

Nama : Luqna Aziziyah

NPM : 21083010020

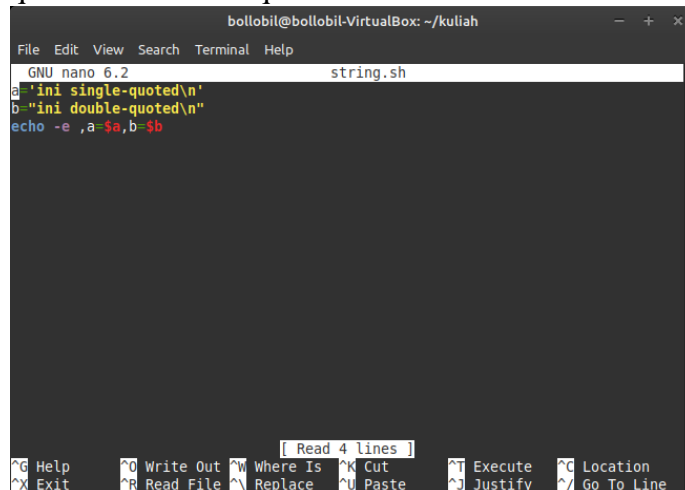
Kelas : Sistem Operasi B

Percobaan String (single-quoted) dan (double-quote):

A quote string atau string yang dikutip adalah sebuah konstanta yang diapit oleh sebuah tanda petik. Tanda petik sendiri memiliki 2 jenis, yakni single-quoted ('...') dan double quote ("...").

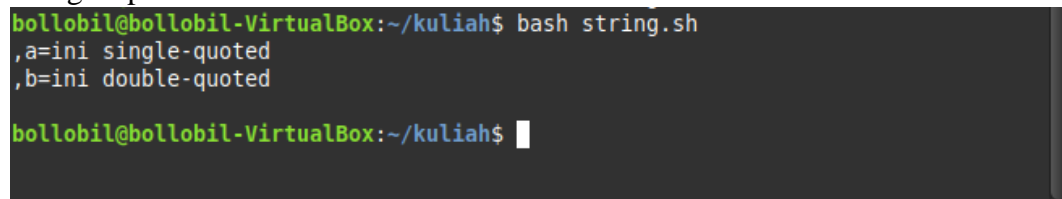
a. Nano version

1. Membuat nano file dengan diberi nama string.sh
2. Disini saya mencoba menjalankan sebuah pemrograman mengenai string single quoted dan double quoted.



```
bollobil@bollobil-VirtualBox: ~/kuliah
GNU nano 6.2 string.sh
a='ini single-quoted\n'
b='ini double-quoted\n'
echo -e ,a-$a,b-$b
```

3. Dapat dilihat pada gambar diatas, bahwa a menggunakan single quoted dan b menggunakan double quoted.
4. \n pada script diatas berfungsi untuk 'enter' bagi kalimat yang selanjutnya.
5. Echo berfungsi sebagai "pemanggil" untuk output yang di inginkan.
6. Setelah selesai tidak lupa untuk **ctrl s** agar file nano kita tersimpan dan **ctrl x** untuk keluar dari file nano.
7. Selanjutnya untuk mengetahui output yang dihasilkan kita dapat menetik bash string.sh pada terminal.



