux:

Lokasi	Penjelasan
/bin	Merupakan direktori yang berisi file <i>executable</i> . Ciri khas dari file yang merupakan executable adalah terlihat huruf “x” di semua user yang ada. “x” merupakan singkatan dari <i>executable</i> .
/boot	Merupakan direktori yang berisikan proses booting.
/dev	Merupakan direktori yang memiliki hubungan dengan sistem yang kita gunakan. Seluruh <i>hardware</i> yang terhubung pada linux tersimpan di dalam ini.
/etc	Merupakan direktori yang memiliki konfigurasi file yang ada di sistem. Kita dapat mengkonfigurasi cronjob dan ssh dan sebagainya disini.
/home	Merupakan direktori yang menyimpan konfigurasi dari user sehingga pada saat diperiksa dapat terlihat beberapa user yang ada di dalam sistem operasi yang digunakan.
/root	Direktori yang hanya dapat diakses oleh user <i>root</i> .
/run	Merupakan penyimpanan proses aplikasi yang sedang berjalan. Kita dapat mengecek error dari suatu aplikasi dari direktori ini.
/sbin	Merupakan direktori yang seperti halnya dengan direktori bin hanya saja sbin dapat diakses oleh <i>root</i> , selain dari itu tidak dapat melakukan akses.
/tmp	Merupakan direktori yang menyimpan file secara temporari karena dalam waktu tertentu file yang ada di dalam direktori ini akan terhapus secara otomatis.
/usr	Merupakan Direktori yang berisikan aplikasi yang telah kita instalasi.
/local	Merupakan sebuah direktori yang bersifat lokal dan menyimpan seluruh software yang dikostumisasi secara lokal.

## Command Line Tools

Selain dari hirarki sistem file itu sendiri, terdapat berbagai macam *Command-Line Interface Tools (CLI)* yang digunakan di dalam sistem operasi. Kegunaan dari tiap command line ini adalah untuk dapat mempermudah dalam pembuatan, penghapusan, pengeditan, pemindahan, dan penyalinan data seluruhnya dengan menggunakan Command Line itu sendiri. dibawah ini merupakan sebuah *command commands* (perintah umum) dari *management file* itu sendiri. Sebelum melihat tabel dibawah, kita harus mengetahui beberapa kepanjangan dari syntax dibawah:

- MKDIR = MK = *Make*, DIR = *Directory* (Membuat direktori)
- CP = *Copy* (Salin)
- -R = *Recursive* (Menyalin keseluruhan isi direktori)

- MV = *Move* (Pemindahan File)
- RM = *Remove* (Menghapus file)
- RMDIR = RM = Remove, DIR = Directory (Menghapus direktori)

Aktifitas	Command Syntax
Membuat Direktori	<code>mkdir {NAMA_FILE}</code>
Penyalinan file	<code>cp {NAMA_FILE_LAMA} {NAMA_FILE_BARU}</code>
Penyalinan direktori bersama konten didalamnya	<code>cp -r {DIREKTORI_LAMA} {DIREKTORI_BARU}</code>
Pemindahan / penggantian nama dari sebuah file atau direktori	<code>mv {NAMA_FILE/DIREKTORI} {NAMA_TEMPAT/FILE_BARU}</code>
Penghapusan file	<code>rm {NAMA_FILE}</code>
Penghapusan direktori beserta konten didalamnya	<code>rm -r {NAMA_DIREKTORI}</code>
Penghapusan direktori kosong ( <i>empty</i> )	<code>rmdir {NAMA_DIREKTORI}</code>

## Links antara File

Dalam sebuah file pastinya sudah dimulai dengan satu buah *hard link* yang merupakan inisiasi nama kepada data pada sistem file. Pada saat membuat sebuah *hard link* pada file yang baru, terdapat nama lain yang menuju kepada data yang sama. *hard link* yang baru ini bekerja sama seperti pada file originalnya sehingga pada saat dibuat, tidak dapat diketahui perbedaannya antara *hard link* yang baru dan *hard link* yang original dengan nama file tersebut.

