**LAPORAN**

**[Praktikum 07] Pemrograman Bash Shell 1**

Disusun Oleh:

Aldow Fan Dzikri (NRP. 3121500022)

1 D3 ITA

Dosen Pengampu:

Fitri Setyorini ST. M.Sc

**PROGRAM STUDI D3 TEKNIK INFORMATIKA**

**DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**2022**

**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

$ vi prog01.sh

#!/bin/shV1=poltek

V2=’:’V3=elektronika

echo “Pemrograman shell”

echo $V1$V2$V3

V3=ITS

echo $V1$V2 di $V3

• Prog02.sh

$ 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”

• 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

• 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

#!/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

**Percobaan 1 : Membuat shell script**

1. Buatlah file prog01.sh dengan editor vi

$ vi prog01.sh

#!/bin/sh

# Program shell

#

var1=x

var2=8 2.





**Analis:**

Untuk membuat shell script pada perintah diatas menggunakan editor vi karena terlihat diperintah ada kata vi dan dikuti dengan nama file prog01.sh, Kemudian, kita masukan kode dengan menekan huruf i pada keyboard, untuk kode sesuai dengan soal diatas, penjelasannya yaitu:

* #!/bin/sh 🡪 sebagai shell yang digunakan , kita juga dapat menggunakan selain shell itu, misalnya /bin/csh, /bin/ksh dan lainnya.
* # 🡪 Program shell ini sebatas komentar saja mengenai informasi program yg dijalankan, komentar tidak akan dieksekusi.
* # 🡪 ini juga hanya komentar saja.
* 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 menggangap sebuah parameter serta variabel bersifat case-sensitive.
* Untuk keluar dari editor vi sekaligus save kita dapat menekan terlebih dahulu esc untuk keluar dari mode editing untuk memasuki mode command. Setelah itu, ketikkan : dan wq.
1. Untuk menjalankan shell, gunakan notasi TITIK di depan nama program

$ . prog01.sh



**Analisis:**

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

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

$ chmod +x prog01.sh

$ ./prog01.sh



**Analisis:**

Demikian juga dengan ini tidak ada hasil yang ditampilkan meskipun menggunakan perintah yang berbeda.

**Percobaan 2 : Variabel**

1. Contoh menggunakan variable pada shell interaktif

$ VPT=poltek

$ echo VPT



**Analisis:**

Pada perintah diatas kita inputkan variabel bernama VPT dengan nilai string poltek menggunakan perintah VPT=poltek, setelah itu, kita bisa menampilkan output dari variabel tersebut dengan perintah echo $VPT, dengan perintah demikian akan mengeluarkan output berupa nilai dari variabel yang didefinisksan sebelumnya, yaitu poltek.

1. 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



**Analisis:**

Pada perintah diatas diperlihatkan bahwa terdapat output yang perlu dibahas, berikut ini pembahasan dari setiap perintah tersebut:

Perintah VPT2=poltek elektronika 🡪 setelah kita selesai mengekseskusi perintah akan munucl pesan “perintah tidak ditemukan” karena ketika kita menulis nilai variabel yang ada beberapa kata, yang diharuskan kita memberikan jarak antar kata dengan spasi, tetapi pada perintah ini melanggar kaidah aturan itu yang mana kita melanggar tidak ada tanda petik dua atas (“) yang seharusnya ada diawal nilai dan diakhir nilai variabel.

* Perintah VPT2=”poltek elektronika” 🡪 kalau perintah ini benar karena sudah mengikuti kaidah aturan.
* Perintah echo VPT2 🡪 sebetulnya untuk output dari perintah ini sudah benar berupa output VPT2 karena kita bukan 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 🡪 untuk perintah ini sudah benar kalau kita mengharapkan ingin menampilkan variabel yang sudah didefiniskan dengan format penulisan seperti perintah ini, yaitu di awal nama variabel terdapat tanda $ sebagai tanda bahwa hal ini adalah sebuah variabel.
1. Menggabungkan dua variable atau lebih

$ V1=poltek $ V2=’:’

$ V3=elektronika

$ V4=$V1$V2$V3

$ echo $V4



**Analisis:**

Pada perintah diatas kita menggabungkan beberapa variabel yang dimana kita membuat variabel $ V1=poltek $ V2=’:’ ,dan $ V3=elektronika terlebih dahulu dari ketiga variabel yang telah dibuat kita masukan ke variabel $V4 dengan perintah seperti ini $ V4=$V1$V2$V3, setelah tahap tersebut kita buat perintah echo $V4 untuk melihat output berupa nilai variabel $V4 yang meliputi nilai dari variabel $V1$V2$V3.

