

# Praktikum 4

---

## Standar Input Output dan Pipa

---

### POKOK BAHASAN:

- ✓ Proses Input Output
- ✓ File Descriptor
- ✓ Pembelokan (*Redirection*)
- ✓ Pipa (*Pipeline*)

### TUJUAN BELAJAR:

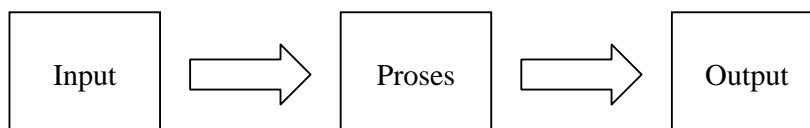
Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- ✓ Memahami proses *redirection* proses I/O ke file
- ✓ Memahami bagaimana menghubungkan proses dengan pipa (*pipe*).

### DASAR TEORI:

#### 1 PROSES INPUT OUTPUT

Sebuah proses memerlukan Input dan Output.



Instruksi (*command*) yang diberikan pada Linux melalui Shell disebut sebagai *eksekusi program* yang selanjutnya disebut *proses*.

Setiap kali instruksi diberikan, maka Linux kernel akan menciptakan sebuah proses dengan memberikan nomor PID (*Process Identity*).

Proses dalam Linux selalu membutuhkan Input dan menghasilkan suatu Output.

Dalam konteks Linux input/output adalah :

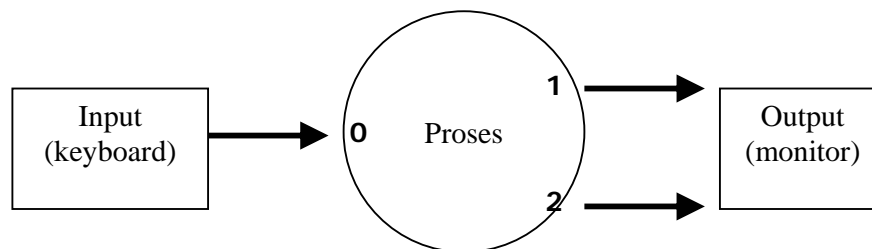
- Keyboard (input)
- Layar (output)
- Files
- Struktur data kernel
- Peralatan I/O lainnya (misalnya Network)

## 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 lalu diciptakan oleh proses adalah :

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



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 < pengganti standar input
- 1> atau > pengganti standar output
- 2> pengganti standar error
- >> menambah file hasil redirection standar output
- 2>> menambah file hasil redirection standar error

#### 4 PIPA (PIPELINE)

Mekanisme pipa digunakan sebagai alat komunikasi antar proses.

Input  $\Rightarrow$  Proses1  $\Rightarrow$  Output = Input  $\Rightarrow$  Proses2  $\Rightarrow$  Output

Proses 1 menghasilkan output yang selanjutnya digunakan sebagai input oleh Proses 2. Hubungan output input ini dinamakan pipa, yang menghubungkan Proses 1 dengan Proses2 dan dinyatakan dengan symbol “[|]”.

Proses1        |        Proses2

#### TUGAS PENDAHULUAN:

Jawablah pertanyaan-pertanyaan di bawah ini :

1. Apa yang dimaksud standar input, output dan error ?
2. Apa yang dimaksud pembelokan atau *redirection* ?
3. Apa yang dimaksud pipa atau *pipeline* ?
4. Apa yang dimaksud perintah tee ?

#### PERCOBAAN:

1. Login sebagai user.
2. Masuk ke TTY1 dengan menekan *Ctrl-Alt-F1* dan lakukan percobaan-percobaan di bawah ini. Perhatikan hasilnya.
3. Selesaikan soal-soal latihan.

### Percobaan 1 : File descriptor

1. Output ke layar (standar output), input dari system (kernel)

```
$ ps
```

2. Perintah cat tanpa argumen akan menerima input dari keyboard (standar input) dan menampilkan output ke layar (standar output)

```
$ cat
hallo, apa khabar
hallo, apa khabar
exit dengan ^d
exit dengan ^d
[Ctrl-d]
```

3. Perintah tr merupakan filter mempunyai 2 argumen, membaca standar input, menterjemahkan kata pada string pertama ke string kedua dan menampilkan ke standar output. Perhatikan outputnya

```
$ tr 'aeiou' 'AEIOU'
hallo, apa khabar
[Ctrl-d]
```

