

LAPORAN PRAKTIKUM SISTEM OPERASI

Praktikum 7 Pemrograman Bash Shell 1



Disusun oleh :

Nama : Shofira Izza Nurrohmah

Kelas : 1 D3 Teknik Informatika A

NRP : 3122500026

Dosen Pengampu : Fitri Setyorini ST., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PRODI TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2023

DAFTAR ISI

HALAMAN SAMPUL	I
DAFTAR ISI	II
A. TUGAS PENDAHULUAN	1
B. PERCOBAAN	2
C. KESIMPULAN	28

TUGAS PENDAHULUAN:

Sebagai tugas pendahuluan, ketiklah program yang ada di latihan praktikum, yaitu prog01.sh, prog02.sh, prog03.sh, prog04.sh dan prog 05.sh, dalam bentuk text, sehingga pada waktu praktikum tinggal copy paste dari tugas ini.

➤ **prog01.sh**

```
$ nano prog01.sh
#!/bin/sh
V1=poltek
V2=':'
V3=elektronika
echo "Pemrograman shell"
echo $V1$V2$V3
V3=ITS
echo $V1$V2 di $V3
```

➤ **prog02.sh**

```
$ nano prog02.sh
#!/bin/sh
# prog02.sh
# membaca nama dan alamat

echo "Nama Anda : "
read nama
echo "Alamat : "
read alamat
echo "Kota : "
read kota

echo
echo "Hasil adalah : $nama, $alamat di $kota"
```

➤ **prog03.sh**

```
$ nano prog03.sh
#!/bin/sh
# prog03.sh
#
NAMA=`whoami`

echo Nama Pengguna Aktif adalah $NAMA

tanggal=`date | cut -c1-10`

echo Hari ini tanggal $tanggal
```

➤ **prog04.sh**

```
$ nano prog04.sh
#!/bin/sh
# prog04.sh versi 3
```

```
# Parameter passing
#
echo "Jumlah parameter yang diberikan adalah $#"
```

```
echo "Nama program adalah $0"
```

```
echo "Parameter 1 adalah $1"
```

```
echo "Parameter 2 adalah $2"
```

```
echo "Parameter 3 adalah $3"
```

```
echo "Total parameter adalah $*"
```

```
echo "PID proses shell ini adalah $$"
```

➤ prog05.sh

```
$ nano prog05.sh
#!/bin/sh
# prog05.sh
# Program akan memberikan konfirmasi apakah nama
# user sedang aktif atau tidak
#
echo -n "Berikan nama pemakai : "
```

```
read nama
```

```
if who | grep $nama > /dev/null
```

```
then echo "$nama sedang aktif"
```

```
else
```

```
echo "$nama tidak aktif"
```

```
fi
```

PERCOBAAN:

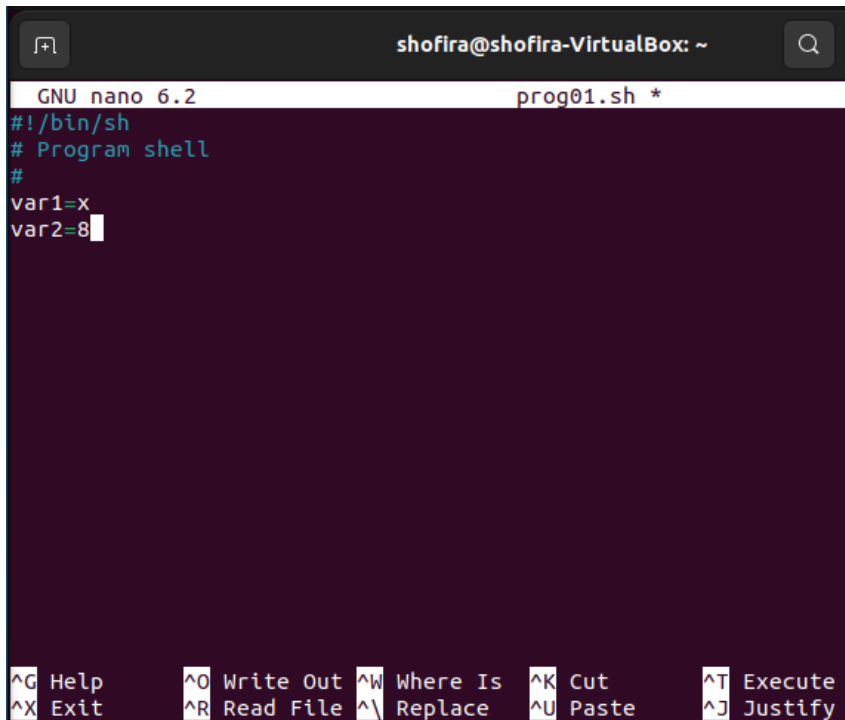
1. Login sebagai user.
2. Bukalah Console Terminal dan lakukan percobaan-percobaan di bawah ini kemudian analisa hasil percobaan.
3. Selesaikan soal-soal latihan.

Percobaan 1 : Membuat shell script

1. Buatlah file prog01.sh dengan editor vi

```
$ vi prog01.sh
#!/bin/sh
# Program shell
#
var1=x
var2=8
```

```
shofira@shofira-VirtualBox:~$ sudo nano
[sudo] password for shofira:
shofira@shofira-VirtualBox:~$ nano prog01.sh
shofira@shofira-VirtualBox:~$
```



```
shofira@shofira-VirtualBox: ~
GNU nano 6.2 prog01.sh *
#!/bin/sh
# Program shell
#
var1=x
var2=8
^G Help  ^O Write Out  ^W Where Is  ^K Cut       ^T Execute
^X Exit  ^R Read File  ^\ Replace   ^U Paste     ^J Justify
```

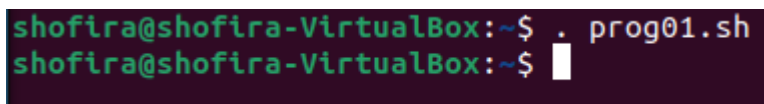
Analisa :

Perintah `nano prog01.sh` digunakan untuk membuat shell script dengan nama file `prog01.sh`.

- `#!/bin/sh` — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya `/bin/csh`, `/bin/ksh` dan lainnya.
- `# Program shell`— Program shell ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `#` — tanda ini hanya sebagai komentar.
- `var1=x` — menambahkan variabel yang mempunyai nilai x.
- `var2=8`— menambahkan variabel yang mempunyai nilai 8, perlu diperhatikan bahwa variabel ini tidak boleh ada spasi karena hal ini shell akan menganggap sebuah parameter serta variabel bersifat case-sensitive.

2. Untuk menjalankan shell, gunakan notasi TITIK di depan nama program

```
$ . prog01.sh
```



```
shofira@shofira-VirtualBox:~$ . prog01.sh
shofira@shofira-VirtualBox:~$
```

Analisa :

Perintah `. prog01.sh` digunakan untuk menjalankan shell bernama `prog01.sh`, hasil dari program tersebut tidak ada karena pada script tidak tertulis kode yang dapat menampilkan output.

3. Untuk menjalankan shell, dapat juga dengan membuat executable file dan dieksekusi relatif dari current directory

```
$ chmod +x prog01.sh  
$ ./prog01.sh
```

```
shofira@shofira-VirtualBox:~$ chmod +x prog01.sh  
shofira@shofira-VirtualBox:~$ ./prog01.sh  
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `chmod +x prog01.sh` digunakan untuk menjalankan shell dengan membuat executable file dan dieksekusi relatif dari current directory, hasil dari program tersebut tidak ada karena pada script tidak tertulis kode yang dapat menampilkan output.
- Perintah `./prog01.sh` digunakan untuk menjalankan shell bernama `prog01.sh`, hasil dari program tersebut tidak ada karena pada script tidak tertulis kode yang dapat menampilkan output.

Percobaan 2 : Variabel

1. Contoh menggunakan variable pada shell interaktif

```
$ VPT=poltek  
$ echo $VPT
```

```
shofira@shofira-VirtualBox:~$ VPT=poltek  
shofira@shofira-VirtualBox:~$ echo $VPT  
poltek  
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `VPT=poltek` digunakan untuk menginputkan variabel bernama `VPT` dengan nilai string `poltek`.
- Perintah `echo $VPT` digunakan untuk menampilkan output berupa nilai dari variabel yang didefinisikan sebelumnya, yaitu `poltek`.