Sangat memungkinkan untuk membuat banyak nama yang mengarah ke file yang sama. Ada dua cara untuk melakukan ini: Dapat membuat sebuah *hard link* kepada file, atau membuat sebuah *soft link* (biasa disebut sebagai sebuah *symbolic link*) terhadap masing-masing file. Kedua metode memiliki kelebihan dan kekurangannya masing-masing.

Kita dapat melihat jumlah *hard link count* pada tiap file seperti berikut:

```
[student@localhost Documents]$ ls -l
```

(Gambar 2. Command untuk melihat list directory content)

```
drwxrwxr-x. 2 student student 6 Feb 13 02:55 newfile.txt
```

(Gambar 3. *Link Count*)

Pada gambar 2 dan 3 terdapat sebuah command yang dapat dijalankan untuk mendapatkan hasil *link count* pada file "*newfile.txt*". Untuk pendalaman mengenai link itu sendiri silahkan melihat contoh pada materi praktikum.

## Managing File Systems

Dalam sistem operasi juga terdapat sistem permission yang mana dapat diatur langsung di dalam terminal yang tersedia. Perintah yang digunakan untuk mengubah izin dari baris perintah adalah `chmod` yang artinya "ubah mode" (izin juga disebut mode file). Perintah `chmod` mengambil instruksi izin diikuti dengan daftar file atau direktori untuk diubah. Izin instruksi dapat dikeluarkan baik secara simbolis (metode simbolik) atau secara numerik (numerik metode).

### 1. Mengubah Izin dengan Metode Simbolik

```
chmod WhoWhatWhich file | directory
```

Who = **u** (user), **g** (group), **o** (others), **a** (all)

What = **+** (add), **-** (remove), **=** (set exactly)

Which = **r** (read), **w** (write), **x** (execute)

Izin sendiri diwakili oleh satu huruf: **r** untuk membaca, **w** untuk menulis, dan **x** untuk menjalankan. Saat menggunakan `chmod` untuk mengubah izin dengan metode simbolik, gunakan huruf **kapital X** sebagai flag izin akan menambahkan izin eksekusi hanya jika file tersebut adalah direktori atau sudah dieksekusi ditetapkan untuk pengguna, grup, atau lainnya. Dapat juga ditambahkan **-R** sehingga penggantian seluruh direktori dapat dilakukan secara recursive

Contoh:

Menghapus **read** dan **write** permission dari group dan other pada file1:

```
[labit@host ~]$ chmod go-rw file1
```

Menambahkan **execute** permission kepada semua pada file2:

```
[labit@host ~]$ chmod a+x file2
```

### 2. Mengubah izin dengan metode numerik

```
chmod ### file | directory
```

4 = **r** (Read)

2 = **w** (Write)

1 = **x** (eXecute)

- Setiap digit mewakili izin untuk tingkat akses: **user**, **group**, **other**.
- Digit dihitung dengan menjumlahkan angka untuk setiap izin yang ingin Anda tambahkan, **4** untuk baca, **2** untuk tulis, dan **1** untuk eksekusi.

Menggunakan metode numerik, izin diwakili oleh 3 digit (atau 4 digit, saat pengaturan izin lanjutan) nomor oktal. Satu digit oktal dapat mewakili nilai tunggal apa pun dari 0-7.

Dalam representasi izin oktal (numerik) 3 digit, setiap digit mewakili satu tingkat akses, dari kiri ke kanan: pengguna, grup, dan lainnya. Untuk menentukan setiap digit:

1. Mulailah dengan 0.
2. Jika izin baca harus ada untuk tingkat akses ini, tambahkan 4.
3. Jika izin menulis harus ada, tambahkan 2.
4. Jika izin eksekusi harus ada, tambahkan 1.