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

bollobil@bollobil-VirtualBox:~/kuliah$
```

8. Output yang akan keluar akan terlihat seperti diatas

b. Non nano version (terminal)

1. Tidak jauh berbeda pada versi file nano dan versi terminal, yang membedakan hanyalah tempatnya

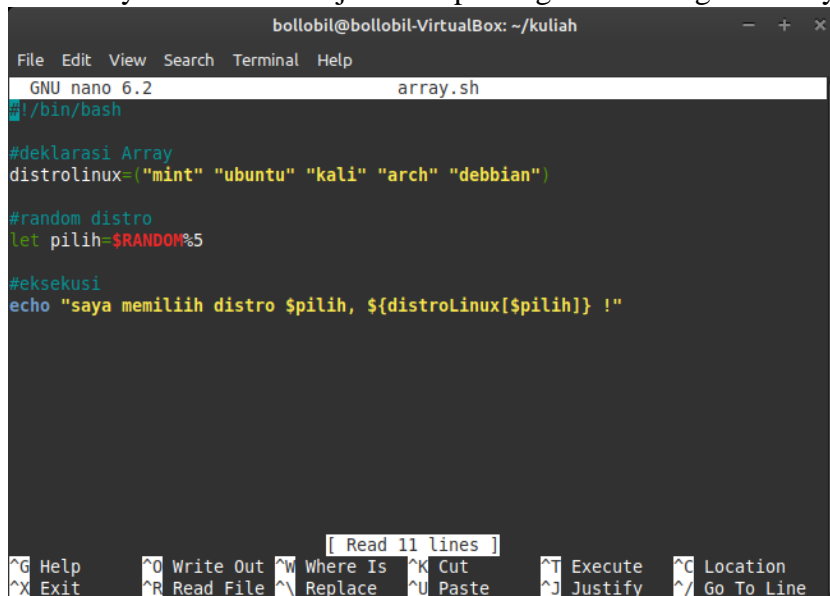
```
bollobil@bollobil-VirtualBox:~/kuliah$ a='ini single-quoted'
bollobil@bollobil-VirtualBox:~/kuliah$ b="ini double-quoted"
bollobil@bollobil-VirtualBox:~/kuliah$ echo -e ,a=$a,b=$b
,a=ini single-quoted,b=ini double-quoted
bollobil@bollobil-VirtualBox:~/kuliah$ a='ini single-quoted\n'
bollobil@bollobil-VirtualBox:~/kuliah$ b="ini double-quoted\n"
bollobil@bollobil-VirtualBox:~/kuliah$ echo -e ,a=$a,b=$b
,a=ini single-quoted
,b=ini double-quoted
bollobil@bollobil-VirtualBox:~/kuliah$
```

2. Pada script diatas, dapat dilihat sebuah perbedaan yang signifikan, itu disebabkan adanya pengaruh dari \n yang memiliki arti “enter” bagi kalimat selanjutnya
3. Untuk mengetahui output yang dikeluarkan, kita dapat langsung menggunakan echo.

Percobaan Array

Array ialah koleksi dari beberapa data yang mirip/sejenis dengan letak yang berdekatan yang direferensikan dengan index atau lainnya.

1. Membuat sebuah file dengan menggunakan nano dan diberi nama array.sh
2. Disini saya mencoba menjalankan pemrograman mengenai array



```
bollobil@bollobil-VirtualBox: ~/kuliah
File Edit View Search Terminal Help
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]} !"

[ Read 11 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

3. tanda `#!/bin/bash` dalam script tes adalah perintah yang diterjemahkan ke kernel linux untuk mengeksekusi path yang disertakan dalam hal ini program **bash** pada direktory `/bin`, sebenarnya tanpa mengikuti baris tersebut anda tetap dapat mengeksekusi script **bash**, dengan catatan **bash** adalah shell aktif.
4. Seperti yang terlihat pada gambar diatas, pada “distroLinux” saya mendeklarasikan sebuah elemen yang berisi dari nama jenis linux, hal tersebut dapat dikatakan sebagai sebuah array
5. Di syntax lanjutannya saya memprogram agar output yang dikeluarkan dapat memilih sebuah array secara random dari array yang telah dibuat

6. untuk mengeksekusi sebuah program agar dapat berjalan dan mengeluarkan output yang saya inginkan, maka dari itu saya menggunakan echo
7. tidak lupa untuk **ctrl s** agar file yang kita buat dapat tersimpan dan **ctrl x** untuk keluar dari file nanto tsb.
8. Lalu dapat kita lihat hasil dari program yang telah kita buat dengan menggunakan bash array.sh

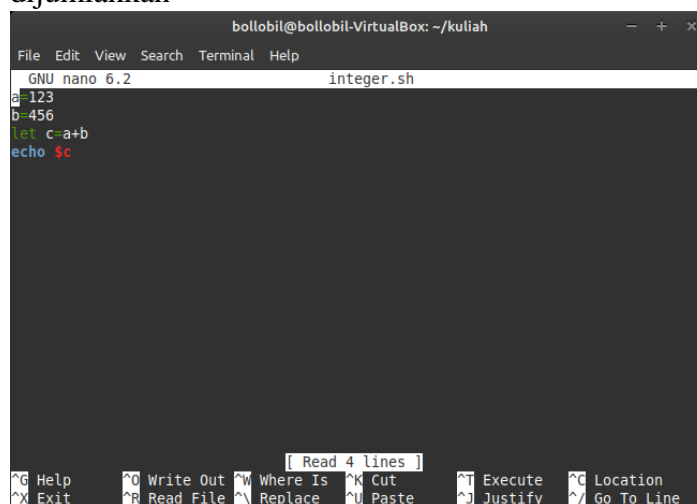
```
bollobil@bollobil-VirtualBox:~/kuliah$ bash array.sh
saya memilih distro 2, !
bollobil@bollobil-VirtualBox:~/kuliah$ bash array.sh
saya memilih distro 1, !
bollobil@bollobil-VirtualBox:~/kuliah$ bash array.sh
saya memilih distro 0, !
bollobil@bollobil-VirtualBox:~/kuliah$ bash array.sh
saya memilih distro 2, !
bollobil@bollobil-VirtualBox:~/kuliah$ bash array.sh
saya memilih distro 4, !
bollobil@bollobil-VirtualBox:~/kuliah$ bash array.sh
saya memilih distro 4, !
bollobil@bollobil-VirtualBox:~/kuliah$
```