2. Pemisahan 2 kata dengan spasi menandakan eksekusi 2 buah instruksi. Karakter \$ harus ada pada awal nama variable untuk melihat isi variable tersebut, jika tidak, maka echo akan mengambil parameter tersebut sebagai string.

```
$ VPT2=poltek elektronika (Terdapat pesan error)
$ VPT2="poltek elektronika"
$ echo VPT2
$ echo $VPT2
```

```
shofira@shofira-VirtualBox:~$ VPT2=poltek elektronika
elektronika: command not found
shofira@shofira-VirtualBox:~$ VPT2="poltek elektronika"
shofira@shofira-VirtualBox:~$ echo VPT2
VPT2
shofira@shofira-VirtualBox:~$ echo $VPT2
poltek elektronika
```

Analisa :

- Perintah `VPT2=poltek elektronika` (Terdapat pesan error) digunakan untuk menginputkan variabel bernama VPT2 dengan nilai string poltek elektronika, tetapi setelah mengeksekusi perintah tersebut muncul pesan error karena ketika menulis nilai variabel yang ada beberapa kata diberikan jarak antar kata dengan spasi dan tanda petik dua atas (") yang ada di awal dan diakhir nilai variabel.
- Perintah `VPT2="poltek elektronika"` digunakan untuk menginputkan variabel bernama VPT2 dengan nilai string poltek elektronika.
- Perintah `echo VPT2` digunakan untuk menampilkan output VPT2 karena tidak memanggil variabel tetapi mencetak langsung string VPT2, untuk memanggil variabel dan dicetak dengan benar di depan variabel harus terdapat tanda \$ dilanjutkan nama variabel-nya.
- Perintah `echo $VPT2` digunakan untuk menampilkan output berupa nilai dari variabel yang didefinisikan sebelumnya, yaitu poltek elektronika.

3. Menggabungkan dua variable atau lebih

```
$ V1=poltek
$ V2=':'
$ V3=elektronika
$ V4=$V1$V2$V3
$ echo $V4
```

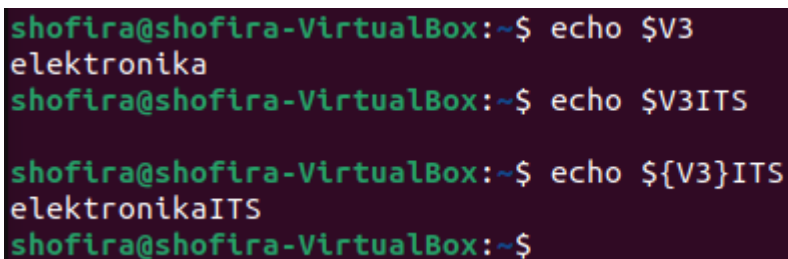
```
shofira@shofira-VirtualBox:~$ V1=poltek
shofira@shofira-VirtualBox:~$ V2=':'
shofira@shofira-VirtualBox:~$ V3=elektronika
shofira@shofira-VirtualBox:~$ V4=$V1$V2$V3
shofira@shofira-VirtualBox:~$ echo $V4
poltek:elektronika
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `V1=poltek` digunakan untuk menginputkan variabel bernama V1 dengan nilai string poltek.
- Perintah `V2=' : '` digunakan untuk menginputkan variabel bernama V2 dengan nilai string dengan tanda titik dua (:).
- Perintah `V3=elektronika` digunakan untuk menginputkan variabel bernama V3 dengan nilai string elektronika.
- Perintah `V4=$V1$V2$V3` digunakan untuk menginputkan ketiga variabel yang telah dibuat ke dalam variabel bernama V4.
- Perintah `echo $V4` digunakan untuk menampilkan output berupa nilai dari variabel \$V4 yang meliputi nilai dari variabel \$V1\$V2\$V3, yaitu poltek:elektronika.

4. Menggabungkan isi variable dengan string yang lain. Jika digabungkan dengan nama variable yang belum didefinisikan (kosong) maka instruksi echo menghasilkan string kosong. Untuk menghindari kekeliruan, nama variable perlu diproteksi dengan { } dan kemudian isi variable tersebut digabung dengan string.

```
$ echo $V3
$ echo $V3ITS
$ echo ${V3}ITS
```



```
shofira@shofira-VirtualBox:~$ echo $V3
elektronika
shofira@shofira-VirtualBox:~$ echo $V3ITS

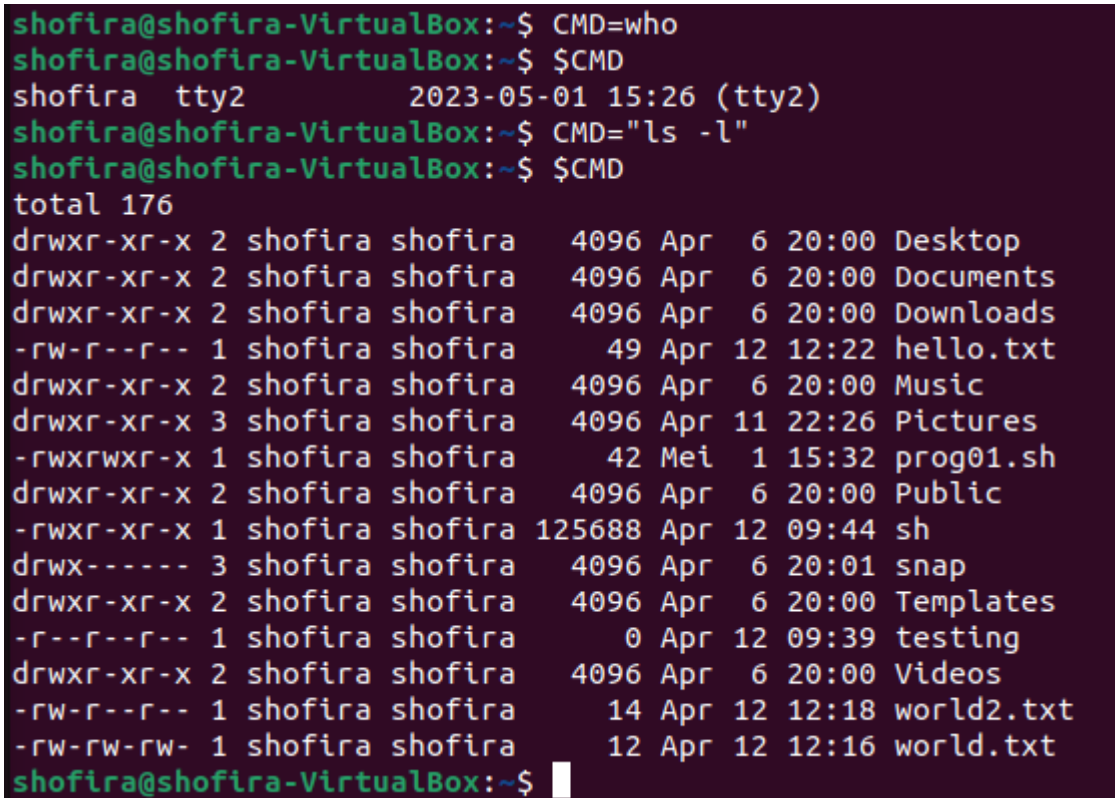
shofira@shofira-VirtualBox:~$ echo ${V3}ITS
elektronikaITS
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `echo $V3` digunakan untuk menampilkan output berupa nilai dari variabel yang didefinisikan sebelumnya, yaitu elektronika.
- Perintah `echo $V3ITS` digunakan untuk menampilkan output gabungan isi variabel \$V3 dengan string lain tanpa spasi yang bertuliskan ITS dan menghasilkan output kosong karena salah dalam pendefinisian variabel dimana seharusnya nama variabel perlu diproteksi dengan tanda kurawal { } dan kemudian isi variabel tersebut digabung dengan string.
- Perintah `echo ${V3}ITS` digunakan untuk menampilkan output gabungan isi variabel dengan string lain tanpa spasi yang bertuliskan ITS dan menghasilkan elektronikaITS.