Periksa izin `-rwxr-x---`. Untuk pengguna, `rwx` dihitung sebagai  $4+2+1=7$ . Untuk grup, `r-x` dihitung sebagai  $4+0+1=5$ , dan untuk pengguna lain, `---` direpresentasikan dengan 0. Menempatkan ketiganya bersama-sama, representasi numerik dari izin tersebut adalah 750. Perhitungan ini juga dapat dilakukan dengan arah yang berlawanan. Lihatlah izin 640. Untuk izin pengguna, 6 mewakili read (4) dan write (2), yang ditampilkan sebagai `rw-`. Untuk grup

Berikut adalah contoh tambahan:

Set **read** dan **write** permission untuk **user**, **read** permission untuk **group** dan **other** pada file "samplefile":

```
[labit@host ~]$ chmod 644 samplefile
```

Set **read**, **write**, and **execute** permissions untuk **user**, dan tidak ada permission untuk **other** pada "sampledir"

```
[labit@host ~]$ chmod 750 sampledir
```

## Changing File dan Directory User atau Group Ownership

Pada tiap direktori dan file terdapat sebuah kepemilikan akses dari file ataupun direktori tersebut. Dasarnya, sebuah file dapat diganti kepemilikannya secara keseluruhan oleh *root* dan selain dari *root* hanya dapat mengakses dan mengganti kepemilikan dari file yang mereka miliki sendiri. Kepemilikan file juga dapat diubah dengan perintah `chown` (ubah pemilik).

Berikut ini beberapa contoh mengganti kepemilikan dengan akses root:

```
[root@host ~]# chown student test_file
```

Ini akan mengganti kepemilikan file menjadi "student"

```
[root@host ~]# chown -R student test_dir
```

Ini akan menggantikan kepemilikan keseluruhan file menjadi "student"

```
[root@host ~]# chown :admins test_dir
```

Menggantikan hanya group dari "admins" dan tidak kepemilikan filenya

```
[root@host ~]# chown visitor:guests test_dir
```

Menggantikan kepemilikan (*ownership*) dan juga *group* dari file "test\_dir"

## Managing Default Permissions and File Access

### Special Permissions

Spesial permission merupakan jenis izin keempat selain pengguna dasar, grup, dan tipe yang lain. Sesuai namanya, izin ini memberikan fitur terkait akses tambahan melebihi apa yang diizinkan oleh jenis izin dasar. Bagian ini merinci dampak khusus izin, diringkas dalam tabel di bawah ini.

Spesial Permission	Efek pada file	Efek pada direktori
u+s (suid)	File dieksekusi sebagai pengguna yang memiliki file, bukan pengguna yang menjalankan file.	Tidak ada efek
g+s (sgid)	File dijalankan sebagai grup yang memiliki file.	File yang baru dibuat di direktori memiliki pemilik grup mereka diatur untuk mencocokkan pemilik grup direktori.
o+t (sticky)	Tidak ada efek	Pengguna dengan akses tulis ke direktori hanya dapat menghapus file yang mereka miliki; mereka tidak dapat menghapus atau paksa menyimpan ke file yang dimiliki oleh pengguna lain.

Izin setuid pada file yang dapat dieksekusi berarti bahwa perintah dijalankan sebagai pengguna yang memiliki file tersebut, bukan sebagai pengguna yang menjalankan perintah. Salah satu contohnya adalah perintah passwd:

```
[user@host ~]$ ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 35504 Jul 16 2010 /usr/bin/passwd
```

Dalam daftar panjang, Anda dapat mengidentifikasi izin setuid dengan huruf kecil s seperti biasanya mengharapkan x (pemilik mengeksekusi izin) menjadi. Jika pemilik tidak memiliki izin eksekusi, ini diganti dengan huruf besar S.

