

Nama : Aisyah Kirana Putri Isyanto

NPM : 21083010065

Kelas : Sistem Operasi – A

SHELL SCRIPTING

TUGAS 1

Mengerjakan seluruh tutorial yang ada pada file Shell Scripting.pdf

1. String

Tipe data string adalah tipe data yang digunakan untuk menyimpan teks. Semua teks tersebut diapit oleh tanda petik satu ("') maupun tanda pentik dua ("""). Pada contoh di bawah, kita buat terlebih dahulu file dengan nama "string.sh" menggunakan command nano.

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop
Berkas Sunting Tampilan Cari Terminal Bantuan
GNU nano 6.2                                     string.sh
a='ini single-quoted\n'
b="ini double-quoted\n"
echo -e ,a=$a,b=$b
```

Lalu, kita cetak file tersebut dengan fungsi bash maka output yang dihasilkan adalah sebagai berikut

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash string.sh
,a=ini single-quoted
,b=ini double-quoted
```

2. Array

Merupakan jenis variabel yang mampu menyimpan banyak nilai dalam 1 variabel. Tipe data ini dapat digunakan oleh semua tipe data baik string, integer, maupun konstanta. Seperti contoh di bawah yakni membuat array yang berisi string.

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop
Berkas Sunting Tampilan Cari Terminal Bantuan
GNU nano 6.2                                     array.sh
#!/bin/bash

#deklarasi array
distroLinux=("Mint" "Ubuntu" "Kali" "Arch" "Debian")

#random distro
let pilih=$RANDOM%5

#eksekusi
echo "Saya Memilih Distro $pilih, ${distroLinux[$pilih]} !"

aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash array.sh
Saya Memilih Distro 0, Mint !
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash array.sh
Saya Memilih Distro 4, Debian !
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash array.sh
Saya Memilih Distro 4, Debian !
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash array.sh
Saya Memilih Distro 1, Ubuntu !
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash array.sh
Saya Memilih Distro 2, Kali !
```

3. Integer

Tipe data ini digunakan untuk menyimpan bilangan bulat. Kita akan membuat variabel dengan tipe data integer lalu menampilkannya dengan perintah echo. Kita juga bisa menampilkan tipe data dari variabel tersebut dengan perintah bash.

The terminal window shows the following sequence:

- The title bar says "aisyah@aisyah-VirtualBox: ~/Tugas-Sisop".
- The menu bar includes "Berkas", "Sunting", "Tampilan", "Cari", "Terminal", and "Bantuan".
- The nano editor window displays the contents of "integer.sh":

```
GNU nano 6.2
integer.sh *
a=123
b=456
let c=a+b
echo $c
```
- The terminal command line shows:

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash integer.sh
```
- The output of the command is:

```
579
```

4. Penggunaan Operasi Matematika (op_mtk.sh)

Untuk melakukan perhitungan seperti pengurangan, perkalian, dll dalam linux dengan menggunakan operasi perhitungan matematika. Ada beberapa catatan yang perlu diingat ketika melakukan operasi perhitungan aritmatika.

Ada 3 cara yang dapat digunakan untuk melakukan operasi matematika, yaitu:

- Menggunakan perintah built-in let (membuat variabel sama dengan ekspresi).
- Menggunakan perintah eksternal expr atau awk (mencetak hasil ekspresi).
- Menggunakan perintah substitusi \$((ekspresi)) (mengembalikan hasil ekspresi).

Berikut ini operator-operator aritmatika, yaitu:

No	Nama Operator	Simbol
1	Penjumlahan	+
2	Pengurangan	-
3	Perkalian	*
4	Pembagian	/
5	Modulus (sisa pembagian)	%
6	Menempatkan nilai di sisi kanan ke variabel di sisi kiri	=
7	Membandingkan 2 nilai yang sama	==
8	Membandingkan 2 nilai yang tidak sama	!=

Berikut ini adalah contoh penggunaan simbol operator aritmatika

```

aisyah@ais
Berkas Sunting Tampilan Cari Terminal Bantu
GNU nano 6.2

a=20
b=8

#memakai let
let jumlah=$a+$b
let kurang=$a-$b
let kali=$a*$b

#memakai expr
bagi= expr $a / $b

#memakai perintah substitusi $((ekspresi))
mod=$((a % b))

echo "a + b = $jumlah"
echo "a - b = $kurang"
echo "a * b = $kali"
echo "a / b = $bagi"
echo "a % b = $mod"

b=$a

echo "a = $a"
echo "b = $b"

```

aisyah@aisyah-VirtualBox:~/Tugas-Sisop\$ bash op_mtk.sh

```

a + b = 28
a - b = 12
a * b = 160
a / b = 2
a % b = 4
a = 20
b = 20