5. Variabel dapat berisi instruksi, yang kemudian bila dijadikan input untuk shell, instruksi tersebut akan dieksekusi

```
$ CMD=who
$ $CMD
$ CMD="ls -l"
$ $CMD
```



```
shofira@shofira-VirtualBox:~$ CMD=who
shofira@shofira-VirtualBox:~$ $CMD
shofira tty2          2023-05-01 15:26 (tty2)
shofira@shofira-VirtualBox:~$ CMD="ls -l"
shofira@shofira-VirtualBox:~$ $CMD
total 176
drwxr-xr-x 2 shofira shofira 4096 Apr  6 20:00 Desktop
drwxr-xr-x 2 shofira shofira 4096 Apr  6 20:00 Documents
drwxr-xr-x 2 shofira shofira 4096 Apr  6 20:00 Downloads
-rw-r--r-- 1 shofira shofira  49 Apr 12 12:22 hello.txt
drwxr-xr-x 2 shofira shofira 4096 Apr  6 20:00 Music
drwxr-xr-x 3 shofira shofira 4096 Apr 11 22:26 Pictures
-rwxrwxr-x 1 shofira shofira  42 Mei  1 15:32 prog01.sh
drwxr-xr-x 2 shofira shofira 4096 Apr  6 20:00 Public
-rwxr-xr-x 1 shofira shofira 125688 Apr 12 09:44 sh
drwx----- 3 shofira shofira 4096 Apr  6 20:01 snap
drwxr-xr-x 2 shofira shofira 4096 Apr  6 20:00 Templates
-r--r--r-- 1 shofira shofira   0 Apr 12 09:39 testing
drwxr-xr-x 2 shofira shofira 4096 Apr  6 20:00 Videos
-rw-r--r-- 1 shofira shofira  14 Apr 12 12:18 world2.txt
-rw-rw-rw- 1 shofira shofira  12 Apr 12 12:16 world.txt
shofira@shofira-VirtualBox:~$
```

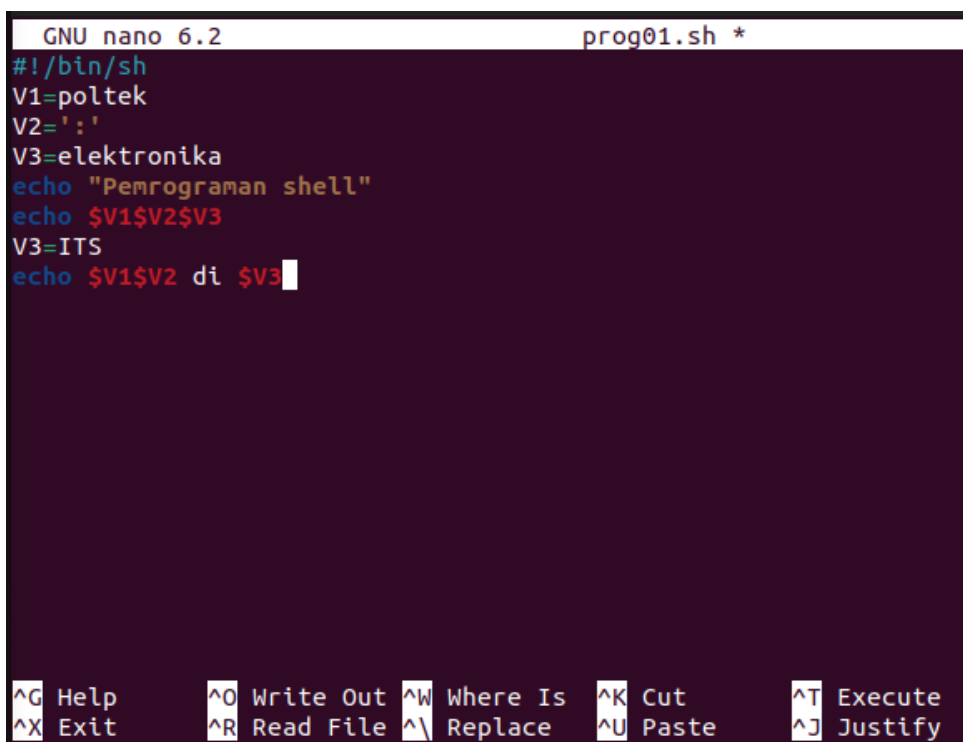
Analisa :

- Perintah `CMD=who` digunakan untuk menetapkan string `who` pada variabel `CMD`.
- Perintah `CMD` digunakan untuk menjalankan perintah yang ada dalam variabel `CMD`, yaitu `who` dan menampilkan daftar pengguna yang sedang login ke sistem.
- Perintah `CMD="ls -l"` digunakan untuk menetapkan string `ls -l` pada variabel `CMD`.
- Perintah `CMD` digunakan untuk menjalankan perintah yang ada dalam variabel `CMD`, yaitu `ls -l` dan menampilkan isi daftar file dan direktori dalam format panjang.

6. Modifikasi file prog01.sh berikut

```
$ vi prog01.sh
#!/bin/sh
V1=poltek
V2=':'
V3=elektronika
echo "Pemrograman shell"
echo $V1$V2$V3
V3=ITS
echo $V1$V2 di $V3
```

```
shofira@shofira-VirtualBox:~$ nano prog01.sh
shofira@shofira-VirtualBox:~$
```



```
GNU nano 6.2 prog01.sh *
#!/bin/sh
V1=poltek
V2=':'
V3=elektronika
echo "Pemrograman shell"
echo $V1$V2$V3
V3=ITS
echo $V1$V2 di $V3
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

Analisa :

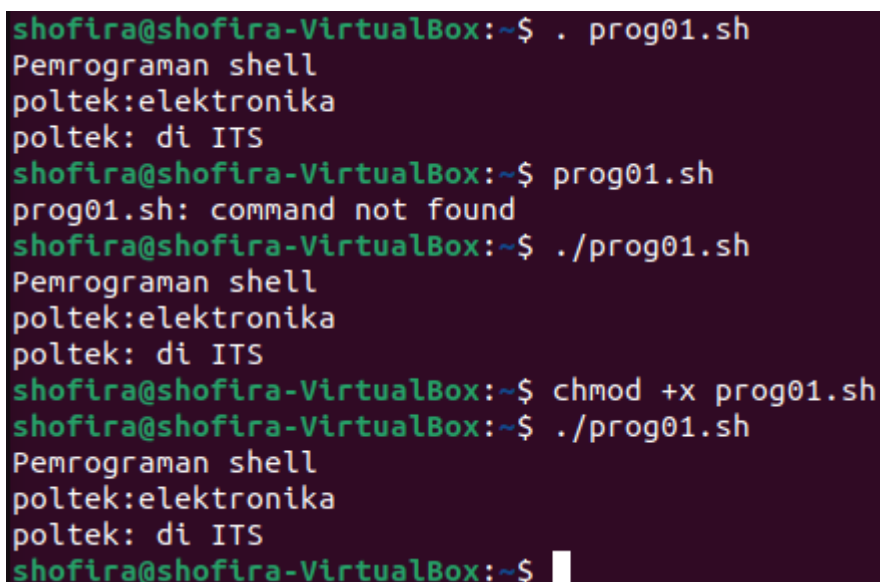
Perintah `nano prog01.sh` digunakan untuk memodifikasi script dari file `prog01.sh` yang sebelumnya telah dibuat dengan menggunakan editor nano.

- `#!/bin/sh` — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya `/bin/csh`, `/bin/ksh` dan lainnya.
- `V1=poltek` digunakan untuk menginputkan variabel bernama V1 dengan nilai string poltek.
- `V2=':'` digunakan untuk menginputkan variabel bernama V2 dengan nilai string dengan tanda titik dua (:).
- `V3=elektronika` digunakan untuk menginputkan variabel bernama V3 dengan nilai string elektronika.
- `echo "Pemrograman shell"` digunakan untuk menampilkan output Pemrograman shell.

- `echo $V1$V2$V3` digunakan untuk menampilkan output berupa nilai dari variabel \$V4 yang meliputi nilai dari variabel \$V1\$V2\$V3, yaitu poltek:elektronika.
- `V3=ITS` digunakan untuk mengubah variabel \$V3 yang awalnya elektronika dengan nilai string ITS.
- `echo $V1$V2` di `$V3` digunakan untuk menampilkan output berupa nilai dari variabel \$V4 yang meliputi nilai dari variabel \$V1\$V2 \$V3, yaitu poltek: di ITS.

7. Cara sederhana mengeksekusi shell adalah dengan menggunakan notasi titik di depan nama shell script tersebut. Bila direktori actual tidak terdaftar dalam PATH, maka command tersebut tidak dapat ditemukan. Bila script belum executable, script tidak dapat dieksekusi.