### Setting Special Permissions:

**Symbolically:** setuid = u+s; setgid = g+s; sticky = o+t

**Numerically** (fourth preceding digit): setuid = 4; setgid = 2; sticky = 1

Contoh:

**setgid pada direktori**

```
[user@host ~]# chmod g+s directory
```

Atur bit setgid dan tambahkan izin baca/tulis/eksekusi untuk pengguna dan grup, tanpa akses untuk yang lain, di direktori:

```
[user@host ~]# chmod 2770 directory
```

### Default File Permission

Ketika Anda membuat sebuah direktori atau file baru, maka akan diberikan izin awal. Izin awal ini dipengaruhi oleh dua faktor, yaitu jenis file yang dibuat (file biasa atau direktori), dan juga nilai umask saat itu.

Perintah umask tanpa argumen akan menampilkan nilai umask shell saat ini:

```
[user@host ~]$ umask
0002
```

Pemilik dan grup keduanya memiliki izin membaca, menulis, dan mengeksekusi pada direktori. Satu-satunya izin untuk yang lain adalah membaca.

```
[user@host ~]$ umask
0002
[user@host ~]$ touch default
[user@host ~]$ ls -l default.txt
-rw-rw-r--. 1 user user 0 May  9 01:54 default.txt
[user@host ~]$ mkdir default
[user@host ~]$ ls -ld default
drwxrwxr-x. 2 user user 0 May  9 01:54 default
```

Dengan menyetel nilai umask ke 0, izin file untuk perubahan lain dari baca menjadi baca dan tulis. Izin direktori untuk perubahan lain dari membaca dan mengeksekusi menjadi membaca, menulis, dan mengeksekusi.

Dan juga Sebagai root, Anda dapat mengubahnya dengan menambahkan skrip startup shell bernama /etc/profile.d/localumask.sh yang terlihat seperti keluaran dalam contoh ini:

```
[root@host ~]# cat /etc/profile.d/local-umask.sh
# Overrides default umask configuration
if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
    umask 007
else
    umask 022
fi
```



Contoh sebelumnya akan menyetel umask ke 007 untuk pengguna dengan UID lebih besar dari 199 dan dengan nama pengguna dan nama grup utama yang cocok, dan ke 022 untuk semua orang. Jika Anda hanya ingin atur umask untuk semua orang ke 022, Anda dapat membuat file itu hanya dengan konten berikut:

```
# Overrides default umask configuration
umask 022
```

Setelah mempelajari mengenai dasar-dasar dari materi pokok, selanjutnya akan masuk pada materi praktikum yang akan mengenali lebih mendalam mengenai berbagai macam kepemilikan file dan cara mengantinya.

## MATERI PRAKTIKUM

### Direktori

Kita dapat melihat seluruh file utama pada “~” yang merupakan halaman utama pada saat membuka terminal. Berikut ini adalah berbagai list yang merupakan aplikasi dasar pada saat awal instalasi:

```
[student@localhost ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[student@localhost ~]$
```

(Gambar 4. List direktori pada saat baru buka terminal)

#### 1. Membuat Direktori

Berikut ini adalah cara **salah** dalam membuat sebuah direktori:

```
[student@localhost ~]$ mkdir downloads/modul2
mkdir: cannot create directory 'downloads/modul2': No such file or directory
```

Kenapa bisa salah? karena dalam membuat sebuah direktori dari asalnya kita harus memperhatikan setiap huruf secara detail yang mana **downloads** harusnya diawali dengan huruf D besar sehingga menjadi **Downloads**

Berikut ini adalah hasil pembuatan direktori yang benar:

```
[student@localhost ~]$ mkdir Downloads/modul2
[student@localhost ~]$
```

(Gambar 5. Pembuatan direktori modul2)

Sekarang direktori telah berhasil di buat, kita juga dapat membuat direktori baru sebagai parent dengan menggunakan -p setelah mkdir.

```
[student@localhost ~]$ cd Downloads
[student@localhost Downloads]$ ls -la
total 4
drwxr-xr-x. 3 student student 20 Feb 15 23:51 .
drwx----- 16 student student 4096 Feb 15 23:50 ..
drwxrwxr-x. 2 student student 6 Feb 15 23:51 modul2
```

(Gambar 6. Melihat keseluruhan isi modul 2 dengan -la)

Terlihat sekarang bahwa sudah ada direktori bernama “modul2” dan telah berhasil membuat direktori baru!

