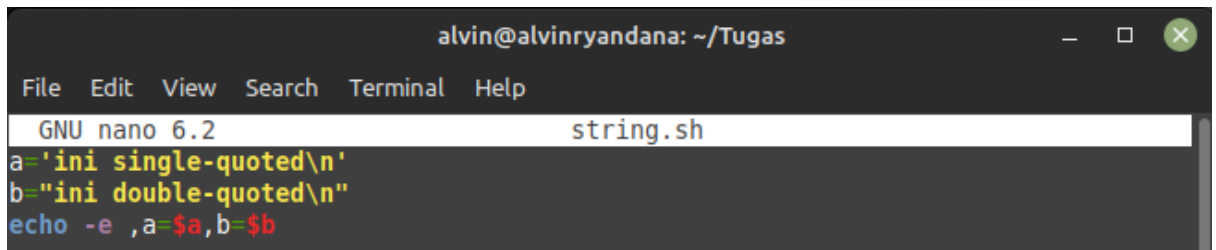


Nama : Alvin Ryan Dana  
NPM : 21083010035  
Kelas : Sistem Operasi B

## 1. Praktik Shell Scripting

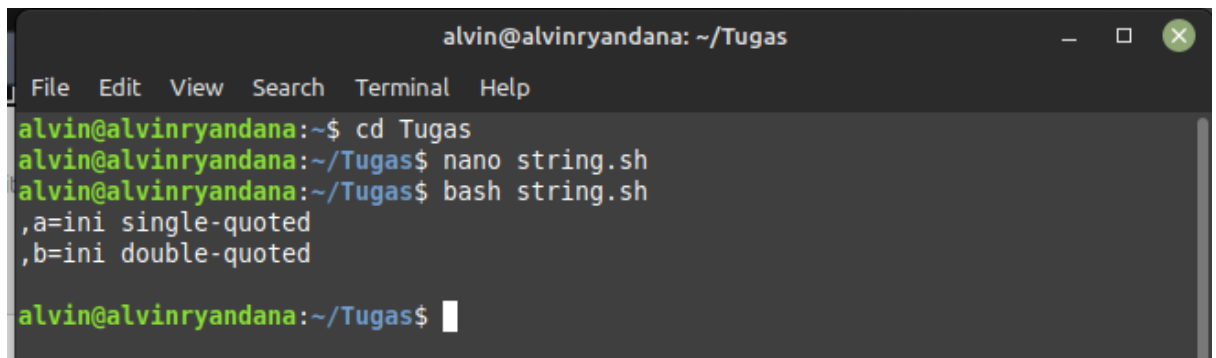
Untuk file bash akan saya simpan pada direktori Tugas bersama dengan laporan dan tugas sebelumnya

### 1. string



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 string.sh
a='ini single-quoted\n'
b="ini double-quoted\n"
echo -e ,a=$a,b=$b
```

membuat file string.sh menggunakan nano dengan syntax diatas. berikut ini adalah contoh penggunaan string single-quoted & double quoted



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano string.sh
alvin@alvinryandana:~/Tugas$ bash string.sh
,a=ini single-quoted
,b=ini double-quoted

alvin@alvinryandana:~/Tugas$
```

kemudian file string.sh dijalankan menggunakan bash (nama\_file.sh), dan akan muncul output dari string

### 2. array



```
alvin@alvinryandana: ~/Tugas
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]} !"
```

pada penggunaan array akan mendeklarasikan array yang berisi 5 variabel yaitu Mint, Ubuntu, Kali, Arch, dan Debian yang kemudian akan dipilih secara random dengan \$RANDOM. kemudian let adalah kata kunci bash yang digunakan untuk pembuatan