```
$ . prog01.sh
$ prog01.sh          (Terdapat pesan error)
$ ./prog01.sh        (Terdapat pesan error)
$ chmod +x prog01.sh
$ ./prog01.sh
```



```
shofira@shofira-VirtualBox:~$ . prog01.sh
Pemrograman shell
poltek:elektronika
poltek: di ITS
shofira@shofira-VirtualBox:~$ prog01.sh
prog01.sh: command not found
shofira@shofira-VirtualBox:~$ ./prog01.sh
Pemrograman shell
poltek:elektronika
poltek: di ITS
shofira@shofira-VirtualBox:~$ chmod +x prog01.sh
shofira@shofira-VirtualBox:~$ ./prog01.sh
Pemrograman shell
poltek:elektronika
poltek: di ITS
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `. prog01.sh` digunakan untuk menjalankan shell bernama prog01.sh dan terdapat hasil dari program tersebut karena pada script tertulis kode yang dapat menampilkan output (echo).
- Perintah `prog01.sh` (Terdapat pesan error) digunakan untuk menjalankan shell bernama prog01.sh namun terdapat pesan error karena menggunakan notasi titik di depan nama shell script tersebut sehingga tidak dapat mengeksekusi shell.
- Perintah `./prog01.sh` (Terdapat pesan error) digunakan untuk menjalankan shell bernama prog01.sh namun terdapat pesan error karena file harus dijadikan executable terlebih dahulu.
- Perintah `chmod +x prog01.sh` digunakan untuk menjalankan shell dengan membuat executable file dan dieksekusi relatif dari current directory..

- Perintah `./prog01.sh` digunakan untuk menjalankan shell bernama `prog01.sh` dan bisa menampilkan output dari program tersebut setelah file `prog01.sh` dijadikan file executable.

Percobaan 3 : Membaca keyboard

1. Menggunakan instruksi `read`

```
$ read nama
amir
$ echo $nama
```

```
shofira@shofira-VirtualBox:~$ read nama
Shofi
shofira@shofira-VirtualBox:~$ echo $nama
Shofi
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `read nama` digunakan untuk mengisi variabel melalui masukan keyboard (yaitu mengetikkan Shofi sebagai isian dari variabel `nama`).
- Perintah `echo $nama` digunakan untuk menampilkan output dari variabel `$nama`.

2. Membaca nama dan alamat dari keyboard

```
$ vi prog02.sh
#!/bin/sh
# prog02.sh
# membaca nama dan alamat

echo "Nama Anda : "
read nama
echo "Alamat : "
read alamat
echo "Kota : "
read kota

echo
echo "Hasil adalah : $nama, $alamat di $kota"
```

```
shofira@shofira-VirtualBox:~$ nano prog02.sh
shofira@shofira-VirtualBox:~$
```

Analisa :

Perintah `nano prog02.sh` digunakan untuk membuat shell script dengan nama file `prog02.sh`.

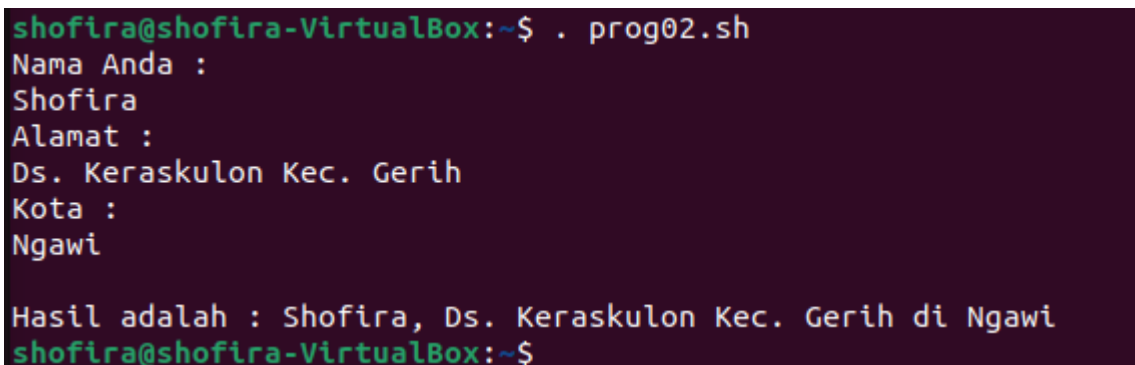
- `#!/bin/sh` — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya `/bin/csh`, `/bin/ksh` dan lainnya.
- `# prog02.sh` — sebagai nama file shell script yang akan diisi.

- # membaca nama dan alamat — Program shell ini sebatas komentar mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- echo "Nama Anda : " digunakan untuk menampilkan output Nama Anda : .
- read nama digunakan untuk mengisi variabel melalui masukan keyboard.
- echo "Alamat : " digunakan untuk menampilkan output Alamat : .
- read alamat digunakan untuk mengisi variabel melalui masukan keyboard.
- echo "Kota : " digunakan untuk menampilkan output Kota : .
- read kota digunakan untuk mengisi variabel melalui masukan keyboard.
- echo digunakan untuk memberikan baris baru.
- echo "Hasil adalah : \$nama, \$alamat di \$kota" digunakan untuk menampilkan output dari variabel \$nama, \$alamat, dan \$kota.

3. Eksekusi program prog02.sh

```
$ . prog02.sh
Nama Anda :
Amir
Alamat :
Jl semangka 67
Kota :
Surabaya

Hasil adalah : Amir, Jl semangka di Surabaya
```



```
shofira@shofira-VirtualBox:~$ . prog02.sh
Nama Anda :
Shofira
Alamat :
Ds. Keraskulon Kec. Gerih
Kota :
Ngawi

Hasil adalah : Shofira, Ds. Keraskulon Kec. Gerih di Ngawi
shofira@shofira-VirtualBox:~$
```

Analisa :

Perintah `. prog02.sh` digunakan untuk menjalankan shell bernama prog02.sh dan terdapat hasil dari program tersebut karena pada script tertulis kode yang dapat menampilkan output (echo).

4. Instruksi `echo` secara otomatis memberikan baris baru, maka untuk menghindari hal tersebut disediakan opsi `-n`, yang menyatakan kepada `echo` untuk menghilangkan baris baru. Modifikasi program `prog02.sh`

