

JOBSHEET 3

REDIRECTION DAN PIPELINE

Nama: Rizqi Bagus Andrean

Kelas: TI-1D

Absen: 25

Redirection adalah pembelokan dilakukan untuk standard input, output dan error, yaitu untuk mengalihkan file descriptor dari 0, 1 dan 2. Redirection dalam sistem Linux merupakan salah satu cara untuk melewati output dari suatu program ke program lain, dengan redirection kita bisa menyimpan output suatu file baik kedalam sebuah file atau diproses dengan program lain.

- Simbol untuk pembelokan adalah :
 - 0< atau < pengganti standard input
 - 1> atau > pengganti standart output
 - 2> pengganti standart error

Pipeline adalah suatu mekanisme pipa digunakan sebagai alat komunikasi antar proses dan juga fasilitas yang di shell Linux yang berfungsi memberikan input dari suatu proses dari output proses yang lain.

Input → Proses1 → Output = Input → Proses2 → 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

Apa yang dimaksud perintah di bawah ini :

echo, cat, more, sort, grep, wc, cut, uniq

- echo digunakan untuk menampilkan output kelayar.
- cat digunakan untuk menghasilkan output kelayar dan merupakan berasal dari input sebuah keyboard.
- more merupakan perintah untuk mempaging halaman.
- sort digunakan untuk mengurutkan masukannya berdasarkan urutan nomor ASCII dari karakter.
- grep digunakan untuk menyaring masukannya dan menampilkan baris-baris yang hanya mengandung pola yang ditentukan. Pola ini disebut regular expression.
- wc digunakan untuk menghitung jumlah baris, kata dan karakter dari baris-baris masukan yang diberikan kepadanya. Untuk mengetahui berapa baris gunakan option -l, untuk mengetahui

berapa kata, gunakan option `-w` dan untuk mengetahui berapa karakter, gunakan option `-c`. Jika salah satu option tidak digunakan, maka tampilannya adalah jumlah baris, jumlah kata dan jumlah karakter.

- `cut` digunakan untuk mengambil kolom tertentu dari baris-baris masukannya, yang ditentukan pada option `-c`.
- `uniq` digunakan untuk menghilangkan baris-baris berurutan yang mengalami duplikasi, biasanya digabungkan dalam pipeline dengan `sort`.

Praktikum:

1. Login sebagai user

```
ubuntuserver login: kadek
Password:
```

2. Bukalah Console Terminal dan lakukan percobaan-percobaan di bawah ini. Perhatikan hasil setiap percobaan

Percobaan 1 : File descriptor

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

```
kadek@ubuntuserver:~$ ps
  PID TTY          TIME CMD
 1213 tty1        00:00:00 bash
 1233 tty1        00:00:00 ps
kadek@ubuntuserver:~$
```

Analisa =

- Perintah `ps` digunakan untuk memperlihatkan proses yang sedang berjalan pada system (kernel) diperlihatkan pada layar (standart output).
- PID (Process Identifier) memperlihatkan nomor setiap proses yang diberikan berurutan oleh sistem linux.

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

```
kadek@ubuntuserver:~$ cat

hello kadek suarjuna
hello kadek suarjuna
who are you today?_
```

Analisa = Perintah `cat` digunakan untuk menghasilkan output kelayar dan merupakan berasal dari input sebuah keyboard.

4. Input dari keyboard dan output ke alamat internet.

```
kadek@ubuntuserver:~$ mail kadeksuarjuna87@gmail.com
Cc: kadeksuarjuna87@polinema.ac.id
Subject: Jobsheet 3
Hallo Kadek Suarjuna
Saya Kadek Suarjuna sedang mengirim pesan kepada kadek
oke
```

Analisa = Perintah mail digunakan untuk menulis email dari keyboard yang ditampilkan ke terminal dan mengirimnya ke alamat yang dituju.

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

```
kadek@ubuntuserver:~$ mkdir mydir
kadek@ubuntuserver:~$ mkdir mydir
mkdir: cannot create directory 'mydir': File exists
kadek@ubuntuserver:~$ _
```

Analisa =

- Perintah mkdir digunakan untuk membuat direktori. Pada percobaan tersebut dibuat direktori Bernama "mydir", setelah itu dibuat lagi direktori Bernama "mydir".
- Jika ada nama direktori yang sama atau yang telah ada, maka muncul pesan error yang menandakan ketersediaan direktori tersebut.