variable dengan evaluasi aritmatika sederhana

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano array.sh
alvin@alvinryandana:~/Tugas$ bash array.sh
saya memilih distro 4, Debian !
alvin@alvinryandana:~/Tugas$
```

pada saat menjalankan bash array.sh maka akan muncul output yang berupa isi dari array yang dipilih secara random

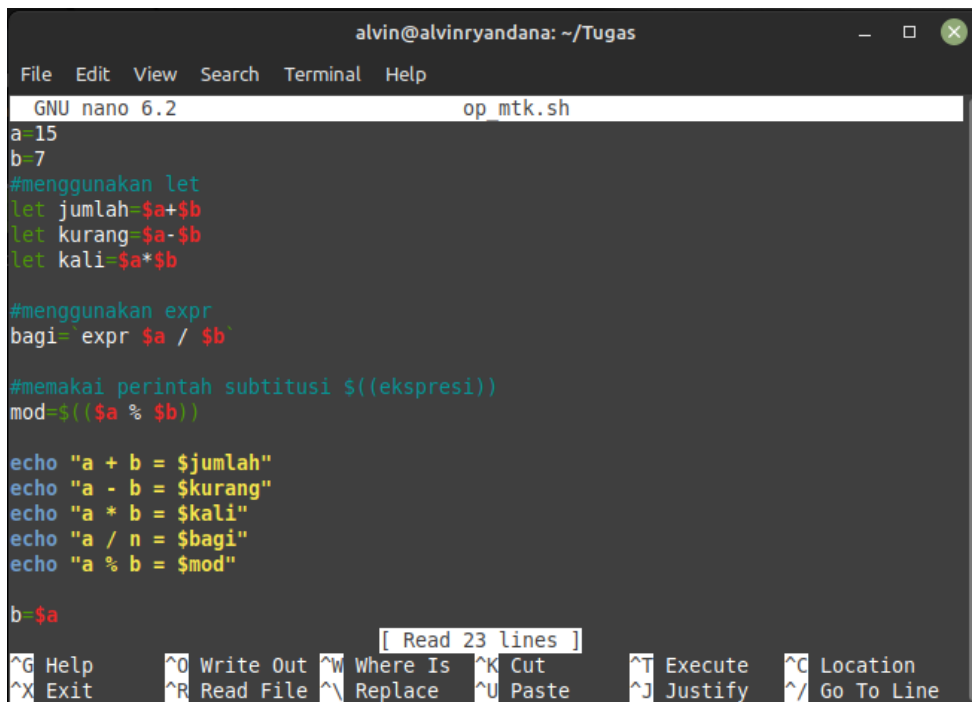
### 3. integer

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 integer.sh
a=123
b=456
let c=a+b
echo $c
```

pada penggunaan integer akan dilakukan penjumlahan antara variabel a dan b, fungsi dari echo \$c adalah menampilkan angka yang tersimpan pada variabel c. variabel c sendiri memiliki hasil dari penjumlahan antara variabel a dan b ( $c=a+b$ )