## 2. Menghapus direktori

Kita dapat menghapus direktori dengan cara rm dan menambahkan -R (recursively) agar seluruh isi direktori juga ikut terhapus

```
[student@localhost Downloads]$ rm -R modul2
[student@localhost Downloads]$ ls
[student@localhost Downloads]$ l
```

(Gambar 7. Menghapus direktori)

Pada gambar 7 terlihat bahwa setelah di remove tidak terdapat lagi file sama sekali pada **Downloads/** sehingga telah berhasil kita hapus.

## 3. Membuat file baru

Sekarang mencoba membuat sebuah file baru, harus di ingat ya bahwa file dan direktori itu berbeda, jangan sampai kebalik!

```
[student@localhost Downloads]$ touch modul2
[student@localhost Downloads]$ ls
modul2
```

(Gambar 8. Pembuatan file baru)

Gambar 8 memperlihatkan pembuatan file baru dan dapat langsung terlihat secara jelas.

## 4. Menghapus file

Menghapus file bisa dilakukan dengan cara rm secara langsung

## Membuat Hard Links & Soft Links

### Hard Links

Kita dapat melihat jumlah link dengan mengetik “ls -l” yang kemudian akan terlihat seperti berikut:

```
[student@localhost Downloads]$ ls -l
total 0
-rw-rw-r-- 1 student student 0 Feb 16 00:02 modul2
```

(Gambar 9. Jumlah Links)

Setelah itu kita akan membuat sebuah hard link baru dengan cara `ln {nama text} {letak}` dan akan membuat sebuah hard link baru seperti contoh dibawah ini:

```
[student@localhost Downloads]$ ln test1.txt /tmp/test1.txt
[student@localhost Downloads]$ ls -l
total 0
-rw-rw-r--. 1 student student 0 Feb 16 00:02 modul2
-rw-rw-r--. 2 student student 0 Mar 13 01:36 test1.txt
[student@localhost Downloads]$ ls -l /tmp/test1.txt
-rw-rw-r--. 2 student student 0 Mar 13 01:36 /tmp/test1.txt
[student@localhost Downloads]$ -
```

(Gambar 10. Hard links)

Gambar 10 menunjukkan proses pembuatan hard link. Hard link sendiri memiliki beberapa batasan. Pertama, Hard link hanya dapat digunakan dengan file biasa. Dan tidak bisa gunakan ln untuk membuat Hard link pada direktori atau file khusus. Kedua, Hard link hanya dapat digunakan jika kedua file berada di sistem file yang sama. Sistem file hirarki dapat terdiri dari beberapa perangkat penyimpanan. Tergantung pada konfigurasi Anda sistem, ketika Anda mengubah ke direktori baru, direktori itu dan isinya dapat disimpan di sistem file yang berbeda.

### Soft Links

Kemudian kita dapat membuat sebuah softlink seperti cara berikut:

```
[user@host ~]$ ln -s /home/user/newfile-link2.txt /tmp/newfile-symlink.txt
[user@host ~]$ ls -l newfile-link2.txt /tmp/newfile-symlink.txt
-rw-rw-r--. 1 user user 12 Mar 11 19:19 newfile-link2.txt
lrwxrwxrwx. 1 user user 11 Mar 11 20:59 /tmp/newfile-symlink.txt -> /home/user/
newfile-link2.txt
[user@host ~]$ cat /tmp/newfile-symlink.txt
Soft Hello World
```

(Gambar 11. Pembuatan Soft link)

Pada saat ln -s (-s disini adalah soft link) yang mana pada saat kita ls -l akan terlihat tambaha "l" pada soft link yang ada.

