

## A. TUJUAN

1. Menggunakan perintah `for`, `while`, `until`, dan `select` untuk menjalankan beberapa perintah secara berulang.
2. Melakukan proses LOOP dengan perintah `while`, `for`, dan `do while`.

## B. DASAR TEORI

### KONSTRUKSI FOR

`For` digunakan untuk pengulangan dengan menggunakan variabel (*name*) yang pada setiap pengulangan akan diganti dengan nilai yang berada pada daftar (*list = word1 word2 ...*). Sintak dari perintah `FOR` adalah sebagai berikut:

#### Sintak 1.

```
for name in word1 word2 ...  
do  
    do-list  
done
```

Contoh:

```
#!/bin/bash  
for buah in apel jeruk mangga salak  
do  
    echo $buah adalah buah  
done
```

Pada contoh program diatas variabel *\$buah* akan diganti dengan data pada list yaitu apel, jeruk, mangga dan salak.

### Sintak 2.

```
for name
do
    do-list
done
```

Contoh:

```
#!/bin/bash
for var
do
    echo $var
done
$./for2 satu 2 tiga
```

Contoh program menggunakan sintak2 variabel *\$var* akan diganti dengan data hasil pembacaan argument (satu, 2, tiga) yang disertakan saat script dijalankan.

## KONSTRUKSI WHILE

While digunakan untuk pengulangan instruksi, yang umumnya dibatasi dengan suatu kondisi. Selama kondisi tersebut TRUE, maka pengulangan terus dilakukan. Loop akan berhenti, bila kondisi FALSE, atau program keluar dari blok while melalui exit atau break. Sintak dari perintah WHILE adalah sebagai berikut:

### Sintak 1. While – end

```
while ( test_condition )
    commands /kumpulan perintah
end
```

Contoh:

```
set i=$#argv
while ($i)
    echo -n $argv[$i]
    @i--
end
```

Pada contoh program menggunakan sintak1 akan mencetak parameter yang diterima oleh program, tetapi dalam susunan terbalik karena nilai variabel "*i*" dikurangi satu persatu dimulai dari nilai yang tertinggi. Perintah ***echo -n*** digunakan agar setiap kali menampilkan satu parameter, parameter berikutnya tidak akan tercetak pada baris berikutnya.

## Sintak 2. While – do

```
while      [ test_condition ]  
do  
           commands  
done
```

Contoh:

```
i=1;  
while [ $i -le 10 ];  
do  
    echo "$i,";  
    let i=$i+2;  
done
```

Contoh program dengan sintak2 diatas menunjukkan kondisi tidak terpenuhi pada saat nilai i=11 (9+2), sehingga perintah dalam blok while tidak dieksekusi lagi dan nilai i=11 tidak pernah ditampilkan pada layar.

## INSTRUKSI DUMMY

Instruksi dummy adalah instruksi yang tidak melakukan apa-apa, namun instruksi ini memberikan status exit 0 (TRUE). Oleh karena itu, instruksi dummy dapat digunakan sebagai kondisi forever pada loop (misalnya ***while***).

Simbol instruksi dummy adalah → :

## KONSTRUKSI UNTIL

Jika while akan mengulang selama kondisi benar, lain halnya dengan statement until yang akan mengulang selama kondisi salah, berikut contoh script menggunakan ***until***

**Sintak**

```
until condition
do
    list
done
```

Contoh:

```
i=1;
until [ $i -gt 10 ];
do
    echo "$i,";
    let i=$i+1;
done
```

perhatikan kondisi until yang salah [ \$i -gt 10], dimana nilai awal i=1 dan akan berhenti apabila nilai i = 11 (bernilai benar) 11 -gt 10.

**KONSTRUKSI SELECT**

Select berguna untuk pembuatan layout berbentuk menu pilihan, sewaktu dijalankan bash akan menampilkan daftar menu yang diambil dari item list.

**Sintak**

```
select varname in (item list)
do
    commands
done
```

**C. PROSEDUR**

1. Masuk ke sistem operasi Linux.
2. Login sebagai stD3XXYYY.
3. Gunakan perintah-perintah perulangan pada pemrograman shell : For ,While dan Select.
4. Catatlah hasil dari setiap perintah pada prosedur no.3 sebagai laporan sementara.