6. Pembelokan standar output.

```
kadek@ubuntuserver:~$ mkdir mydir
kadek@ubuntuserver:~$ mkdir mydir
mkdir: cannot create directory 'mydir': File exists
kadek@ubuntuserver:~$ cat 1> myfile.txt
Halo, berikut ini adalah isi dari myfile.txt_
```

Analisa =

- Perintah cat digunakan untuk menghasilkan output kelayar dan merupakan berasal dari input sebuah keyboard.
- Input dari keyboard akan dibelokkan atau dimasukkan kedalam file bernama "myfile.txt", kemudian output akan ditampilkan ke layar.

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

```
kadek@ubuntuserver:~$ cat 0< myfile.txt
Halo, berikut ini adalah isi dari myfile.txt
kadek@ubuntuserver:~$ cat myfile.txt
Halo, berikut ini adalah isi dari myfile.txt
kadek@ubuntuserver:~$ _
```

Analisa =

- Perintah cat digunakan untuk menghasilkan output kelayar dan merupakan berasal dari input sebuah keyboard.
- Pada percobaan diatas terdapat "0<" yang berarti standar input, artinya isi file "myfile.txt" akan menjadi input, kemudian output di tampilkan ke layar.

8. Pembelokan standar error untuk disimpan di file.

```
kadek@ubuntuserver:~$ mkdir mydir
mkdir: cannot create directory 'mydir': File exists
kadek@ubuntuserver:~$ mkdir mydir 2> myerror.txt
kadek@ubuntuserver:~$ cat myerror.txt
mkdir: cannot create directory 'mydir': File exists
kadek@ubuntuserver:~$ _
```

Analisa =

- Perintah mkdir digunakan untuk membuat direktori. Direktori “mydir” sudah ada sebelumnya, jadi muncul error artinya tidak bisa membuat direktori dengan nama yang sama.
- Pada percobaan diatas terdapat “2>” yang berarti standar error, jadi peringatan error dari “mkdir mydir” akan digunakan sebagai input yang dimasukan ke file bernama “myerror.txt”.
- Ketika menulis perintah “cat myerror.txt” akan menghasilkan output berupa hasil error dari “mkdir mydir”.

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

```
kadek@ubuntuserver:~$ ls filebaru
ls: cannot access 'filebaru': No such file or directory
kadek@ubuntuserver:~$ ls filebaru 2> out.txt
kadek@ubuntuserver:~$ cat out.txt
ls: cannot access 'filebaru': No such file or directory
kadek@ubuntuserver:~$ ls filebaru 2> out.txt 2>&1
ls: cannot access 'filebaru': No such file or directory
kadek@ubuntuserver:~$ cat out.txt
```

Analisa =

- Perintah ls digunakan untuk melihat isi dari direktori. Ketika perintah “ls filebaru” di masukan akan muncul error karena tidak terdapat direktori bernama “filebaru” dalam komputer.
- Kemudian dibuat lagi direktori “filebaru” yang hasil error outputnya akan dimasukan ke dalam file Bernama “out.text”.
- Perintah “cat out.text” akan menampilkan output error yang berasal dari “filebaru” tadi.
- Dan terakhir perintah “2>&1” artinya isi error file “out.text” akan menjadi output ke layar, tetapi tidak disimpan. Maka dari itu perintah “cat out.text” tidak menampilkan apa-apa.

10. Notasi 1>&2 (atau >&2) : pembelokan standar output adalah sama dengan file descriptor 2 yaitu standar error.

```
kadek@ubuntuserver:~$ echo "mencoba menulis file" 1> baru
kadek@ubuntuserver:~$ cat filebaru 2> baru 1>&2
kadek@ubuntuserver:~$ cat baru
```

Analisa =

- Perintah echo digunakan untuk menampilkan output kelayar. Kalimat “mencoba menulis file” akan menjadi output file bernama “baru”.
- Lalu, ditampilkan file “filebaru” yang tentu saja akan menghasilkan error karena tidak ada file Bernama “filebaru” di dalam komputer, dan hasil error akan menjadi input file bernama “baru”.

- Ketika perintah “cat baru” dijalankan maka muncullah isi dari file tersebut berupa error dari file “filebaru” tadi.