```

5. Pembuatan Input

Untuk membaca inputan dari user gunakanlah "read" dengan format penulisan "read nama_var"

```

aisyah@ais
Berkas Sunting Tampilan Cari Terminal Bantu
GNU nano 6.2
#!/bin/bash
clear

echo -n "Hallo, masukkan nama anda: ";
read nama;
echo "Selamat datang $nama";

```

Inputan dari user tersimpan ke dalam variabel nama. Maka, hasilnya

```

Hallo, masukkan nama anda : Aisyah Kirana
Selamat datang Aisyah Kirana
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ 

```

6. Penampilan Output

Echo digunakan untuk menampilkan output dengan syntax sebagai berikut:

- o Menampilkan teks biasa echo "teks"

o Menampilkan isi dari sebuah variabel echo \$nama_var

Catatan:

Jika ingin menggunakan new line character (\n) pada echo, gunakan echo -e "teks\n teks"

```
GNU nano 6.2                                     output1.sh *
#!/bin/bash
matakuliah="Sistem Operasi"

echo "Siapa namamu?"
read nama
echo -e "\nHai $nama!\nSelamat datang di praktikum $matakuliah:))"
```

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash output1.sh
Siapa namamu?
Aisyah Kirana Putri Isyanto

Hai Aisyah Kirana Putri Isyanto!
Selamat datang di praktikum Sistem Operasi:))
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$
```

Selain menggunakan output "echo", output bisa juga dilakukan dengan menggunakan *printf* seperti pada bahasa pemrograman C. Dengan format control sebagai berikut

- %d untuk format integer
- %o untuk format octal
- %f untuk format float atau decimal
- %x untuk format hexadecimal

```
Berkas Sunting Tampilan Cari Terra
GNU nano 6.2
#Inisialisasi Var
a=55;
b=4;
distroLinux="Ubuntu 19.04 LTS";
let c=a%b;

#Mencetak Printf
printf "OS : $distroLinux \n";
printf "$c \n";
printf "%.2f float \n" $a;
printf "%.1f float \n" $a
```

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash output2.sh
OS : Ubuntu 19.04 LTS
3
55,00 float
55,0 float
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$
```

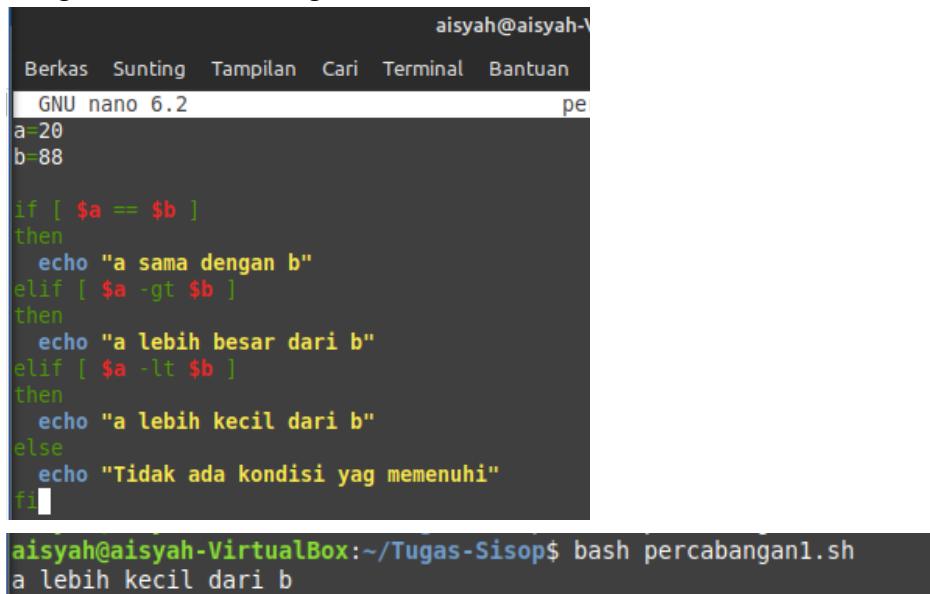
7. Percabangan

Tipe ini digunakan untuk memutuskan apakah akan menjalankan sepotong kode atau tidak berdasarkan kondisi yang telah ditentukan. Sebelum melangkah ke percabangan alangkah baiknya terlebih dahulu mengetahui dasar operasi relasional yang mana biasanya digunakan bersama dengan conditional statements.