## D. PERCOBAAN

### Percobaan 1 : Statement For

1. Pengulangan untuk mengambil data pada list (apel, jeruk, mangga dan salak)

```
$ vi for1.sh
#!/bin/bash
for buah in apel jeruk mangga salak
do
    echo $buah adalah buah
done
```

2. Pengulangan untuk mengambil data pada list (1,2,...,5)

```
$ vi for2.sh
#!/bin/bash
for angka in 1 2 3 4 5;
do
    echo "angka=$angka";
done
```

3. Pengulangan untuk mengambil data pada list

```
$ vi for3.sh
#!/bin/bash
for i in hello 1 * 2 goodbye
do
    echo "Looping ... i is set to $i"
done
```

4. Pembacaan argument (satu 2 tiga) yang disertakan saat script dijalankan

```
$ vi for4.sh
#!/bin/bash
for var
do
    echo $var
done

$ ./for4 satu 2 tiga
```

5. Pembacaan argument (dari posisi terakhir ke pertama) yang disertakan saat script dijalankan

```
$ vi for5.sh
#!/bin/bash
list=" "
for arg
do
    list="$arg $list"
done
```

```
echo $list
```

```
$ ./for5.sh Rina teman Rani
```

6. Menampilkan isi direktori */etc/passwd*

```
$vi for6.sh
```

```
#!/bin/bash
```

```
for var in `cat /etc/passwd`
```

```
do
```

```
    echo $var
```

```
done
```

```
$cat /etc/passwd (bandingkan hasilnya dengan program for6.sh)
```

7. Pemakaian statement *for* dan *case* untuk pembacaan argument

```
$vi for7.sh
```

```
#!/bin/bash
```

```
for breed
```

```
do
```

```
    case $breed in
```

```
        arabian|palomino|clydesdale) echo $breed is a horse ;;
```

```
        jersey|guernsey|holstein) eco $breed is a cow ;;
```

```
        huskey|shepherd|setter|Labrador) echo $breed is a dog ;;
```

```
        Siamese|Persian|angora) echo $breed is a cat ;;
```

```
        *) echo $breed is not in our catalog ;;
```

```
    esac
```

```
done
```

```
$./for7.sh husky holstein terrier
```

8. Konstruksi *for* dan *wildcard* untuk menampilkan file pada *current directori*

```
$ vi for8.sh
```

```
#!/bin/bash
```

```
for F in *
```

```
do
```

```
    echo $F
```

```
done
```

9. Menampilkan *long list* dari file yang mempunyai ekstensi *lst*

```
$ vi for9.sh
```

```
#!/bin/bash
```

```
for F in *.lst
```

```
do
```

```
    ls -l $F
```

```
done
```

10. Menampilkan file pada *current directori*

```
$ vi fora.sh
```

```
#!/bin/bash
```

```
for filename in *
```

```
do
    wc -c $filename
done
```

## Percobaan 2 : Statement while

1. Membaca argument yang disertakan saat script dijalankan

```
$ vi whil1.sh

#!/bin/csh
set i=$#argv
while ($i)
    echo -n $argv[$i]
    @ i--
end

$ ./whil1.sh Nama saya adalah Bagus Fajar
```

2. Mengcopy file ke direktori */tmp*

```
$ vi whil2.sh

#!/bin/csh
set i=$#argv
while ($i)
    cp $argv[$i] /tmp
    @ i--
end

$ ls -l /tmp
```

3. Loop dengan *while* dan perintah *test*

```
$ vi whil3.sh

#!/bin/bash
count=$#
cmd=echo
while test $count -gt 0
do
    cmd="$cmd \$$count"
    count=`expr $count - 1`
done
eval $cmd

$ ./whil3.sh 2 4 6 8 1 3 5 7
```

4. Loop dengan *while* dan perintah *break*

```
$ vi whil4.sh

#!/bin/bash
count=$#
cmd=echo
while true
do
    cmd="$cmd \$$count"
```

```

        count=`expr $count - 1`
        if test $count -eq 0
        then break
        fi
    done
    eval $cmd

$ ./whil4.sh 2 4 6 8 1 3 5 7

```

5. Loop dengan *while* dan perintah *continue*

```

$ vi whil5.sh

#!/bin/bash
count=$#
cmd=echo
while true
do
    cmd="$cmd \$$count"
    count=`expr $count - 1`
    if test $count -eq 0
    then continue
    fi
    eval $cmd
    exit
done

$ ./whil5.sh 2 4 6 8 1 3 5 7 (bandingkan dgn no 3 & 4)

```

6. Proses pengulangan/loop untuk mencetak string

```

$ vi whil6.sh

#!/bin/bash
while let "ulang <=3"
do
    echo $ulang Selamat belajar
    let "ulang=ulang+1"
done

```

7. Proses pengulangan/loop untuk mencetak karakter

```

$ vi whil7.sh

#!/bin/bash
let "ulang=1"
while let "ulang <=10"
do
    echo -n $ulang
    let "ulang=ulang+1"
    echo -n " "
done