11. Notasi >> (append)

```
cat: filebaru: No such file or directory
kadek@ubuntuserver:~$ echo "kata pertama"> surat
kadek@ubuntuserver:~$ echo "kata kedua">> surat
kadek@ubuntuserver:~$ echo "kata ketiga">> surat
kadek@ubuntuserver:~$ cat surat
kata pertama
kata kedua
kata ketiga
kadek@ubuntuserver:~$ echo "kata keempat" > surat
kadek@ubuntuserver:~$ cat surat
kata keempat
kadek@ubuntuserver:~$
```

Analisa =

- Penggunaan karakter > digunakan untuk membelokkan output dari echo menjadi sebuah filebaru Bernama surat, maka “kata pertama” akan menjadi input dari surat.
- Penggunaan karakter >> digunakan untuk menyisipkan output dari echo di kelanjutan isi dari file tujuan.
- Kemudian “kata kedua” dan “kata ketiga” juga menjadi input dalam surat.
- Perintah “cat surat” dijalankan akan menghasilkan ketiga kata yang telah diinputkan.
- Lalu, “kata keempat” akan mereplace isi dari surat karena karakter >.

12. Notasi here document (<<++ ++) dan (<<%%% %%%) 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.

```
kadek@ubuntuserver:~$ cat <<++
> Hallo kadek, Apa Kabar nya?
> baik-baik ya?
> Siap
> ++
Hallo kadek, Apa Kabar nya?
baik-baik ya?
Siap
kadek@ubuntuserver:~$

kadek@ubuntuserver:~$ cat <<%%%
> Hallo, apa kabarnya?
> Luar biasa kan?
> Ok
> %%%
Hallo, apa kabarnya?
Luar biasa kan?
Ok
kadek@ubuntuserver:~$ _
```

Analisa = Dengan menggunakan notasi here document, kita tidak perlu lagi menekan ctrl+d untuk keluar dari editor untuk menyimpan teks.

13. 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.

```

kadek@ubuntuserver:~$ cat myfile.txt - surat
Halo, berikut ini adalah isi dari myfile.txt

Ini input dari keyboard setelah menampilkan file 1
Ini input dari keyboard setelah menampilkan file 1
kata keempat

```

Analisa =

- Perintah cat digunakan untuk menghasilkan output kelayar dan merupakan berasal dari input sebuah keyboard.
- Pertama isi dari file "myfile.txt" yang ditulis sebelumnya pada percobaan ke (2) akan ditampilkan.
- Kemudian kita bisa menginputkan apa saja kedalam layer dan hasil input tadi akan ditampilkan di layer, terakhir isi dari file "surat" yang ditulis sebelumnya pada percobaan ke (6) akan ditampilkan.

14. Untuk membelokkan standart output ke file, digunakan operator >

```

kadek@ubuntuserver:~$ echo hai guys
hai guys
kadek@ubuntuserver:~$ echo hai guys > output
kadek@ubuntuserver:~$ cat output
hai guys
kadek@ubuntuserver:~$

```

Analisa =

- Perintah echo digunakan untuk menampilkan output kelayar. Perintah echo akan menampilkan kalimat "hai guys" ke terminal.
- Selanjutnya kalimat "hai guys" dibelokkan ke output dan output dibaca muncul kalimat "hai guys".

15. Untuk menambahkan output ke file digunakan operator >>

```

kadek@ubuntuserver:~$ echo bye >> output
kadek@ubuntuserver:~$ cat output
hai guys
bye
kadek@ubuntuserver:~$

```

Analisa =

- Perintah echo digunakan untuk menampilkan output kelayar. Perintah echo akan menampilkan kalimat "hai guys" ke terminal.
- Selanjutnya kalimat "bye" akan ditambahkan ke output dan output dibaca muncul kalimat "hai guys" dan "bye".

16. Untuk membelokkan standart input digunakan operator <

```

kadek@ubuntuserver:~$ cat < output
hai guys
bye
kadek@ubuntuserver:~$

```

Analisa = Perintah "cat < output" akan menampilkan output pada terminal.