```
$ vi prog02.sh
#!/bin/sh
# prog02.sh
# membaca nama dan alamat

echo -n "Nama Anda : "
read nama
echo -n "Alamat : "
read alamat
echo -n "Kota : "
read kota

echo
echo "Hasil adalah : $nama, $alamat di $kota"
```

```
shofira@shofira-VirtualBox:~$ nano prog02.sh
shofira@shofira-VirtualBox:~$
```



```
GNU nano 6.2 prog02.sh *
#!/bin/sh
# prog02.sh
# membaca nama dan alamat

echo -n "Nama Anda : "
read nama
echo -n "Alamat : "
read alamat
echo -n "Kota : "
read kota

echo
echo "Hasil adalah : $nama, $alamat di $kota"
```

Analisa :

Perintah `nano prog02.sh` digunakan untuk membuat shell script dengan nama file `prog02.sh`.

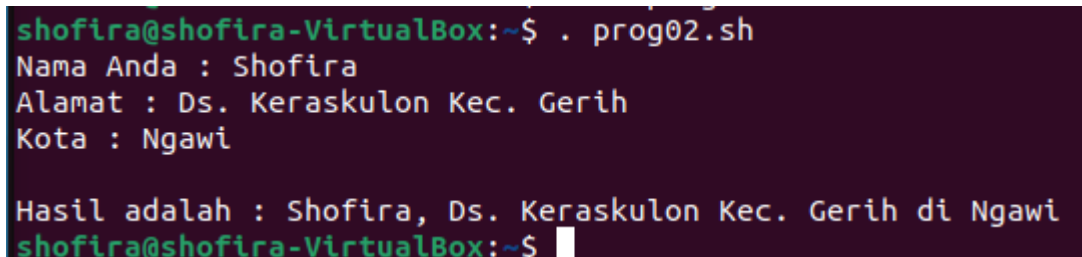
- `#!/bin/sh` — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya `/bin/csh`, `/bin/ksh` dan lainnya.
- `# prog02.sh` — sebagai nama file shell script yang akan diisi.
- `# membaca nama dan alamat` — Program shell ini sebatas komentar mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.

- `echo -n "Nama Anda : "` digunakan untuk mengatasi hal ini agar tidak memberikan baris baru karena secara default menulis echo pada script tanpa ada opsi tambahan, echo dijalankan akan memberikan baris baru.
- `read nama` digunakan untuk mengisi variabel melalui masukan keyboard.
- `echo -n "Alamat : "` digunakan untuk mengatasi hal ini agar tidak memberikan baris baru karena secara default menulis echo pada script tanpa ada opsi tambahan, echo dijalankan akan memberikan baris baru.
- `read alamat` digunakan untuk mengisi variabel melalui masukan keyboard.
- `echo -n "Kota : "` digunakan untuk mengatasi hal ini agar tidak memberikan baris baru karena secara default menulis echo pada script tanpa ada opsi tambahan, echo dijalankan akan memberikan baris baru.
- `read kota` digunakan untuk mengisi variabel melalui masukan keyboard.
- `echo` digunakan untuk memberikan baris baru.
- `echo "Hasil adalah : $nama, $alamat di $kota"` digunakan untuk menampilkan output dari variabel \$nama, \$alamat, dan \$kota.

5. Eksekusi program prog02.sh

```
$ . prog02.sh
Nama Anda : Amir
Alamat : Jl semangka 67
Kota : Surabaya

Hasil adalah : Amir, Jl semangka di Surabaya
```



```
shofira@shofira-VirtualBox:~$ . prog02.sh
Nama Anda : Shofira
Alamat : Ds. Keraskulon Kec. Gerih
Kota : Ngawi

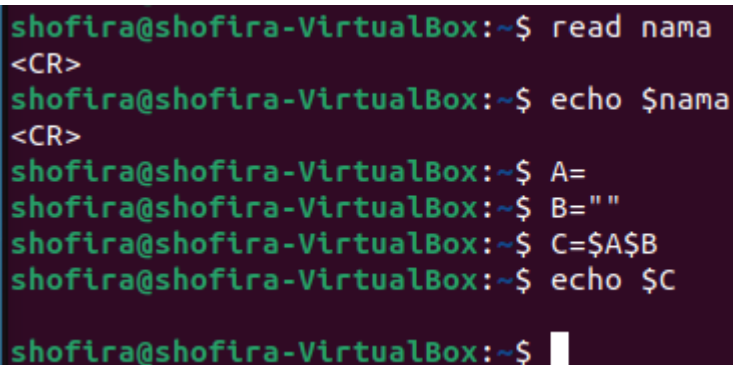
Hasil adalah : Shofira, Ds. Keraskulon Kec. Gerih di Ngawi
shofira@shofira-VirtualBox:~$
```

Analisa :

Perintah `. prog02.sh` digunakan untuk menjalankan shell bernama prog02.sh dan terdapat hasil dari program tersebut karena pada script tertulis kode yang dapat menampilkan output (echo).

6. Variabel kosong adalah variable yang tidak mempunyai nilai. Variabel ini didapat atas assignment atau membaca dari keyboard atau variable yang belum didefinisikan

```
$ read nama
<CR>
$ echo $nama
$ A=
$ B=""
$ C=$A$B
$ echo $C
```



```
shofira@shofira-VirtualBox:~$ read nama
<CR>
shofira@shofira-VirtualBox:~$ echo $nama
<CR>
shofira@shofira-VirtualBox:~$ A=
shofira@shofira-VirtualBox:~$ B=""
shofira@shofira-VirtualBox:~$ C=$A$B
shofira@shofira-VirtualBox:~$ echo $C
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `read nama` digunakan untuk mengisi variabel melalui masukan keyboard.
- Perintah `echo $nama` digunakan untuk menampilkan output dari variabel \$nama.
- Perintah `A=` digunakan untuk menginputkan variabel kosong bernama A.
- Perintah `B=""` digunakan untuk menginputkan variabel kosong bernama B.
- Perintah `C=AB` digunakan untuk menginputkan kedua variabel yang telah dibuat ke dalam variabel bernama C.
- Perintah `echo $C` digunakan untuk menampilkan output dari variabel \$C.

7. Variabel dapat disubstitusikan dengan hasil eksekusi dari sebuah instruksi. Pada contoh dibawah , instruksi `pwd` dieksekusi lebih dahulu dengan sepasang Back Quate (tanda kutip terbalik). Hasil dari eksekusi tersebut akan masuk sebagai nilai variable `DIR`

```
$ pwd
$ DIR=`pwd`
$ echo $DIR
```



```
shofira@shofira-VirtualBox:~$ pwd
/home/shofira
shofira@shofira-VirtualBox:~$ DIR=`pwd`
shofira@shofira-VirtualBox:~$ echo $DIR
/home/shofira
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `pwd` digunakan untuk mengetahui direktori mana yang saat ini sedang dibuka.
- Perintah `DIR=`pwd`` digunakan untuk mengeksekusi perintah `pwd` dengan sepasang Back Quate (tanda kutip terbalik). Hasil dari eksekusi tersebut akan masuk sebagai nilai variabel `DIR`.
- Perintah `echo $DIR` digunakan untuk menampilkan output dari variabel `$DIR`.

8. Buatlah shell script `prog03.sh`

```
$ vi prog03.sh
#!/bin/sh
# prog03.sh
#
NAMA=`whoami`

echo Nama Pengguna Aktif adalah $NAMA

tanggal=`date | cut -c1-10`

echo Hari ini tanggal $tanggal
```

```
shofira@shofira-VirtualBox:~$ nano prog03.sh
shofira@shofira-VirtualBox:~$
```

```
GNU nano 6.2                                prog03.sh *
#!/bin/sh
# prog03.sh
#
NAMA=`whoami`

echo Nama Pengguna Aktif adalah $NAMA

tanggal=`date | cut -c1-10`

echo Hari ini tanggal $tanggal

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Analisa :

Perintah `nano prog03.sh` digunakan untuk membuat shell script dengan nama file `prog03.sh`.