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano integer.sh
alvin@alvinryandana:~/Tugas$ bash integer.sh
579
alvin@alvinryandana:~/Tugas$
```

#### 4. operasi matematika



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 op_mtk.sh
a=15
b=7
#menggunakan let
let jumlah=$a+$b
let kurang=$a-$b
let kali=$a*$b

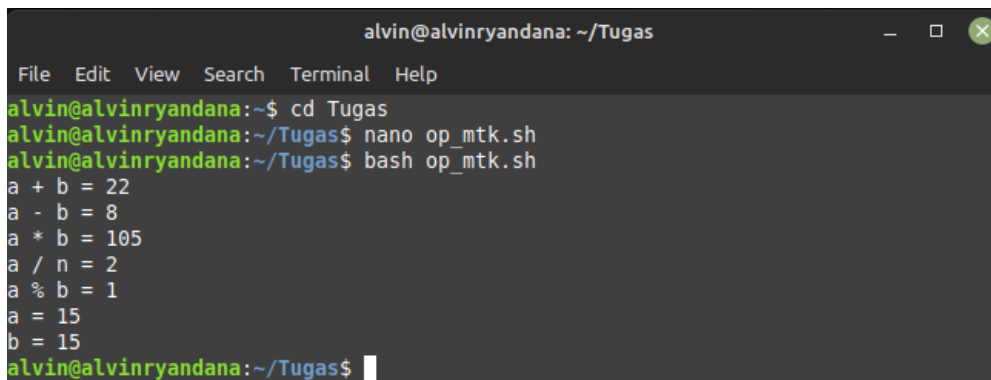
#menggunakan expr
bagi= expr $a / $b`

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

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

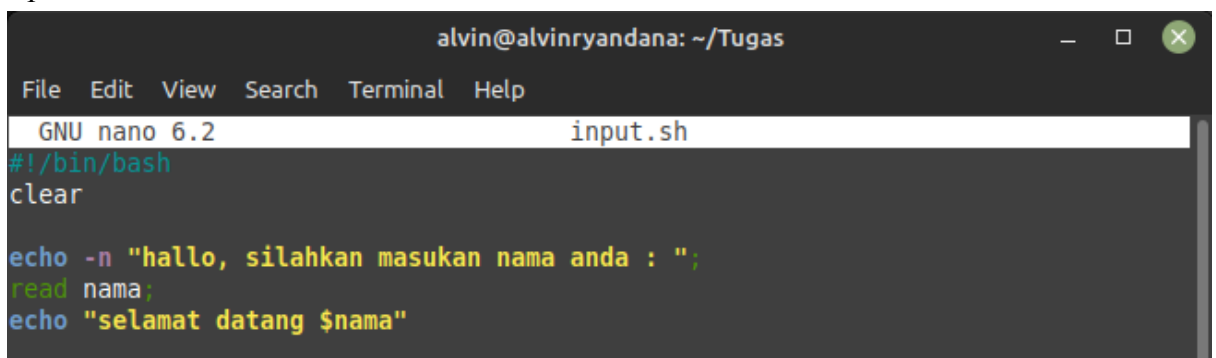
b=$a
```

pada penggunaan operasi aritmatika ada 3 cara yang bisa di lakukan yang pertama adalah menggunakan built in let, menggunakan perintah eksternal expr atau awk dan yang terakhir menggunakan perintah substitusi \$. dari sumber yang saya temukan Perintah let digunakan untuk mengevaluasi ekspresi aritmatika pada variabel shell dan perintah expr adalah variabel counter baris perintah manual yang digunakan untuk mengevaluasi ekspresi di linux bawah



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano op_mtk.sh
alvin@alvinryandana:~/Tugas$ bash op_mtk.sh
a + b = 22
a - b = 8
a * b = 105
a / n = 2
a % b = 1
a = 15
b = 15
alvin@alvinryandana:~/Tugas$
```

#### 5. input



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 input.sh
#!/bin/bash
clear

echo -n "hallo, silahkan masukan nama anda : ";
read nama;
echo "selamat datang $nama"
```

Pada penggunaan input disini akan mendeklarasikan read nama, fungsi dari read disini adalah untuk mengambil value dari inputan user, dimana inputan user tersebut akan disimpan ke dalam variabel nama lalu variabel nama tersebut di cetak pada perintah echo yang kedua

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
hallo, silahkan masukan nama anda : wd
selamat datang wd
alvin@alvinryandana:~/Tugas$ bash input.sh

alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
hallo, silahkan masukan nama anda : alvin ryan dana
selamat datang alvin ryan dana
alvin@alvinryandana:~/Tugas$
```

saat script dijalankan disini saya memberi contoh memasukan nama saya sendiri

#### 6. output 1

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 output1.sh
#untuk menampilkan teks biasa atau dari sebuah variabel
echo "selamat datang ,silahkan masukan nama anda"
read nama
echo -e "selamat datang $nama\nselamat menikmati liburan"
```

Sama seperti penggunaan input pada output juga menggunakan read untuk mengambil value dari inputan user, disini variabel nama akan menyimpan inputan dari user yang kemudian pada akhir akan dimunculkan dengan \$nama