9. Hasil atau output yang dikeluarkan akan seperti gambar diatas, program yang dilakukan dapat dikatakan berhasil karena output yang dihasilkan pada setiap pemanggilan memiliki hasil yang berbeda/random.

Percobaan Integer

Integer sendiri memiliki arti sebuah bilangan bulat, yang dapat terdiri dari bilangan negative, 0, dan bilangan positif. Bilangan pecahan dan decimal tidak termasuk kedalam bilangan integer.

1. Membuat file nano dengan diberi nama test_integer.sh
2. Pada kali ini saya membuat sebuah pemrograman mengenai bilangan Integer yang dijumlahkan



```
bollobil@bollobil-VirtualBox: ~/kuliah
GNU nano 6.2 integer.sh
a=123
b=456
let c=a+b
echo $c
```

3. Pada gambar diatas dapat dilihat bahwa a mendeklarasikan angka 123, dan b mendeklarasikan angka 456.
4. Selanjutnya, c mendeklarasikan penjumlahan antara a dan b
5. Untuk menghasilkan output pemrograman yang telah dibuat, dapat menggunakan echo
6. Setelah kita simpan dan keluar dari file nano, kita dapat melihat hasil outputnya dengan menggunakan bash integer.sh dan akan keluar seperti gambar dibawah berikut.

```
bollobil@bollobil-VirtualBox: ~/kuliah
File Edit View Search Terminal Help
bollobil@bollobil-VirtualBox:~/kuliah$ bash integer.sh
579
bollobil@bollobil-VirtualBox:~/kuliah$
```

Percobaan operasi aritmatika

Operasi aritmatika adalah operasi dasar dari matematika, yang terdiri dari penjumlahan, pengurangan, perkalian, dan sebagainya. Dalam bash beberapa operasinya dapat dilakukan dengan menggunakan *let* dan dilanjutkan dengan pendeklarasian yang ingin dilakukan.

Operasi aritmatika dalam bash terdiri

- + untuk penjumlahan
- - untuk pengurangan
- * untuk perkalian
- / untuk pembagian
- % untuk modulus/melihat sisa dari hasil bagi
- = menempatkan nilai di sisi kanan ke variable di sisi kiri
- == membandingkan 2 nilai yang sama
- != membandingkan 2 nilai yang tak sama

1. Membuat sebuah file nano dan diberi nama aritmatika.sh
2. Dalam pemrograman kali ini, saya membuat operasi aritmatika yang ada pada bash.

```
bollobil@bollobil-VirtualBox: ~/kuliah
File Edit View Search Terminal Help
GNU nano 6.2 aritmatika.sh *
#!/bin/bash

a=15
b=7

#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"
```