- `#!/bin/sh` — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya `/bin/csh`, `/bin/ksh` dan lainnya.
- `# Program shell`— Program shell ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `#` — tanda ini hanya sebagai komentar.
- `NAMA=`whoami`` digunakan untuk mengeksekusi perintah `whoami` dengan sepasang Back Quate (tanda kutip terbalik). Hasil dari eksekusi tersebut akan masuk sebagai nilai variable `NAMA`.
- `echo Nama Pengguna Aktif adalah $NAMA` digunakan untuk menampilkan output pada variabel `$nama` yang berisi perintah `whoami` untuk mengetahui nama pengguna.
- `tanggal=`date | cut -c1-10`` digunakan untuk mengeksekusi perintah `date` dan `cut` dengan sepasang Back Quate (tanda kutip terbalik). Hasil dari eksekusi tersebut akan masuk sebagai nilai variable `tanggal`.
- `echo Hari ini tanggal $tanggal` digunakan untuk menampilkan output pada variabel `$nama` yang berisi perintah `whoami` untuk mengetahui nama pengguna dan perintah `date` mengetahui hari dan tanggal saat ini serta perintah `cut` untuk memotong bagian dari setiap baris file dan menulis hasilnya ke output standar.

9. Eksekusi `prog03.sh`

```
$ . prog03.sh
```

```
shofira@shofira-VirtualBox:~$ . prog03.sh
Nama Pengguna Aktif adalah shofira
Hari ini tanggal Sel 02 Mei
shofira@shofira-VirtualBox:~$
```

Analisa :

Perintah `. prog03.sh` digunakan untuk menjalankan shell bernama `prog03.sh` dan terdapat hasil dari program tersebut karena pada script tertulis kode yang dapat menampilkan output (echo).

Percobaan 4 : Parameter

1. Membuat shell script `prog04.sh`

```
$ vi prog04.sh
#!/bin/sh
# prog04.sh versi 1
# Parameter passing
#
echo "Nama program adalah $0"
echo "Parameter 1 adalah $1"
echo "Parameter 2 adalah $2"
echo "Parameter 3 adalah $3"
```

```
shofira@shofira-VirtualBox:~$ nano prog04.sh
shofira@shofira-VirtualBox:~$
```

```
GNU nano 6.2                                prog04.sh *
#!/bin/sh
# prog04.sh versi 1
# Parameter passing
#
echo "Nama program adalah $0"
echo "Parameter 1 adalah $1"
echo "Parameter 2 adalah $2"
echo "Parameter 3 adalah $3"

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

Analisa :

Perintah `nano prog04.sh` digunakan untuk membuat shell script dengan nama file `prog04.sh`.

- `#!/bin/sh` — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya `/bin/csh`, `/bin/ksh` dan lainnya.
- `# prog04.sh versi 1` — `# prog04.sh versi 1` ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `# Parameter passing` — `# Parameter passing` ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `#` — tanda ini hanya sebagai komentar.
- `NAMA=`whoami`` digunakan untuk mengeksekusi perintah `whoami` dengan sepasang Back Quate (tanda kutip terbalik). Hasil dari eksekusi tersebut akan masuk sebagai nilai variable `NAMA`.
- `echo "Nama program adalah $0"` digunakan untuk menampilkan output pada variabel `$0`.
- `echo "Parameter 1 adalah $1"` digunakan untuk menampilkan output pada variabel `$1`.
- `echo "Parameter 2 adalah $2"` digunakan untuk menampilkan output pada variabel `$2`.
- `echo "Parameter 3 adalah $3"` digunakan untuk menampilkan output pada variabel `$3`.

2. Eksekusi `prog04.sh` tanpa parameter, dengan 2 parameter, dengan 4 parameter

```
$ . prog04.sh
$ . prog04.sh amir hasan
$ . prog04.sh amir hasan badu ali
```

```
shofira@shofira-VirtualBox:~$ . prog04.sh
Nama program adalah bash
Parameter 1 adalah
Parameter 2 adalah
Parameter 3 adalah
shofira@shofira-VirtualBox:~$ . prog04.sh amir hasan
Nama program adalah bash
Parameter 1 adalah amir
Parameter 2 adalah hasan
Parameter 3 adalah
shofira@shofira-VirtualBox:~$ . prog04.sh amir hasan badu ali
Nama program adalah bash
Parameter 1 adalah amir
Parameter 2 adalah hasan
Parameter 3 adalah badu
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `. prog04.sh` digunakan untuk menjalankan shell bernama `prog04.sh` dan hanya di eksekusi tanpa diikuti dengan parameter maka akan menampilkan nama program dengan `$0` adalah `bash`.
- Perintah `. prog04.sh amir hasan` digunakan untuk menjalankan shell bernama `prog04.sh` dan di eksekusi dengan diikuti 2 parameter maka akan menampilkan nama program dengan `$0` adalah `bash`, `$1` adalah `amir` dan `$2` adalah `hasan`.
- Perintah `. prog04.sh amir hasan badu ali` digunakan untuk menjalankan shell bernama `prog04.sh` dan di eksekusi dengan diikuti 4 parameter maka akan menampilkan nama program dengan `$0` adalah `bash`, `$1` adalah `amir`, `$2` adalah `hasan`, `$3` adalah `badu`, sedangkan `$4` tidak muncul karena belum dibuat pada shell script.

3. Membuat shell script prog04.sh versi 2 dengan memberikan jumlah parameter

```
$ vi prog04.sh
#!/bin/sh
# prog04.sh versi 2
# Parameter passing
#
echo "Jumlah parameter yang diberikan adalah $#"
```

```
echo "Nama program adalah $0"
echo "Parameter 1 adalah $1"
echo "Parameter 2 adalah $2"
echo "Parameter 3 adalah $3"
```

```
shofira@shofira-VirtualBox:~$ nano prog04.sh
shofira@shofira-VirtualBox:~$
```

```
GNU nano 6.2                                prog04.sh *
#!/bin/sh
# prog04.sh versi 2
# Parameter passing
#
echo "Jumlah parameter yang diberikan adalah $#"
```

Analisa :

Perintah nano prog04.sh digunakan untuk membuat shell script dengan nama file prog04.sh.

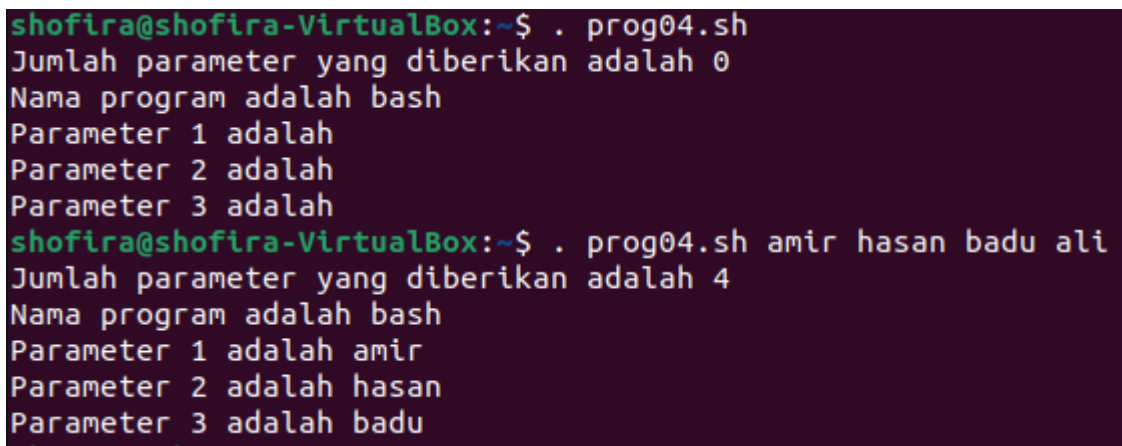
- #!/bin/sh — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya /bin/csh, /bin/ksh dan lainnya.
- # prog04.sh versi 2 — # prog04.sh versi 2 ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- # Parameter passing — # Parameter passing ini sebatas komentar saja mengenai

informasi program yg dijalankan, komentar tidak akan dieksekusi.