4. Input dari keyboard dan output ke alamat internet

```
$ mail arna@eepis-its.edu
contoh surat yang langsung
dibuat pada standard input (keyboard)
[Ctrl-d]
```

5. Input nama direktori, output tidak ada (membuat direktori baru), bila terjadi error maka tampilan error pada layar (standard error)

```
$ mkdir mydir
$ mkdir mydir      (Terdapat pesan error)
```

### Percobaan 2 : Pembelokan (*redirection*)

1. Pembelokan standar output

```
$ cat 1> myfile.txt
Ini adalah teks yang saya simpan
Ke file myfile.txt
```

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

```
$ cat 0< myfile.txt
$ cat myfile.txt
```

**3. Membelokkan file pada perintah `find`**

```
$ find /etc/\ -name passwd
$ find /etc/\ -name passwd > findresult
$ cat findresult
```

**4. Menindih dan menambah file**

```
$ find /etc -name passwd > output
$ cat output
$ find /etc -name passwd >> output
$ cat output
$ find /etc -name passwd > output
$ cat output
```

**5. Pembelokan standar**

```
$ find /etc -name passwd 2> finderrors
$ cat finderrors
$ find /etc -name passwd 2>> finderrors
$ cat finderrors
```

**6. Pembelokan standar output dan error**

```
$ find /etc -name passwd 2> errs > results
$ cat errs
$ cat result
```

**7. Pembelokan standar output dan error ke file yang sama**

```
$ find /etc -name passwd > alloutput 2>&1
$ cat alloutput
```

**Percobaan 3 : Pipa (*pipeline*)**

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

```
$ who
$ who | sort
$ ls -l /etc | more
$ ls -l /etc | sort | more
$ ls -l /usr/lib | less
$ cut -f1 -d: /etc/passwd | sort -r | less
$ ls -C | tr 'a-z' 'A-Z'
```

2. Perintah tee untuk membelokkan output ke file sementara proses pipa ke program lain masih berjalan.

```
$ set | tee set.out | less
$ ls -lR /etc | tee stage1.out | sort | tee stage2.out \
| uniq -c | tee stage3.out | sort -r | tee stage4.out |\
less
```

### LATIHAN:

1. Dengan perintah cat buatlah file package1.txt yang berisi 8 baris seperti di bawah ini

```
$ cat > package1.txt
> amanda
> galeon
> metacity
> firefox
> postgresql
> procinfo
> rpmfind
> squid
[Ctrl-D]
```

2. Lakukan hal yang sama dengan no 1 untuk file package2.txt yang berisi 6 baris yaitu :

```
anaconda
openssh
gnome-core
samba
sendmail
xscreensaver
```

3. Belokkan file input packages1.txt ke file output packages1.catfile kemudian bandingkan kedua file tersebut dengan perintah diff file1 file2. Tuliskan perintahnya dan perhatikan outputnya.
4. Tambahkan file packages2.txt ke file packages1.catfile dengan perintah redirection.
5. Buatlah sebuah file typedin.txt dengan perintah cat yang berisi kalimat :  
Mencoba menuliskan teks pada keyboard  
dan membelokkan output ke file typedin.txt

6. Buatlah file `trfile.txt` dengan perintah `tr` untuk mengubah huruf vokal menjadi huruf kapital pada kalimat di latihan 5.

7. Perhatikan output untuk perintah `set -o` seperti di bawah ini:

```
$ set -o
$ ls -l /etc/passwd > trfile.txt
$ cat trfile.txt
```

8. Modifikasi latihan 7 menggunakan pilihan `noclobber` seperti di bawah. Bagaimana output yang dihasilkan dan apa perbedaannya dengan latihan 7?

```
$ set -o noclobber
$ echo "new contents" > trfile.txt
```

9. Ubahlah huruf vokal menjadi kapital dari file `packages1.txt` dan belokkan output ke `packages1.trfile` dengan perintah `tr`.

7. Gunakan perintah di bawah ini dan perhatikan hasilnya.

```
$ cat > hello.txt
dog cat
cat duck
dog chicken
chicken duck
chicken cat
dog duck
[Ctrl-d]
$ cat hello.txt | sort | uniq
$ cat hello.txt | grep "dog" | grep -v "cat"
```

### LAPORAN RESMI:

1. Analisa hasil percobaan 1 sampai dengan 3, untuk setiap perintah jelaskan tampilannya.
2. Kerjakan latihan diatas dan analisa hasilnya
3. Berikan kesimpulan dari praktikum ini.