3. Dapat dilihat pada gambar diatas, bahwa a mendeklarasikan angka 15 dan b mendeklarasikan angka 7

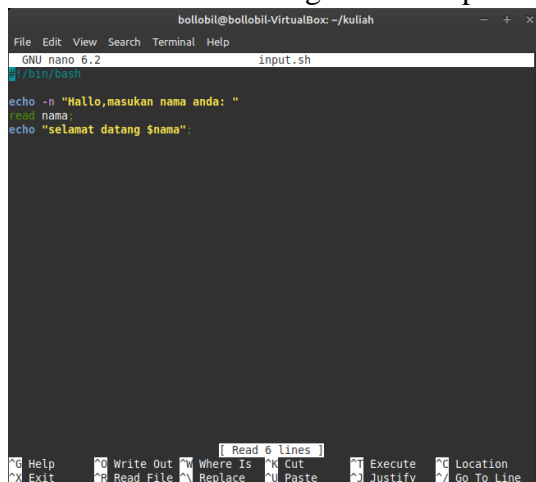
4. Pada pengoprasian aritmatika penjumlahan, pengurangan, dan perkalian kita dapat menggunakan *let* (*build-in*) dan untuk pembagian menggunakan *expr* atau *awk* (*perintah eksternal*) dan untuk modulus menggunakan syntax *perintah substitusi* *\$((ekspresi))* seperti diatas
5. Untuk memanggil pemrograman diatas, kita dapat menggunakan *echo*
6. Pada bagian *b=\$a* dapat diartikan bahwa deklarasi *b* yang sebelumnya angka 7, berubah menjadi sama seperti deklarasi *a* yakni angka 15.
7. Setelahnya kita dapat menyimpan file nano tsb dengan **ctrl s** dan untuk keluar dapat menggunakan **ctrl x**.
8. Output dari program yang telah kita buat dapat dilihat dengan menggunakan *bash aritmatika.sh*, dan akan terlihat seperti dibawah ini

```
bollobil@bollobil-VirtualBox:~/kuliah$ bash aritmatika.sh
a + b = 22
a - b = 8
a * b = 105
a / b = 2
a % b = 1
a = 15
b = 15
bollobil@bollobil-VirtualBox:~/kuliah$
```

Percobaan Input

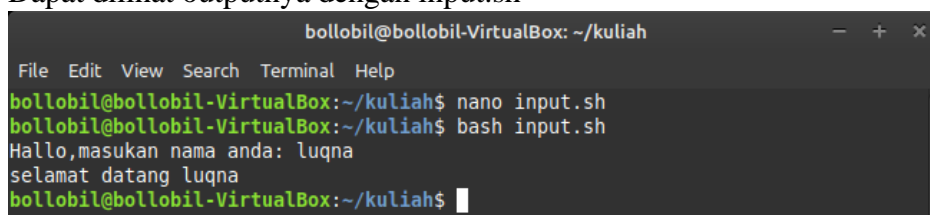
Input memiliki arti sesuatu data yang dimasukan oleh user setelah/saat proses output terjadi.

1. Membuat file nano dengan nama *input.sh*



```
bollobil@bollobil-VirtualBox: ~/kuliah
File Edit View Search Terminal Help
GNU nano 6.2 input.sh
~/bin/bash
echo -n "Hallo,masukan nama anda: "
read nama
echo "selamat datang $nama"
```

2. *Read nama* yang terdapat diatas memiliki fungsi agar user dapat menginput sebuah nama yang akan disimpan ke dalam variable, yang dimana variable tersebut terpanggil dalam sebuah *echo* lanjutannya
3. Setelahnya tidak lupa untuk menyimpan dan keluar dari file nano.
4. Dapat dilihat outputnya dengan *input.sh*



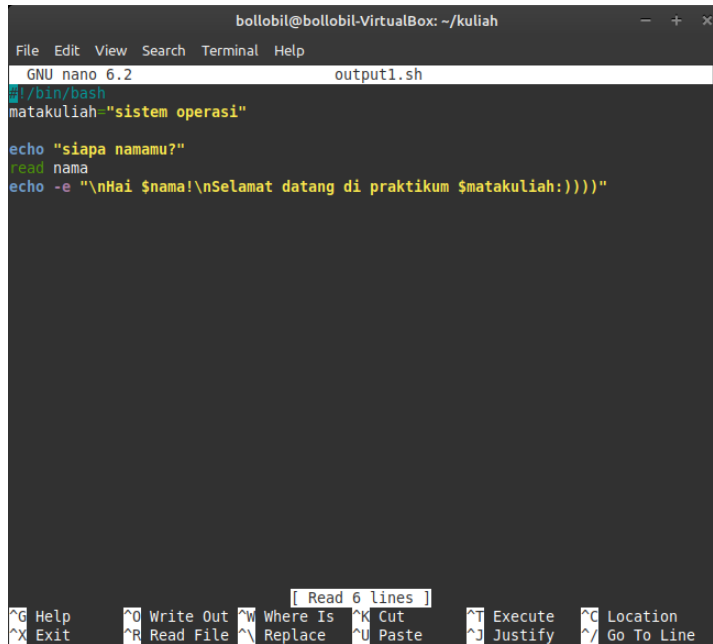
```
bollobil@bollobil-VirtualBox: ~/kuliah
File Edit View Search Terminal Help
bollobil@bollobil-VirtualBox:~/kuliah$ nano input.sh
bollobil@bollobil-VirtualBox:~/kuliah$ bash input.sh
Hallo,masukan nama anda: luqna
selamat datang luqna
bollobil@bollobil-VirtualBox:~/kuliah$
```

