



IKO31204
Pemrograman Sistem
Jilid 3: Scripting B

Fakultas Ilmu Komputer - Universitas Indonesia
Sep 2012

topik

Praktik membuat script
Contoh-contoh scripting

membuat script

1) membuat berkas script

```
shell> vi suatu-script.sh
```

2) mendeklarasikan bahwa
"suatu-script.sh" dapat di
Eksekusi di atas shell

Hint: baca ***man chmod*** !!!

```
shell> chmod +x suatu-script.sh
```

Isi dari *suatu-script.sh*

```
#!/bin/bash
```

```
echo "Ini percobaan pertama"
```

Mengeksekusi script

pemanggilan

```
shell> ./suatu-script.sh
```

Atau

```
shell> bash suatu-script.sh
```

Contoh soal #1

*Menjumlahlah berapa total byte dari file **apache.log** (20000 baris) antara baris 335 dan 14105 ? kemudian dimasukkan dalam berkas bernama **total.txt***

```
#!/bin/bash
```

```
head -n 14105 < apache-log.txt | tail -n +335 >  
proses.txt
```

```
awk 'BEGIN{  
total = 0;  
}  
{  
sub(/^[^"]*" [^"]*" [^ ]* /, "");  
total += $1;  
}  
END {  
printf("%d\n", total);}' proses.txt > total.txt
```

Contoh soal #2

Mendapatkan berapa jumlah IP yang Unik dari offset data pada soal #1, kemudian dimasukkan dalam berkas bernama unik.txt


```
#!/bin/bash
```

```
head -n 14105 < apache-log.txt | tail -n +335 >  
proses.txt
```

```
awk '{  
printf("%s\n", $1);  
' proses.txt | sort -u | wc -l > unik.txt
```

Contoh soal #3

*Carilah berkas yang paling **sering** didownload, lengkap dengan absolute pathnya dari offset data pada poin 2. Apabila terdapat lebih dari satu berkas yang sama seringnya, maka masukkan salah satu nama berkas tersebut. Masukkan nama berkas (dengan absolute pathnya) kedalam berkas bernama **sering.txt***

```
#!/bin/bash
```

```
head -n 14105 < apache-log.txt | tail -n +335 >  
proses.txt
```

```
grep GET proses.txt | awk 'BEGIN{  
namaBerkas = "";  
jumlahDL = 0;  
}  
{  
sub (/^[^"]*"GET /, "");  
download[$1]++  
}  
END {  
for(DL in download){  
if(download[DL]>jumlahDL){  
jumlahDL = download[DL];  
namaBerkas = DL; }}  
printf("%s\n", namaBerkas);} ' > sering.txt
```

Contoh soal #4

Membuat laporan total BYTE dan IP yang paling banyak download pada MENIT tersebut, dari offset data pada poin 2 KHUSUS untuk HTTP/1.1 Response 200. Masukkan informasi jam:menit SPASI total-byte-per-menit SPASI IP-yang-paling-banyak-download kedalam berkas bernama laporan.txt, Cth: file bernama laporan.txt berisi

09:54 274432 152.118.212.43

09:55 1992243 152.118.80.2

09:56 45644 152.118.21.55

Hint soal #4

- 1. Total Byte per menit adalah total byte yang terjadi pada jam-menit tersebut*
- 2. HTTP/1.1 Response 200 adalah string **HTTP/1.1" 200***
- 3. Agar mudah, gunakan Array dengan index asosiatif "**jam:menit IP**"*

```
grep "HTTP/1.1\" 200" proses.txt | awk 'BEGIN{
    IPMaks= "";
    jumlahDL = 0;
} {
    sub(/-[^:]*:/, "");
    sub(/:[0-9][0-9] [^"]*" [^"]*" [^ ]*/, "");
    total[$2] += $3;
    IP[$2, $1] += $3;
} END {
    for(jam in total) {
        IPMaks = "";
        jumlahDL = 0;
        for(pengguna in IP) {
            if((jam == substr(pengguna, 1, 5)) && (IP[pengguna]
> jumlahDL)) {
                IPMaks = substr(pengguna, 6);
                jumlahDL = IP[pengguna];
            }
        }
        printf("%s %d %s\n", jam, total[jam], IPMaks);
    }
}' | sort -k1 -n > laporan.txt
```

Contoh soal #5

Buat script bernama deteksi.sh untuk mendeteksi kabel jaringan tersambung / terputus. Behaviournya sbb:

- 1. Apabila tersambung menampilkan “TERSAMBUNG”, apabila jaringan putus menampilkan “PUTUS”.*
- 2. Interval pemeriksaan 1 detik sekali.*
- 3. Tulisan “TERSAMBUNG” dan “PUTUS” hanya tampil kalau ada perubahan saja*

Hint soal #5

- 1. gunakan while true BASH, if-else-fi*
- 2. dmesg, cut, sleep*
- 3. cari kata-kata “link up”/“link-down”
dari kernel message*


```
#!/bin/bash

STATUS="awal"

while true;
do
    LAST=`dmesg | grep e1000 | tail -n 1 | cut -d " "
-f 7`
    if [ $STATUS != $LAST ]
    then
        STATUS=$LAST
        if [ $STATUS == "Down" ]
        then
            echo "PUTUS"

        else
            echo "TERSAMBUNG"
        fi
    fi
    sleep 1
done
```

tanya jawab