No	Operator	Deskripsi
1	-eq	Memeriksa apakah nilai kedua operan sama (==)
2	-ne	Memeriksa apakah nilai kedua operan tidak sama (!=)
3	-gt	Memeriksa apakah nilai operan kiri lebih besar daripada operan kanan (>)
4	-lt	Memeriksa apakah nilai operan kiri lebih kecil daripada operan kanan (<)
5	-ge	Memeriksa apakah nilai operan kiri lebih besar atau sama dengan operan kanan (>=)
6	-le	Memeriksa apakah nilai operan kiri lebih kecil atau sama dengan operan kanan (<=)

- If Else

Terkadang kita ingin melakukan serangkaian tindakan tertentu jika pernyataan itu benar, dan serangkaian tindakan lain jika pernyataan itu salah. Kita bisa mengakomodasi ini dengan mekanisme *else*.



```

aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash percabangan1.sh
a lebih kecil dari b

```

```

aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ cat percabangan1.sh
#!/bin/bash
a=20
b=88

if [ $a == $b ]
then
    echo "a sama dengan b"
elif [ $a -gt $b ]
then
    echo "a lebih besar dari b"
elif [ $a -lt $b ]
then
    echo "a lebih kecil dari b"
else
    echo "Tidak ada kondisi yang memenuhi"
fi

```

Percabangan 2

- Case Statements

Terkadang kita mungkin ingin mengambil jalur yang berbeda berdasarkan variabel yang cocok dengan serangkaian pola. Kita bisa menggunakan serangkaian pernyataan *if* dan *elif* tapi itu akan segera berkembang menjadi tidak efektif. Untungnya ada pernyataan kasus yang dapat membuat segalanya lebih bersih

```
#!/bin/bash

printf "Jajan apa yang kamu suka ?\n"
printf "pentol ?\n"
printf "batagor ?\n"
printf "cireng ?\n"

read jajan
case "$jajan" in
    "pentol")
        echo "Pentol buk mah wenak slur!"
        ;;
    "batagor")
        echo "Batagore mas budi mantap bat"
        ;;
    "cireng")
        echo "Cirenge kantin rasane unch-unch"
        ;;
    *)
        echo "Makanan yanng kamu suka gaenak wkwk"
        ;;
esac
```

Output yang dihasilkan apabila memilih “cireng”

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash percabangan2.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
cireng
Cirenge kantin rasane unch-unch
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$
```

Output yang dihasilkan apabila memilih “batagor”

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash percabangan2.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
batagor
Batagore mas budi mantap bat
```

TUGAS 2

Buatlah program percabangan sederhana aritmatika yang menerapkan beberapa konsep pemrograman bash seperti diatas!

Skrip Bash harus menampilkan nilai huruf siswa berdasarkan rata-rata numerik dari nilai tes gabungan. Script harus meminta pengguna untuk memasukkan nilai numerik (nilai tes). Terapkan struktur loop ke bagian skrip tempat pengguna diminta memasukkan nilai numerik.

Nilai huruf yang sesuai untuk kelas angka, harus dihitung dengan menggunakan skala :

90 hingga 100: Menampilkan nilai huruf "A"

80 hingga 89: Menampilkan nilai huruf "B"

70 hingga 79: Menampilkan nilai huruf " C"

65 hingga 69: Menampilkan nilai huruf "D"

0 hingga 64: Menampilkan nilai huruf "F"

```
GNU nano 6.2                                     arithmetic.sh
echo -e "\nPlease enter number grades for the students in one line like 23 44 57 89"
#input grades for students
declare -i stunum=1 n1 n2 n3 n4 sum avg N slimit
echo -ne "\nNumber of Students : "
read N

slimit=N+1

while [ $stunum -lt $slimit ]
do
    echo -ne "Student #$stunum : "

    #Read the four number grades
    read n1 n2 n3 n4

    #Compute the average
    sum=n1+n2+n3+n4
    avg=sum/4
    echo -n "Average = $avg, Grade ="

    #Compute the corresponding letter grade now
    if [ $avg -lt 65 ]
    then
        echo -e " F\n"
    elif [ $avg -lt 70 ]
    then
        echo -e " D\n"
    elif [ $avg -lt 80 ]
    then
```

```
        echo -e " C\n"
elif [ $avg -lt 90 ]
then
        echo -e " B\n"
else
        echo -e " A\n"
fi

#Show the Student Number
stunum=$stunum+1
done
```

[baris 41/41 (100%). kol 1/ 1 ()]

Kemudian kita panggil file dengan command “bash”. Setelah itu muncul output yang mngharuskan kita mengisi banyaknya murid yang akan dihitung nilainya. Lalu, kita masukkan nilai mereka satu per satu dan tekan ‘enter’ untuk melihat rata-rata nilai mereka.

```
aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ bash arithmetic.sh
Please enter number grades for the students in one line like 23 44 57 89
Number of Students : 
```

```
Please enter number grades for the students in one line like 23 44 57 89
Number of Students : 3
Student #1 : 88 89 84 90
Average = 87, Grade = B

Student #2 : 78 89 85 84
Average = 84, Grade = B

Student #3 : 90 75 64 89
Average = 79, Grade = C

aisyah@aisyah-VirtualBox:~/Tugas-Sisop$ 
```