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

```

kadek@ubuntuserver:~$ cat < output > out
kadek@ubuntuserver:~$ cat out
hai guys
bye
kadek@ubuntuserver:~$ cat < output >> out
kadek@ubuntuserver:~$ cat out
hai guys
bye
hai guys
bye
kadek@ubuntuserver:~$ cat < output > output
kadek@ubuntuserver:~$ cat output
kadek@ubuntuserver:~$ cat < out >> out
cat: -: input file is output file
kadek@ubuntuserver:~$ cat out
hai guys
bye
hai guys
bye
kadek@ubuntuserver:~$ _

```

Analisa =

- Perintah “cat < output > output” digunakan untuk menyimpan hasil dari file “output” ke “output” itu sendiri dan isi dari “output” akan di menjadi output dari file “out”.
- Perintah “cat < output >> output” digunakan untuk menyisipkan isi file out kedalam baris selanjutnya dari file out itu sendiri.
- Proses ini akan terus menerus menambah baris teks karena isi file out akan terus diperbaharui tanpa henti. Jika diberikan perintah cat out maka baris teks isi file out tidak akan berhenti.

18. Pipeline

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

```

kadek@ubuntuserver:~$ who
kadek    tty1      2024-02-25 01:09
kadek@ubuntuserver:~$ who | sort
kadek    tty1      2024-02-25 01:09
kadek@ubuntuserver:~$ who | sort -r
kadek    tty1      2024-02-25 01:09
kadek@ubuntuserver:~$ who > tmp
kadek@ubuntuserver:~$ sort tmp
kadek    tty1      2024-02-25 01:09
kadek@ubuntuserver:~$ rm tmp
kadek@ubuntuserver:~$ ls -l /etc | more_

```

Analisa =

- Perintah “who” menampilkan siapa saja yang sedang login. Who -> akan menampilkan informasi mengenai login name, jenis terminal, waktu login dan remote hostname untuk setiap user yang saat itu sedang login.
- Perintah “who” yang diberi pipe “|” sort akan membuat data yang ditampilkan secara urut. Perintah pipe “|” sort dan more “-r” digunakan untuk menampilkan file dan atribut secara lengkap dan ditampilkan secara urut.

- hasil perintah “who” dimasukan kedalam file “temp”, lalu di tampilkan secara urut menggunakan “sort”, kemudian file “temp” di hapus menggunakan perintah “rm”.
- Perintah “ls -l /etc | more” artinya menampilkan list direktori etc secara lengkap beserta atributnya.
- Perintah “ls -l /etc | sort | more” artinya menampilkan list direktori etc secara lengkap beserta atributnya dan juga berurutan.

19. Filter

Pipa juga digunakan untuk mengkombinasikan utilitas sistem untuk membentuk fungsi yang lebih kompleks

```
Ketikkan:
w -h | grep kadek
grep kadek /etc/passwd
ls /etc | wc
ls /etc | wc -l
cat > kelas1.txt
Nama kelas 1
cat > kelas2.txt
Nama kelas 2
Cat kelas1.txt kelas2.txt | sort
Cat kelas1.txt kelas2.txt > kelas.txt
Cat kelas.txt | sort | uniq
```

Analisa =

- Penggunaan “w -h | grep Damelia” digunakan untuk menyaring informasi atau mencari kata kunci user yang memiliki id kadek.
- Perintah “grep Damelia /etc/passwd” digunakan untuk mencari informasi user dengan kata kunci Damelia pada direktori “/etc/passwd”.
- Perintah “wc” digunakan untuk menghitung jumlah baris, kata dan karakter dari baris-baris masukan yang diberikan kepadanya. Untuk mengetahui berapa baris gunakan option - l, untuk mengetahui berapa kata, gunakan option -w dan untuk mengetahui berapa karakter, gunakan option -c. Jika salah satu option tidak digunakan, maka tampilannya adalah jumlah baris, jumlah kata dan jumlah karakter.
- Penggunaan “ls /etc | wc -l” digunakan untuk menampilkan jumlah baris, kata dan byte yang ada didalam direktori “/etc” secara lengkap.
- Perintah “cat” digunakan untuk menghasilkan output kelayar dan merupakan berasal dari input sebuah keyboard. Maka “cat > kelas1.txt” akan meminta input dari keyboard, lalu disimpan kedalam file “kelas1.txt”
- Perintah sort digunakan untuk mengurutkan data. Kemudian perintah “cat kelas1.txt kelas2.txt | sort” akan menampilkan isi file kelas1.txt dan kelas2.txt pada layer secara terurut.
- Perintah “cat kelas1.txt kelas2.txt > kelas.txt” akan menjadikan isi dari file kelas1.txt dan kelas2.txt sebagai input dan memasukannya ke file kelas.txt.
- Perintah uniq digunakan untuk menghilangkan duplikasi. Dan perintah “cat kelas.txt | sort | uniq” akan menampilkan isi dari file “kelas.txt” ke layer secara terurut dan tanpa duplikasi