Percobaan output

Output sendiri memiliki arti hasil dari suatu proses, baik hasil berupa data maupun informasi yang telah diolah.

a. Output1

1. Membuat file nano dengan diberi nama output1.sh seperti pada gambar di bawah ini



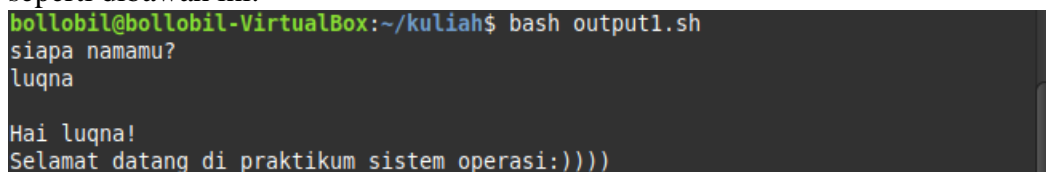
The screenshot shows a terminal window titled 'bollobil@bollobil-VirtualBox: ~/kuliah'. Inside, the GNU nano 6.2 editor is open with a file named 'output1.sh'. The script content is as follows:

```
#!/bin/bash
matakuliah="sistem operasi"

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

At the bottom of the editor, a status bar indicates '[Read 6 lines]' and provides keyboard shortcuts for various actions like Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

2. Di awal kita dapat melihat bahwa dalam matakuliah mendeklarasikan kalimat system operasi
3. Read nama yang terdapat diatas memiliki fungsi agar user dapat menginput sebuah nama yang akan disimpan ke dalam variable, yang dimana variable tersebut terpanggil dalam sebuah echo lanjutannya
4. Setelahnya, kita dapat melihat outputnya dengan bash output1.sh yang terlihat seperti dibawah ini.



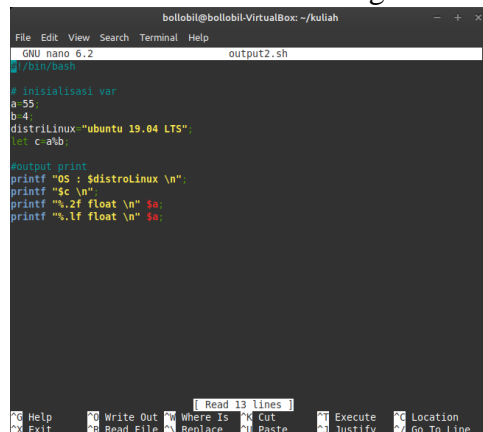
The screenshot shows a terminal window titled 'bollobil@bollobil-VirtualBox: ~/kuliah'. The user has executed the command 'bash output1.sh'. The output of the script is:

```
siapa namamu?
luqna

Hai luqna!
Selamat datang di praktikum sistem operasi:)))))
```

b. Output2

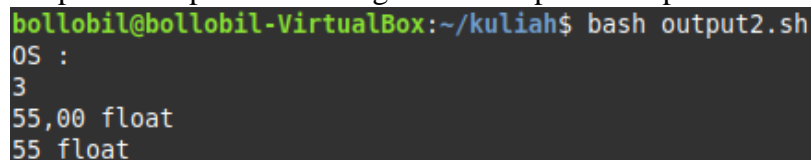
1. Membuat new file nano dengan nama output2.sh



```
bollobil@bollobil-VirtualBox: ~/kuliah
File Edit View Search Terminal Help
GNU nano 6.2 output2.sh
#!/bin/bash
#inisialisasi var
a=55
b=4
distrolinux="ubuntu 19.04 LTS"
c=a*b

#output print
printf "OS : $distrolinux \n"
printf "%c \n"
printf "%.2f float \n" $a
printf "%.1f float \n" $a
```

2. Script diatas menunjukkan hal pertama yang dilakukan ialah inisialisasi variable
3. Selanjutnya kita menggunakan operasi aritmatika yaitu modulus
4. Nama lain dari echo yaitu printf
5. %.2f float dapat diartikan bahwa hasil dari pengoperasiannya hanya akan terdapat 2 angka dibelakang koma
6. Tidak lupa untuk **ctrl s** sebelum **ctrl x** agar file yang kita buat tersimpan.
7. Output akan dapat dilihat dengan bash output2.sh seperti dibawah ini