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano output1.sh
alvin@alvinryandana:~/Tugas$ bash output1.sh
selamat datang ,silahkan masukan nama anda
alvin
selamat datang alvin
selamat menikmati liburan
alvin@alvinryandana:~/Tugas$
```

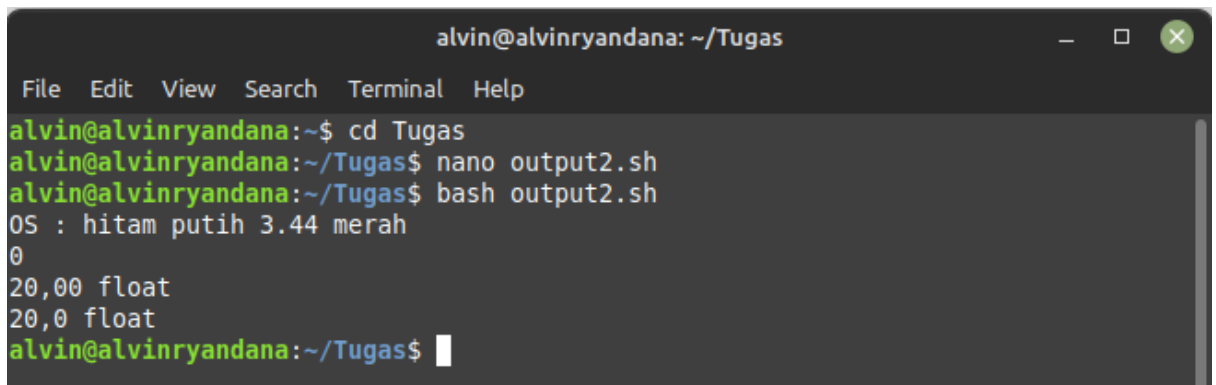
## 7. output 2



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 output2.sh
#inisialisasi Var
a=20;
b=2;
distrowarna="hitam putih 3.44 merah";
let c=a%b;

# Output
printf "OS : $distrowarna \n";
printf "$c \n";
printf "%.2f float \n" $a;
printf "%.1f float \n" $a;
```

pada contoh output kedua selain menggunakan echo, output juga dapat dimunculkan dengan menggunakan bahasa pemrograman C salah satunya adalah printf



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano output2.sh
alvin@alvinryandana:~/Tugas$ bash output2.sh
OS : hitam putih 3.44 merah
0
20,00 float
20,0 float
alvin@alvinryandana:~/Tugas$
```

## 8. percabangan 1



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 percabangan1.sh
#!/bin/bash
a=15
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
```

pada percabangan 1 disini dibuat untuk membandingkan value dari 2 variabel yaitu

variabel a dan b, dengan kondisi if then yang sudah ditetapkan di atas jika  $a == b$  akan menampilkan “a sama dengan b”, jika  $a -gt b$  akan menampilkan “a lebih besar dari b”, jika  $a -lt b$  akan menampilkan “a lebih kecil dari b”, dan ketika apabila tidak memenuhi akan menampilkan “Tidak ada kondisi yang memenuhi”