Latihan:

1. Lihat daftar secara lengkap pada direktori aktif, blokir tampilan standard output ke file baru. Dengan mengetikkan:


```
ls -l 1> baru
cat < baru
```

Analisislah hasil diatas

```
root@bagusok:~# ls -l 1>baru
root@bagusok:~#
root@bagusok:~# cat < baru
total 4
-rw-r--r-- 1 root root    0 Feb 27 06:45 baru
drwxr-xr-x 2 root root 4096 Feb 27 06:45 test
root@bagusok:~# |
```

2. Lihat daftar secara lengkap pada direktori /etc/passwd, belokkan tampilan standard output ke file baru tanpa menghapus file baru sebelumnya.

```
ls -l /etc/passwd >> baru
cat < baru
```

Analisislah hasil diatas

```
root@bagusok:~# ls -l /etc/passwd >> baru
root@bagusok:~# cat < baru
total 4
-rw-r--r-- 1 root root    0 Feb 27 06:45 baru
drwxr-xr-x 2 root root 4096 Feb 27 06:45 test
-rw-r--r-- 1 root root 1966 Feb 27 06:27 /etc/passwd
root@bagusok:~# |
```

3. Urutkan file baru dengan cara membelokkan standard input.

```
Sort < baru
```

Analisislah hasil diatas

```
root@bagusok:~# sort < baru
drwxr-xr-x 2 root root 4096 Feb 27 06:45 test
-rw-r--r-- 1 root root    0 Feb 27 06:45 baru
-rw-r--r-- 1 root root 1966 Feb 27 06:27 /etc/passwd
total 4
root@bagusok:~# |
```

4. Urutkan file baru dengan cara membelokkan standard input dan standard output ke file baru urut.

```
Sort < baru > baru.urut
Sort < baru.urut
```

Analisislah hasil diatas

```
root@bagusok:~# sort < baru > baru.urut
root@bagusok:~# sort < baru.urut
drwxr-xr-x 2 root root 4096 Feb 27 06:45 test
-rw-r--r-- 1 root root    0 Feb 27 06:45 baru
-rw-r--r-- 1 root root 1966 Feb 27 06:27 /etc/passwd
total 4
root@bagusok:~# |
```

5. Buatlah direktori Latihan 2 sebanyak 2 kali dan belokkan standard error ke file rmdirerror.txt.

```
root@bagusok:~# cat rmdirerror.txt
mkdir: cannot create directory 'latihan': File exists
root@bagusok:~#
```

6. Urutkan kalimat berikut :

Jakarta

Bandung

Surabaya

Padang

Palembang

Lampung

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

```
root@bagusok:~# cat | sort <@@@
> Jakarta
> Bandung
> Surabaya
> Padang
> Palembang
> Lampung
> @@@
Bandung
Jakarta
Lampung
Padang
Palembang
Surabaya
```

7. Hitung jumlah baris, kata dan karakter dari file baru.urut dengan menggunakan filter dan tambahkan data tersebut ke file baru.

```
Surabaya
root@bagusok:~# wc -l < baru.urut >> a.txt && wc -w < baru.urut >> a.txt && wc -c <
baru.urut >> a.txt
root@bagusok:~# cat a.txt
6
6
50
root@bagusok:~#
```

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"
```

Analisislah hasil diatas

```
root@bagusok:~# cat > hello.txt
dog cat
cat duck
dog chicken
chicken duck
chicken cat
dog duck
root@bagusok:~# cat hello.txt | sort | uniq
cat duck
chicken cat
chicken duck
dog cat
dog chicken
dog duck
root@bagusok:~# cat hello.txt | grep "dog" | grep -v "cat"
grep: "cat": No such file or directory
root@bagusok:~# cat hello.txt | grep "dog" | grep -v "cat"
grep: "cat": No such file or directory
root@bagusok:~# ls
a.txt baru.urut hello.txt
root@bagusok:~# cat
^C
root@bagusok:~# cat hello.txt
dog cat
cat duck
dog chicken
chicken duck
chicken cat
dog duck
root@bagusok:~# cat hello.txt |grep "dog"
dog cat
dog chicken
dog duck
root@bagusok:~# cat hello.txt |grep "dog" | grep -v "cat"
dog chicken
dog duck
root@bagusok:~# |
```