1. 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



**Analisis:**

Pada perintah ini perihal penggabungan variabel dengan string lain, jadi di pemrograman shell penggabungan variabel tidak hanya variabel dengan variabel saja. Akan tetapi, juga dapat menggabung variabel dengan string yang lainnya. Adapun contohnya seperti pada CLI diatas dimana pertama kali kita menulis perintah echo $V3$ yang menampilkan output “informatika” ini akan dilakukan percobaan terlebih dahulu untuk menggabungkan variabel $V3 dilanjutkan string tanpa spasi yang bertuliskan PENS untuk perintahnya seperti ini echo $V3PENS yang akan menghsilkan output kosong karena salah dalam pendefisian variabel untuk yang benar variabel harus dikurung dengan kurung kurwal dengan catatan tanda $ tidak termasuk dikurung seperti halnya pada perintah diatas.

5. Variabel dapat berisi instruksi, yang kemudian bila dijadikan input untuk

shell, instruksi tersebut akan dieksekusi

$ CMD=who

$ $CMD

$ CMD=”ls –l”

$ $CMD



**Analisis:**

Nilai dari sebuah variabel tidak hanya untuk menampilkan sesuatu pada CLI, akan tetapi, dalam program shell nilai dari variabel ini bisa di isi dengan instruksi dan apabila variabel yang di isi dengan sebuah intruksi ditulis di CLI akan mengekseskusi instrusksi sesuai dengan built-in yang diterapkan pada sistem operasi yang digunakan oleh users. Ini seperti hal-nya contoh diatas.

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



**Analisis:**Pada perintah diatas kita memodifikasi script dari file prog01.sh yang sebelumnya telah di buat dengan menggunakan editor vi.

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



**Analisis :**

Setelah selesai memodifikasi script pada file prog01.sh, jalanka file script dengan menggunakan perintah $ . prog01.sh akan keluar output berupa isi script dari file prog01.sh selain menggunakan perintah tersebut ada perintah ./prog1.sh tetapi file harus dijadikan executable terlebih dahulu, untuk menjadikan file prog01.sh menjadi file executable menggunakan perintah chmod +x <namafile> , dalam hal ini nama filenya adalah prog01.sh. dengan demikan baru bisa dijalankan file prog01.sh menggunakan perintah ./prog01.sh.

**Percobaan 3 : Membaca keyboard**

1. Menggunakan instruksi read

$ read nama

amir

$ echo $nama



**Analisis:**

Untuk mengisi variabel selain menggunakan perintah seperti ini $ variabel=<isi variabel> secara langsung, program shell dapat menggunakan perintah read untuk mengisi variabel melalui masukan keyboard. Contoh seperti perintah pada CLI diatas.

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”





**Analisis:**

Berdasarkan perintah diatas bahwa kita membuat script pada file prog02.sh menggunakan editor. Isi dari file ini terdapat kode read , dimana read ini nantinya dimunculkan untuk menginstruksikan users memasukan inputan pada keyboard.

3. Eksekusi program prog02.sh

$ . prog02.sh

Nama Anda : Amir

Alamat : Jl semangka 67

Kota :

Surabaya

Hasil adalah : Amir, Jl semangka di Surabaya



**Analisis:**

Setelah script dibuat maka selajutnya eksekusi program dg memasukan kata pada keyboard. Untuk hasilnya seperti diatas.

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



**Analisis:**

Secara default menulis echo pada script tanpa ada opsi tambahan, echo dijalankan akan memberikan baris baru, untuk mengatasi hal ini agar tidak memberikan baris baru dengan cara memberikan opsi -n pada echo, seperti CLI diatas.

5. Eksekusi program prog02.sh

$ . prog02.sh

Nama Anda : Amir

Alamat : Jl semangka 67

Kota : Surabaya

Hasil adalah : Amir, Jl semangka di Surabaya

**Analisis:**

Pada gambar CLI ditas adalah prog02.sh yang telah dijalankan sekaligus diberikan inputan pada keyboard. Dapat dilihat bahwa ketika users menginputkan pada keyboard tidak memberikan baris baru lagi, ini karena echo yang sudah ditambahkan opsi -n.

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



**Analisis:**

Pada gambar CLI diatas diperlihatkan variabel dapat dikosongi nilainya dengan cara seperti diatas.