- # — tanda ini hanya sebagai komentar.
- NAMA=`whoami` digunakan untuk mengeksekusi perintah whoami dengan sepasang Back Quate (tanda kutip terbalik). Hasil dari eksekusi tersebut akan masuk sebagai nilai variable NAMA.
- echo "Jumlah parameter yang diberikan adalah \$# " digunakan untuk menampilkan output pada variabel \$#.
- echo "Nama program adalah \$0" digunakan untuk menampilkan output pada variabel \$0.
- echo "Parameter 1 adalah \$1" digunakan untuk menampilkan output pada variabel \$1.
- echo "Parameter 2 adalah \$2" digunakan untuk menampilkan output pada variabel \$2.
- echo "Parameter 3 adalah \$3" digunakan untuk menampilkan output pada variabel \$3.

4. Eksekusi prog04.sh tanpa parameter dan dengan 4 parameter

```
$ . prog04.sh
$ . prog04.sh amir hasan badu ali
```



```
shofira@shofira-VirtualBox:~$ . prog04.sh
Jumlah parameter yang diberikan adalah 0
Nama program adalah bash
Parameter 1 adalah
Parameter 2 adalah
Parameter 3 adalah
shofira@shofira-VirtualBox:~$ . prog04.sh amir hasan badu ali
Jumlah parameter yang diberikan adalah 4
Nama program adalah bash
Parameter 1 adalah amir
Parameter 2 adalah hasan
Parameter 3 adalah badu
```

Analisa :

- Perintah `. prog04.sh` digunakan untuk menjalankan shell bernama prog04.sh dan di eksekusi tanpa diikuti dengan parameter maka `$#` akan menampilkan 0 dan nama program dengan `$0` adalah bash.
- Perintah `. prog04.sh amir hasan badu ali` digunakan untuk menjalankan shell bernama prog04.sh dan di eksekusi sebagai 4 parameter maka `$#` akan menampilkan jumlah parameternya dan nama program dengan `$0` adalah bash, `$1` adalah amir, `$2` adalah hasan, `$3` adalah badu, sedangkan `$4` tidak muncul karena belum dibuat pada shell script.

5. Membuat shell script prog04.sh versi 3 dengan menambahkan total parameter dan nomor proses id (PID)

```
$ vi prog04.sh
#!/bin/sh
# prog04.sh versi 3
# Parameter passing
#
echo "Jumlah parameter yang diberikan adalah $#"
```

```
echo "Nama program adalah $0"
echo "Parameter 1 adalah $1"
echo "Parameter 2 adalah $2"
echo "Parameter 3 adalah $3"
echo "Total parameter adalah $*"
echo "PID proses shell ini adalah $$"
```

```
shofira@shofira-VirtualBox:~$ nano prog04.sh
shofira@shofira-VirtualBox:~$
```



```
GNU nano 6.2                                prog04.sh *
#!/bin/sh
# prog04.sh versi 3
# Parameter passing
#
echo "Jumlah parameter yang diberikan adalah $#"
```

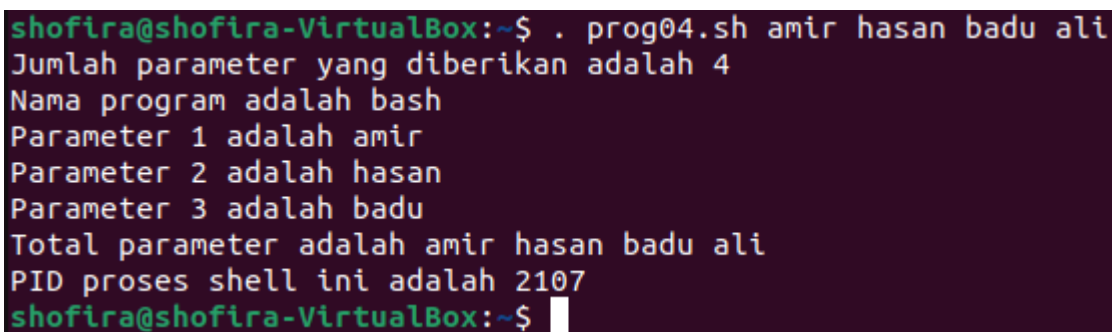
Analisa :

Perintah `nano prog04.sh` digunakan untuk membuat shell script dengan nama file prog04.sh.

- `#!/bin/sh` — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya `/bin/csh`, `/bin/ksh` dan lainnya.
- `# prog04.sh versi 3` — `prog04.sh` versi 3 ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `# Parameter passing` — Parameter passing ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `#` — tanda ini hanya sebagai komentar.
- `NAMA=`whoami`` digunakan untuk mengeksekusi perintah `whoami` dengan sepasang Back Quate (tanda kutip terbalik). Hasil dari eksekusi tersebut akan masuk sebagai nilai variable `NAMA`.
- `echo "Jumlah parameter yang diberikan adalah $#"` digunakan untuk menampilkan output pada variabel `$#`.
- `echo "Nama program adalah $0"` digunakan untuk menampilkan output pada variabel `$0`.
- `echo "Parameter 1 adalah $1"` digunakan untuk menampilkan output pada variabel `$1`.
- `echo "Parameter 2 adalah $2"` digunakan untuk menampilkan output pada variabel `$2`.
- `echo "Parameter 3 adalah $3"` digunakan untuk menampilkan output pada variabel `$3`.
- `echo "Total parameter adalah $*"` digunakan untuk menampilkan output seluruh parameter yang diinputkan menggunakan variabel `$*`.
- `echo "PID proses shell ini adalah $$"` digunakan untuk menampilkan output pada variabel `$$` untuk mendefinisikan nomor proses id (PID).

6. Eksekusi `prog04.sh` dengan 4 parameter

```
$ . prog04.sh amir hasan badu ali
```



```
shofira@shofira-VirtualBox:~$ . prog04.sh amir hasan badu ali
Jumlah parameter yang diberikan adalah 4
Nama program adalah bash
Parameter 1 adalah amir
Parameter 2 adalah hasan
Parameter 3 adalah badu
Total parameter adalah amir hasan badu ali
PID proses shell ini adalah 2107
shofira@shofira-VirtualBox:~$
```

Analisa :

Perintah `. prog04.sh amir hasan badu ali` digunakan untuk menjalankan shell bernama `prog04.sh` dan di eksekusi sebagai 4 parameter maka `$#` akan menampilkan jumlah parameternya dan nama program dengan `$0` adalah `bash`, `$1` adalah `amir`, `$2` adalah `hasan`, `$3` adalah `badu`, sedangkan `$4` tidak muncul karena belum dibuat pada shell script. Untuk total parameter adalah merupakan hasil untuk menampilkan `amir hasan badu` dengan instruksi `$*`. Sedangkan PID proses shell ini adalah ditampilkan dari perintah `$$`

Percobaan 5 : Status Exit

1. String tidak ditemukan, maka status exit adalah 1

```
$ grep xyz /etc/passwd
$ echo $?
```

```
shofira@shofira-VirtualBox:~$ grep xyz /etc/passwd
shofira@shofira-VirtualBox:~$ echo $?
1
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `grep xyz /etc/passwd` digunakan untuk mencari string dalam file `etc/passwd`.
- Perintah `echo $?` digunakan untuk menampilkan output pada variabel `$?`. Dan akan ditampilkan status exit dengan perintah `echo $?` adalah 1, artinya tidak ditemukan karena string yang tidak ditemukan pada perintah `grep` seperti diatas.

2. String ditemukan, maka status exit adalah 0

```
$ grep <user> /etc/passwd
$ echo $?
```

```
shofira@shofira-VirtualBox:~$ grep shofira /etc/passwd
shofira:x:1000:1000:shofira,,,:/home/shofira:/bin/bash
shofira@shofira-VirtualBox:~$ echo $?
0
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `grep <user> /etc/passwd` digunakan untuk mencari string dalam file `etc/passwd`.
- Perintah `echo $?` digunakan untuk menampilkan output pada variabel `$?`. Dan akan ditampilkan status exit dengan perintah `echo $?` adalah 0, artinya string ditemukan pada perintah `grep` seperti diatas.