```
bollobil@bollobil-VirtualBox:~/kuliah$ bash output2.sh
OS :
3
55,00 float
55 float
```

Percobaan percabangan

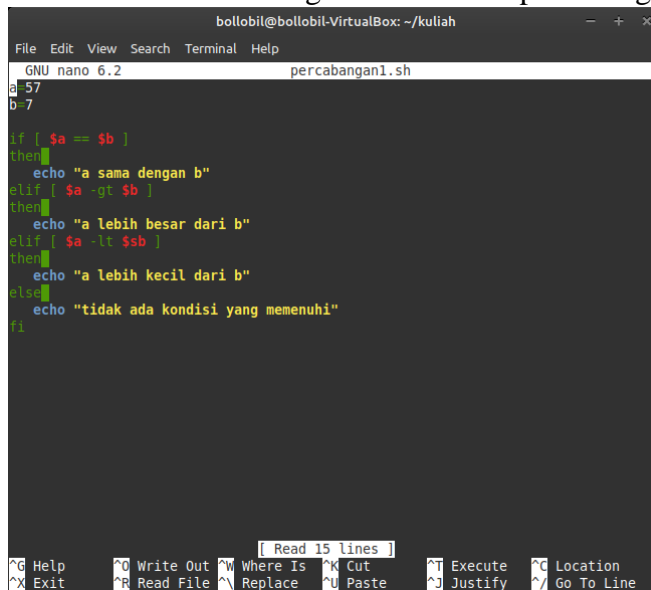
Percabangan adalah sebagai kontrol seleksi yang memungkinkan program untuk menjalankan suatu perintah berdasarkan kondisi tertentu.

Adapun no operator deskripsi

- -eq nilai kedua operan sama (==)
- -ne nilai kedua operan tidak sama (!=)
- -gt nilai operan kiri lebih besar dari kanan (>)
- -lt nilai operasi kanan lebih besar dari kiri (<)
- -ge nilai operan kiri lebih besar sama dengan dari kanan (>=)
- -le nilai operasi kanan lebih besar sama dengan dari kiri (<=)

a. Percabangan1

1. Membuat file nano dengan diberi nama percabangan1.sh

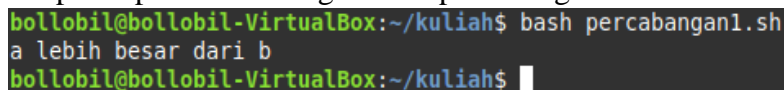


```
bollobil@bollobil-VirtualBox: ~/kuliah
GNU nano 6.2 percabangan1.sh
a=57
b=7

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

[ Read 15 lines ]
^G Help  ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit  ^R Read File  ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

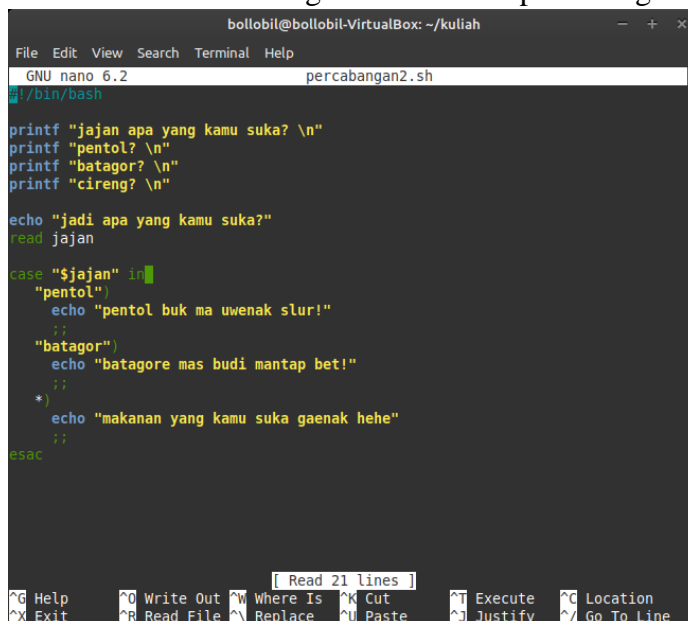
2. Keadaan diatas adalah sebuah kondisi perbandingan angka yang terjadi pada a dan b, dan semua keadaan yang terjadi apabila a dan b berubah nominalnya
3. Tidak lupa untuk **ctrl s** sebagai menyimpan file nano
4. Dan keluar dari file menggunakan **ctrl x**
5. Output dapat dilihat dengan bash percabangan1.sh



```
bollobil@bollobil-VirtualBox:~/kuliah$ bash percabangan1.sh
a lebih besar dari b
bollobil@bollobil-VirtualBox:~/kuliah$
```

