

Modul Praktikum Sistem Operasi



Jurusan Teknik Informatika

Rudhi Wahyudi Febrianto, S.Kom., M.Kom
Sekolah Tinggi Teknologi Bandung

Modul V

Proses Input/Output (I/O)

SASARAN

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

1. Memahami konsep proses I/O dan redirection
2. Memahami standar input, output dan error
3. Menggunakan notasi output, append dan here document
4. Memahami konsep PIPE dan filter

Pipeline adalah teknik menjadikan output sebuah proses menjadi input proses lainnya. Sedangkan redirection adalah teknik mengalihkan input atau output dari standarnya.

Redirection ada dua jenis yaitu output redirection dan input redirection. Selain itu ada beberapa standar input output di konsol linux yang perlu kita tahu. Beberapa standar itu adalah sbb:

Standar input adalah keyboard standar output adalah layar monitor, Standar error adalah layar monitor Untuk standar error tidak akan kita bahas, mengingat pembahasan tentang redirection untuk standar error lebih kearah programing atau scripting.

Link pembahasan lain :

<https://www.candra.web.id/pipeline-dan-redirection-di-linux/>

<https://repository.unikom.ac.id/32576/1/Pertemuan%203.pdf>

<https://www.awonapa.com/2020/04/belajar-linux-operasi-input-output-pada.html>

PRAKTIKUM

1. Login sebagai user.
-

2. Bukalah Console Terminal dan lakukan percobaan-percobaan di bawah ini. Perhatikan hasil setiap percobaan.
3. Selesaikan soal-soal latihan.

Percobaan 1 : File Descriptor

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

```
$ ps
```

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

```
$ cat
```

```
hallo, apa khabar
```

```
hallo, apa khabar
```

```
exit dengan ^d
```

```
exit dengan ^d
```

```
[Ctrl-d]
```

3. Input dari keyboard dan output ke alamat internet

```
mail nama@yahoo.com
```

```
contoh surat yang langsung
```

```
dibuat pada standard input (keyboard)
```

```
[Ctrl-d]
```

4. 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. Pembelokan standar error untuk disimpan di file

```
$ mkdir mydir (Terdapat pesan error)
$ mkdir mydir 2> myerror.txt
$ cat myerror.txt
```

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

```
$ ls filebaru (Terdapat pesan error)
$ ls filebaru 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 error

```
$ echo "mencoba menulis file" 1> baru
$ cat filebaru 2> baru 1>&2
$ cat baru
```

6. Notasi >> (append)

```
$ echo "kata pertama" > surat
$ echo "kata kedua" >> surat
$ echo "kata ketiga" >> surat
$ cat surat
$ echo "kata keempat" > surat
$ cat surat
```

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 <<++
Hallo, apa kabar ?
Baik-baik saja ?
Ok!
++
$ cat <<%%%
```

Hallo, apa kabar ?

Baik-baik saja ?

Ok!

%%%

8. Notasi `-` (input keyboard) adalah representan 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 myfile.txt - surat
```

9. Untuk membelokkan standart output ke file, digunakan operator `>`

```
$ echo hello
```

```
$ echo hello > output
```

```
$ cat output
```

10. Untuk menambahkan output ke file digunakan operator `>>`

```
$ echo bye >> output
```

```
$ cat output
```

11. Untuk membelokkan standart input digunakan operator <

```
$ cat < output
```

12. Pembelokan standart input dan standart output dapat dikombinasikan tetapi tidak boleh menggunakan nama file yang sama sebagai standart 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
```

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
```

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
Badu
Zulkifli
Yulizir
Yudi
Ade
[Ctrl-d]
$ cat > kelas2.txt
Budi
Gama
Asep
Muchlis
[Ctrl-d]
$ cat kelas1.txt kelas2.txt | sort
$ cat kelas1.txt kelas2.txt > kelas.txt
$ cat kelas.txt | sort | uniq
```

LATIHAN

1. Lihat daftar secara lengkap pada direktori aktif, belokkan tampilan standard output ke file baru.
 2. Lihat daftar secara lengkap pada direktori /etc/passwd, belokkan tampilan standard output ke file baru tanpa menghapus file baru sebelumnya.
 3. Urutkan file baru dengan cara membelokkan standard input.
 4. Urutkan file baru dengan cara membelokkan standard input dan standard output ke file baru.urut.
 5. Buatlah direktori latihan2 sebanyak 2 kali dan belokkan standard error ke file rmdirerror.txt.
 6. Urutkan kalimat berikut :
Jakarta
-

Bandung

Surabaya

Padang

Palembang

Lampung

Dengan menggunakan notasi here document (<@@@ ...@@@)

7. Hitung jumlah baris, kata dan karakter dari file `baru.urut` dengan menggunakan filter dan tambahkan data tersebut ke file `baru`.
8. 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 4, untuk setiap perintah jelaskan tampilannya.
 2. Kerjakan latihan diatas dan analisa hasilnya
 3. Berikan kesimpulan dari praktikum ini.
-