Percobaan 6 : Konstruksi if

1. Instruksi dengan exit status 0

```
$ who
$ who | grep <user>
$ echo $?
```

```
shofira@shofira-VirtualBox:~$ who
shofira  tty2          2023-05-02 12:55 (tty2)
shofira@shofira-VirtualBox:~$ who | grep shofira
shofira  tty2          2023-05-02 12:55 (tty2)
shofira@shofira-VirtualBox:~$ echo $?
0
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `who` digunakan untuk menampilkan user yang sedang aktif.
- Perintah `who | grep <user>` digunakan untuk menampilkan user yang sedang aktif dan mencari string dalam file.
- Perintah `echo $?` digunakan untuk menampilkan output pada variabel `$?`. Dan akan ditampilkan status exit dengan perintah `echo $?` adalah 0, artinya string ditemukan pada perintah `grep` seperti diatas.

2. If membandingkan exit status dengan 0, bila sama, maka blok program masuk ke dalam blok then-fi

```
$ if [ $? = 0 ]
> then
>     echo "Pemakai tersebut sedang aktif"
> fi
```

```
shofira@shofira-VirtualBox:~$ if [ $? = 0 ]
> then
> echo "Pemakai tersebut sedang aktif"
> fi
Pemakai tersebut sedang aktif
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `if [$? = 0]` digunakan untuk membandingkan status exit dengan 0.
- `> then`
- `> echo "Pemakai tersebut sedang aktif"`
- `> fi`
- Apabila status exit sama dengan 0, maka program masuk ke dalam blok then if yaitu Pemakai tersebut sedang aktif, karena perintah sebelumnya user sedang aktif hasilnya 0. Perintah di atas menggunakan metode konstruksi if.

3. Nomor (1) dan (2) diatas dapat disederhanakan dengan

```
$ if who|grep <user> >/dev/null
> then
>     echo okay
> fi
```

```
shofira@shofira-VirtualBox:~$ if who|grep shofira >/dev/null
> then
> echo okay
> fi
okay
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `if who|grep <user> >/dev/null` digunakan untuk membelokkan ke `/dev/null` pada perintah `if who` dengan `grep user`.
- `> then`
- `> echo okay`
- `> fi`
- Maka apabila perintahnya berstatus exit 0 akan ditampilkan okay, karena perintah sebelumnya user sedang aktif maka hasilnya 0. Perintah di atas menggunakan metode konstruksi `if`.

Percobaan 7 : Konstruksi if then else

1. Membuat shell script prog05.sh

```
$ vi prog05.sh
#!/bin/sh
# prog05.sh
# Program akan memberikankonfirmasi apakah nama
# user sedang aktif atau tidak
#
echo -n "Berikan nama pemakai : "
read nama
if who | grep $nama > /dev/null
then
    echo "$nama sedang aktif"
else
    echo "$nama tidak aktif"
fi
```

```
shofira@shofira-VirtualBox:~$ nano prog05.sh
shofira@shofira-VirtualBox:~$
```

```
GNU nano 6.2                                prog05.sh *
#!/bin/sh
# prog05.sh
# Program akan memberikan konfirmasi apakah nama
# user sedang aktif atau tidak
#
echo -n "Berikan nama pemakai : "
read nama
if who | grep $nama > /dev/null
then
echo "$nama sedang aktif"
else
echo "$nama tidak aktif"
fi

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

Analisa :

Perintah `nano prog05.sh` digunakan untuk membuat shell script dengan nama file `prog05.sh`.

- `#!/bin/sh` — sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya `/bin/csh`, `/bin/ksh` dan lainnya.
- `# prog05.sh` — `prog05.sh` ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `# Program akan memberikan konfirmasi apakah nama` — ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `# user sedang aktif atau tidak` — ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
- `#` — tanda ini hanya sebagai komentar.
- `echo -n "Berikan nama pemakai : "` digunakan untuk mengatasi agar tidak memberikan baris baru karena secara default menulis `echo` pada script tanpa ada opsi tambahan, `echo` dijalankan akan memberikan baris baru.
- `read nama` digunakan untuk mengisi variabel melalui masukan keyboard.
- `if who | grep $nama > /dev/null` digunakan untuk membelokkan ke `/dev/null` pada perintah `if who` dengan `grep` dari variabel `$nama`.
- `then`
- `echo "$nama sedang aktif"`
- `else`
- `echo "$nama tidak aktif"`
- `fi`

- Kemudian then apabila \$nama sedang aktif atau else jika \$nama tidak aktif dan ditutup dengan fi.

2. Jalankan `prog05.sh`, masukkan nama pemakai yang aktif yang tampil pada instruksi `who` dan coba juga untuk nama pemakai yang tidak aktif

```
$ who
$ . prog05.sh [nama=<user>]
$ . prog05.sh [nama=studentOS]
```

```
shofira@shofira-VirtualBox:~$ nano prog05.sh
shofira@shofira-VirtualBox:~$ who
shofira tty2          2023-05-02 12:55 (tty2)
shofira@shofira-VirtualBox:~$ . prog05.sh
Berikan nama pemakai : shofira
shofira sedang aktif
shofira@shofira-VirtualBox:~$ . prog05.sh
Berikan nama pemakai : izza
izza tidak aktif
shofira@shofira-VirtualBox:~$
```

Analisa :

- Perintah `who` digunakan untuk menampilkan user yang sedang aktif. (misal user yang aktif adalah shofira)
- Perintah `. prog04.sh` digunakan untuk menjalankan shell bernama `prog04.sh` dan diisi nama pemakai dengan shofira, maka akan ditampilkan shofira sebagai nilai variabel nama yaitu shofira sedang aktif.
- Perintah `. prog04.sh` digunakan untuk menjalankan shell bernama `prog04.sh` dan diisi dengan nama pemakai student (selain shofira) maka instruksi ini dinyatakan tidak aktif karena user yang aktif adalah shofira.
- Jadi, \$nama sedang aktif adalah user yang sedang aktif merupakan hasil output dari perintah `who` dan jika yang dimasukkan bukan user yang sedang aktif maka \$nama tidak aktif.

C. KESIMPULAN

- Shell script dibuat dengan editor teks (ASCII editor) dan umumnya diberikan ekstensi “.sh”.serta Variabel shell adalah variable yang dapat mempunyai nilai berupa nilai String. Variable harus dimulai dengan alfabet, disusul dengan alfanumerik dan karakter lain. Variabel dapat ditulis dalam huruf kecil atau huruf besar atau campuran keduanya. Nilai variable dapat diisi melalui keyboard (stdin) dengan instruksi `read`.
- Setiap program setelah selesai dieksekusi akan memberikan informasi melalui variable spesial `$?`. Indikasi yang diberikan adalah :
 - o Bila program berakhir dengan sukses, `$? = 0`
 - o Bila program berakhir dengan error, `$? 0`

- Pada konstruksi if

if akan mengeksekusi instruksi-awal, dan exit status dari instruksi tersebut akan menjadi kondisi. Bila 0, maka instruksi selanjutnya masuk ke dalam blok then. Bila tidak 0, maka alur program diteruskan setelah kunci kata fi.