b. Percabangan2

1. Membuat file nano dengan diberi nama percabangan2.sh



```
bollobil@bollobil-VirtualBox: ~/kuliah
GNU nano 6.2 percabangan2.sh
#!/bin/bash

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

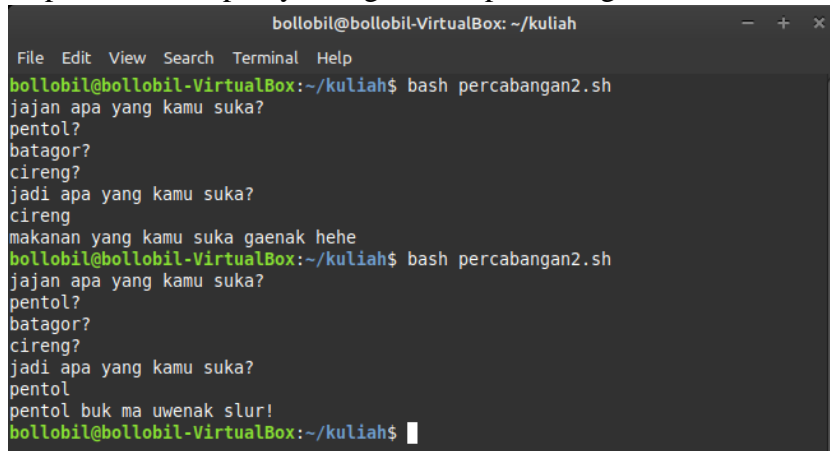
echo "jadi apa yang kamu suka?"
read jajan

case "$jajan" in
    "pentol")
        echo "pentol buk ma uwenak slur!"
        ;;
    "batagor")
        echo "batagore mas budi mantap bet!"
        ;;
    *)
        echo "makanan yang kamu suka gaenak hehe"
        ;;
esac

[ Read 21 lines ]
^G Help  ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit  ^R Read File  ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

2. Pada pemrograman diatas, dapat dilihat bahwa setelah terjadi penawaran mengenai jajan, selanjutnya terdapat sebuah pertanyaan yang dimana nantinya user dapat mengimput sebuah jawaban

3. Dalam case jajan, disana terdapat beberapa kemungkinan user menjawab yang dimana kita telah mempersiapkan balasan dari inputan user yang akan datang.
4. Dapat dilihat outputnya dengan bash percabangan2.sh



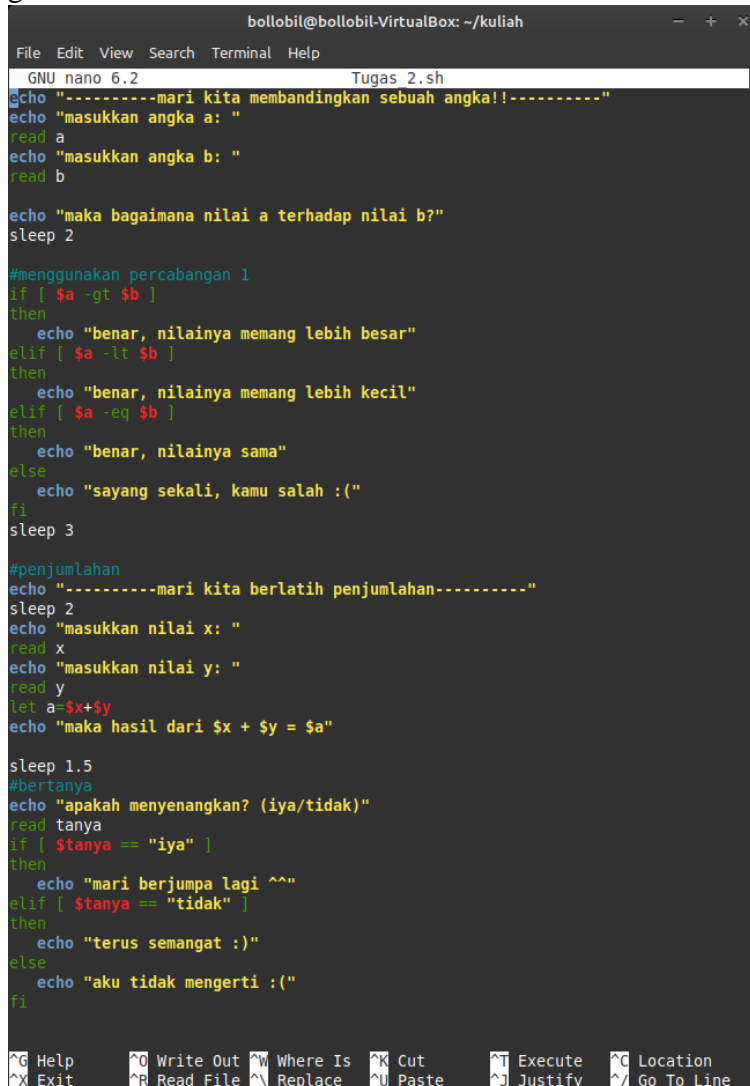
```
bollobil@bollobil-VirtualBox: ~/kuliah
File Edit View Search Terminal Help
bollobil@bollobil-VirtualBox:~/kuliah$ bash percabangan2.sh
jajan apa yang kamu suka?
pentol?
batagor?
cireng?
jadi apa yang kamu suka?
cireng
makanan yang kamu suka gaenak hehe
bollobil@bollobil-VirtualBox:~/kuliah$ bash percabangan2.sh
jajan apa yang kamu suka?
pentol?
batagor?
cireng?
jadi apa yang kamu suka?
pentol
pentol buk ma uwenak slur!
bollobil@bollobil-VirtualBox:~/kuliah$
```

5. Dapat dilihat bahwa inputan yang diberi oleh user akan memiliki jawaban yang berbeda pula.

Tugas

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

1. Membuat sebuah file nano, setelahnya diberi nama. Disini saya memberi nama Tugas_2.sh
2. Setelahnya saya membuat script bash yang diperintahkan. Disini saya memakai percobaan percabangan dan aritmatika penjumlahan. Seperti yang terlampir pada gambar.