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano percabangan1.sh
alvin@alvinryandana:~/Tugas$ nano percabangan1.sh
alvin@alvinryandana:~/Tugas$ bash percabangan1.sh
a lebih besar dari b
alvin@alvinryandana:~/Tugas$
```

pada saat menjalankan script percabangan1.sh disini karena value dari a lebih besar dari b maka disini ditetapkan a -gt b yang memunculkan “a lebih besar dari b”

#### 9. percabangan 2

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 percabangan2.sh
#!/bin/bash

printf "jajan apa yang kamu suka? \n"
printf "lumpiah ?\n"
printf "siomay ?\n"
printf "pentol ?\n"

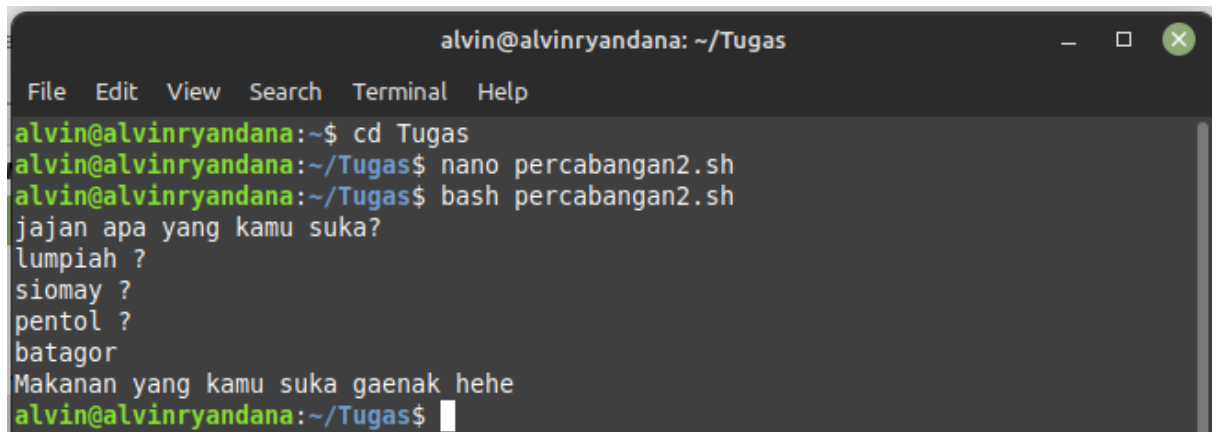
read jajan

case "$jajan" in
    "lumpiah")
        echo "Lumpiah buk mah wenak slur!"
        ;;
    "siomay")
        echo "Siomay mas budi mantap bat"
        ;;
    "pentol")
        echo "Pentol kantinrasane unch-unch"
        ;;
    *)
        ;;
esac
```

[ Read 23 lines ]

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

pada percabangan 2 ini menggunakan case untuk menyederhanakan pemakaian if yang berantai atau kompleks, disini juga menggunakan inputan dari user yang akan mempengaruhi hasil dari output percabangan

A terminal window titled 'alvin@alvinryandana: ~/Tugas' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano percabangan2.sh
alvin@alvinryandana:~/Tugas$ bash percabangan2.sh
jajan apa yang kamu suka?
lumpiah ?
siomay ?
pentol ?
batagor
Makanan yang kamu suka gaenak hehe
alvin@alvinryandana:~/Tugas$
```

pada contoh menjalankan script percabangan2.sh disini saya inputkan batagor yang mengeluarkan output “Makanan yang kamu suka gaenak hehe”

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

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_2.sh
a=18
b=33
printf "a = 18 \n"
printf "b = 33 \n"

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

printf "silahkan memilih operasi aritmatika \n"
printf "jumlah \n"
printf "kurang \n"
printf "kali \n"

read pilih

case "$pilih" in
    "jumlah")
        echo "hasil dari penjumlahan adalah $jumlah"
        ;;
    "kurang")
        echo "hasil dari pengurangan adalah $kurang"
        ;;
    "kali")
        echo "hasil dari perkalian adalah $kali"
        ;;
    *)
        echo "mohon untuk masukan pilihan yang tersedia"
        ;;
esac

^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
```

```
*)
    echo "mohon untuk masukan pilihan yang tersedia"
    ;;
esac

^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
```

pada script di atas saya menggunakan operasi matematika penjumlahan, pengurangan, dan perkalian dengan menggunakan built in let, untuk percabangan di atas saya menggunakan contoh seperti percabangan2 dengan menggunakan case, di script ini user akan menentukan operasi matematika antara dua variabel a dan b yang dimana value dari a adalah 18 dan b adalah 33. user dapat memilih ketiga operasi tersebut dengan memasukkan inputan yang sesuai

```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~$ cd Tugas
alvin@alvinryandana:~/Tugas$ nano Tugas_2.sh
alvin@alvinryandana:~/Tugas$ bash Tugas_2.sh
a = 18
b = 33
silahkan memilih operasi aritmatika
jumlah
kurang
kali
jumlah
hasil dari penjumlahan adalah 51
alvin@alvinryandana:~/Tugas$
```



pada saat menjalankan script Tugas\_2 saya memilih jumlah seperti di atas dan akan mengeluarkan output berupa penjumlahan antara kedua variabel