Akan tetapi, pada saat kita menghapus file masternya, maka keseluruhan soft linknya juga akan hilang.

```
[user@host ~]$ rm -f newfile-link2.txt
[user@host ~]$ ls -l /tmp/newfile-symlink.txt
lrwxrwxrwx. 1 user user 11 Mar 11 20:59 /tmp/newfile-symlink.txt -> /home/user/
newfile-link2.txt
[user@host ~]$ cat /tmp/newfile-symlink.txt
cat: /tmp/newfile-symlink.txt: No such file or directory
```

(Gambar 12. Saat soft link dihapus)

## LEMBAR KERJA

### KEGIATAN 1

#### Mengelola files

1. Di direktori home, gunakan perintah untuk membuat tiga subdirektori: Musik, Gambar, dan Video.  
Subdirektori **musik**  
Subdirektori **Foto**  
Subdirektori **Videos**
2. Selanjutnya pada direktori home buatlah 5 file kosong dengan penamaan seperti berikut:  
**lagu1.mp3 ... lagu5.mp3**  
**gambar1.jpg ... gambar5.jpg**  
**rekamanan1.avi ... rekaman5.avi**
3. Setelah membuat total 15 file kosong, kemudian dimasukkan kedalam setiap subdirektori sesuai dengan penamaan yang ada. (\*sesuaikan yaa)
4. Kemudian pada direktori home student (direktori yang ~) buatlah tiga Subdirektori dengan nama:  
Subdirektori **keluarga**  
Subdirektori **teman**  
Subdirektori **kerja**
5. Kemudian pindahkan file yang telah dibuat pada bagian 3 untuk dimasukkan pada subdirektori yang telah dibuat pada bagian 5
6. Perlihatkan hasil yang sudah dibuat dan dijelaskan
7. Exit

### KEGIATAN 2

1. Buat sebuah hardlink pada /home/student/backups/source.backup untuk file bernama "**source.file**" (hint: letak source file ada di /home/student/files/source.file)
2. Buat sebuah soft link pada /home/student/ dengan nama "tempdir" dan link pada file /tmp.
3. Perlihatkan hasil soft link pada /home/student/tempdir dan dijelaskan
4. Exit

### KEGIATAN 3

1. Dari user student pindah menjadi **root user**
2. Buat sebuah direktori bernama modul2 di /home/
3. Ganti kepemilikan file group menjadi **student**
4. Gunakan perintah ls untuk mengkonfirmasi izin dari **student** grup mengizinkan anggota grupnya untuk membuat file, dan menghapus file dari /home/modul2
5. Tambahkan *permission* **write** pada direktori modul2
6. Gunakan perintah chmod untuk melarang orang lain mengakses file di /home/modul2 dengan menggunakan **angka** (bukan huruf rwx).

7. Sekarang buat sebuah file bernama **modul2.txt**
8. Perlihatkan permission dari file yang baru saja dibuat
9. Ganti kepemilikan file menjadi **student**

#### KEGIATAN 4

1. switch user ke operator1
2. pakai command untuk lihat default umask pada user operator1
3. Buat direktori /tmp/shared. Gunakan ls -ld perintah untuk melihat daftar izin dari direktori baru tersebut
4. Buat sebuah file bernama default di /tmp/shared direktori.
5. Perlihatkan hasil pembuatan file
6. Gunakan perintah chmod untuk mengatur ID grup ke grup operator untuk /tmp/ yang bersifat **shared** direktori
7. Gunakan perintah echo untuk mengubah umask default untuk pengguna **operator1** dengan umask default **007**.
8. exit dari **operator1**
9. switchuser lagi dan perlihatkan umask default baru dari **operator1**

#### RUBRIK PENILAIAN

Aspek Penilaian	Bobot Penilaian
Ketepatan menjawab semua kegiatan	65%
Pemahaman setiap aspek materi yang dibahas	20%
Quiz pertemuan materi minggu pertama	15%
<b>Total</b>	<b>100%</b>