```
bollobil@bollobil-VirtualBox: ~/kuliah
GNU nano 6.2 Tugas_2.sh
echo "-----mari kita membandingkan sebuah angka!!-----"
echo "masukkan angka a: "
read a
echo "masukkan angka b: "
read b

echo "maka bagaimana nilai a terhadap nilai b?"
sleep 2

#menggunakan percabangan 1
if [ $a -gt $b ]
then
    echo "benar, nilainya memang lebih besar"
elif [ $a -lt $b ]
then
    echo "benar, nilainya memang lebih kecil"
elif [ $a -eq $b ]
then
    echo "benar, nilainya sama"
else
    echo "sayang sekali, kamu salah :("
fi
sleep 3

#penjumlahan
echo "-----mari kita berlatih penjumlahan-----"
sleep 2
echo "masukkan nilai x: "
read x
echo "masukkan nilai y: "
read y
let a=$x+$y
echo "maka hasil dari $x + $y = $a"

sleep 1.5
#bertanya
echo "apakah menyenangkan? (iya/tidak)"
read tanya
if [ $tanya == "iya" ]
then
    echo "mari berjumpa lagi ^^"
elif [ $tanya == "tidak" ]
then
    echo "terus semangat :)"
else
    echo "aku tidak mengerti :("
fi

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^G Location
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

3. Pada script terdapat sleep dan angka di sebelahnya, itu dapat diartikan bahwa program yang akan terjadi selanjutnya akan tertunda selama beberapa saat tergantung angka yang disebutkan di sebelahnya. Disana saya hanya menggunakan hitungan detik.
4. Dengan script diatas, dapat kita hasilkan output seperti dibawah berikut.

```

bollobil@bollobil-VirtualBox:~/kuliah$ bash Tugas_2.sh
-----mari kita membandingkan sebuah angka!-----
masukkan angka a:
80
masukkan angka b:
65
maka bagaimana nilai a terhadap nilai b?
benar, nilainya memang lebih besar
-----mari kita berlatih penjumlahan-----
masukkan nilai x:
254
masukkan nilai y:
869
maka hasil dari  $254 + 869 = 1123$ 
apakah menyenangkan? (iya/tidak)
iya
mari berjumpa lagi ^^
bollobil@bollobil-VirtualBox:~/kuliah$ bash Tugas_2.sh
-----mari kita membandingkan sebuah angka!-----
masukkan angka a:
43
masukkan angka b:
98
maka bagaimana nilai a terhadap nilai b?
benar, nilainya memang lebih kecil
-----mari kita berlatih penjumlahan-----
masukkan nilai x:
555
masukkan nilai y:
4320
maka hasil dari  $555 + 4320 = 4875$ 
apakah menyenangkan? (iya/tidak)
tidak
terus semangat :)
bollobil@bollobil-VirtualBox:~/kuliah$

```

5. Terlihat bahwa output yang dihasilkan diatas tidak akan sama apabila angka atau jawaban yang di input tidak sama