7. Variabel dapat disubtitusikan 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



**Analisis:**

Instruksi biasanya kita jalankan pada CLI dapat kita masukan kedalam varibel, seperti pada gambar CLI diatas diperihatkan perintah $ pwd yang dijalnkan lebih dulu kemudian kita masukan perintah yang ditulis kedalam sebuah variabel DIR=`pwd` (dengan catatatan perintah yang dimasukan dalam variabel diapit oleh tanda kutip terbalik [`]), terakhir lihat hasilnya dengan perintah echo $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





**Analisis:**

perintah diatas diminta untuk membuat file dengan berisi coding yang sudah ditentukan sehingga kita hanya tinggal mengetikkan kodingan tersebut pada shell. Untuk outputnya pada perintah kali ini yaitu mengetahui nama pengguna dan mengetahui hari dan tanggal saat ini.

9. Eksekusi prog03.sh

$ . prog03.sh



**Analisis:**

Berikut hasil eksekusi percobaan yang sudah dibuat tadi. $NAMA menjalankan

intruksi whoami yang sudah sesuai pada perintah sebelumnya. sedangkan $tanggal

menampilkan intruksi date yang sudah di sort sesuai pada perintah yang ada pada shell.

**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”





**Analisis:**

Perintah diatas digunakan untuk membuat script prog04.sh versi 1

yang akanmenampilkan nama program dan parameter.

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

parameter

$ . prog04.sh

$ . prog04.sh amir hasan

$ . prog04.sh amir hasan badu ali



**Analisis:**

Program diatas merupakan perintah yang akan di eksekusi dengan menggunakan

2 parameter dan 4 parameter sehingga variable yang belum terisi akan terisi parameter

parameter tersebut.

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”





**Analisis:**

Perintah diatas merupakan perintah unuk membuat shell script versi 2 dengan

menambahkan jumlah parameter di yang di berikan.

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

$ . prog04.sh

$ . prog04.sh amir hasan badu ali



**Analisis:**

Perintah ini digunakan untuk mengeksekusi perintah sebelumnya dengan 4

parameter yang sudah disediakan. Pada percobaan ini di tampiljan jumlah parameter yang di berikan oleh user.

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 $$”





**Analisis:**

Pada percobaan ini digunakan untuk membuat script shell versi 5 dengan

menambahkan nomer proses id atau PID.

6. Eksekusi prog04.sh dengan 4 parameter

$ . prog04.sh amir hasan badu ali



**Analisis:**

Pada percobaan ini merupakan perintah untuk mengeksekusi perintah

sebelumnya. namun pada perintah ini ada sedikit perbedaan yaitu ditampilkannya nomer PID shell.

**Percobaan 5 : Status Exit**

1. String tidak diketemukan, maka status exit adalah 1

$ grep xyz /etc/passwd

$ echo $?



**Analisis:**

Program diatas merupakan program mencari string yang hasilnya tidak di

temukan sehingga outputnya adalah 1 yang artinya error.

2. String diketemukan, maka status exit adalah 0

$ grep <user> /etc/passwd

$ echo $?



**Analisis:**

Program di atas merupakan program mencari string yang hasilnya di temukan

sehingga status exitnya adalah 0 yang artinya berhasil.

**Percobaan 6 : Konstruksi if**

1. Instruksi dengan exit status 0

$ who

$ who | grep <user>

$ echo $?



**Analisis:**

perintah diatas digunakan untuk menampilkan user yang sedang aktif. Bila status

exitnya 0 maka program berakhir dengan sukses.

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



**Analisis:**

Perintah diatas digunakan untuk penentuan Jika dan maka , Bila status sama

dengan 0,maka kondisi menjadi TRUE dan instruksi setelah else akan dijalankan.

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

$ if who|grep <user> >/dev/null

> then

> echo okay

> fi



**Analisis:**

Perintah diatas merupakan perintah penyederhanaan nomor 1 dan 2 yang

menerapkan metode kontruksi if. Program diatas menampilkan outputan okay melalui

perintah if then.

**Percobaan 7 : Kontruksi 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



**Analisis:**

Perintah diatas digunakan untuk menampilkan nama yang sedang aktif menggunakan metode kontruksi if then else. Bila status exit tidak sama dengan 0, maka kondisi menjadi FALSE dan instruksi setelah else akan dijalankan.

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]



**Analisis:**

Pada percobaan kali ini diminta untuk.memasukkan nama yang tampil pada

instruksi untuk melihat nama yang aktif. Nama yang dikatakan sedang aktif adalah nama yang sesuai outputan dari who