```

8. Loop dengan *while* dan perintah *break* untuk informasi pemakai

```

$ vi whil8.sh

```



```
#!/bin/sh
PILIH=1
while [ $PILIH -ne 4 ]
do
    echo "1. Siapa yang aktif"
    echo "2. Tanggal hari ini"
    echo "3. Kalender bulan ini"
    echo "4. Keluar"
    echo " Pilihan : \c"
    read PILIH
    if [ $PILIH -eq 4 ]
    then
        break
    fi
    clear
done
echo "Program berlanjut di sini setelah break"
```

9. Loop dengan *while* dan perintah *dummy* ( : )

```
$ vi whil9.sh

#!/bin/sh
PILIH=1
while :
do
    echo "1. Siapa yang aktif"
    echo "2. Tanggal hari ini"
    echo "3. Kalender bulan ini"
    echo "4. Keluar"
    echo " Pilihan : \c"
    read PILIH
    if [ $PILIH -eq 4 ]
    then
        break
    fi
    clear
done
echo "Program berlanjut di sini setelah break"
```

(bandingkan dengan program no 8)

10. Proses loop untuk menghitung banyak bilangan ganjil dan genap

```
$ vi whila.sh

#!/bin/bash
i=0;
bil_genap=0;
bil_ganjil=0;
echo -n "Batas loop : ";
read batas
if [ -z $batas ] || [ $batas -lt 0 ]; then
```

```

    echo "Ops, tidak boleh kosong atau Batas loop
    harus >= 0";
    exit 0;
fi
while [ $i -le $batas ];
do
    echo -n "$i,";
    if [ `expr $i % 2` -eq 0 ]; then
        let bil_genap=$bil_genap+1;
    else
        let bil_ganjil=$bil_ganjil+1;
    fi
    let i=$i+1;    #counter untuk mencapai batas
done
echo
echo "banyak bilangan genap  = $bil_genap";
echo "banyak bilangan ganjil = $bil_ganjil";

```

### Percobaan 3 : Statement until dan select

#### 1. Proses loop dengan until

```

$ vi unt1.sh
#!/bin/bash
i=1;
until [ $i -gt 10 ];
do
    echo $i;
    let i=$i+1
done

```

#### 2. Proses pengulangan dengan *until* untuk mencetak string *hello* dengan

```

$ until test -r stopfile ; do
> sleep 2 ; echo Hello ; done &

```

#### 3. Statement until dan perintah dasar shell

```

$ vi unt2.sh
command="init" # Initialization.
until [ "$command" = "exit" ]
do
    echo -n "Enter command or \"exit\" to quit: "
    read command
    echo
case $command in
ls)
    echo "Command is ls."
;;
who)
    echo "Command is who."
;;

```

```

*)
  if [ $command != "exit" ]
  then
    echo "Why did you enter $command?"
  fi
  ;;
esac
done

```

4. Statement select untuk menampilkan daftar menu pada variable *reply* dengan menyertakan item list saat script dijalankan

```

$ vi sel1.sh

#!/bin/bash
clear
select menu
do
  echo "Anda memilih $REPLY yaitu $menu"
done

$ ./sel1.sh Slackware Redhat Mandrake

```

5. Statement select untuk menampilkan daftar menu pada variable *reply* tanpa menyertakan item list saat script dijalankan

```

$ vi sel2.sh

#!/bin/bash
lagi='y'
while [ $lagi == 'y' ] || [ $lagi == 'Y' ];
do
  clear
  select menu in "Bakso" "Gado-Gado" "Exit";
  case $REPLY in
    1) echo -n "Banyak mangkuk =";
       read jum
       let bayar=jum*1500;
       ;;
    2) echo -n "Banyak porsi =";
       read jum
       let bayar=jum*2000;
       ;;
    3) exit 0
       ;;
    *) echo "Sorry, tidak tersedia"
       ;;
  esac
done

echo "Harga bayar = Rp. $bayar"
echo "THX"
echo

```

```

echo -n "Hitung lagi (y/t) :";
read lagi;

    #untuk validasi input
    while [ $lagi != 'y' ] && [ $lagi != 'Y' ] && [
$lagi != 't' ] && [ $lagi != 'T' ];
    do
        echo "Ops, isi lagi dengan (y/Y/t/Y)";
        echo -n "Hitung lagi (y/t) :";
        read lagi;
    done
done

```

## E. Latihan

1. Buatlah program bash script menggunakan statement **for** untuk menyalin file yang berekstensi doc (.doc) ke file baru dengan ekstensi bak (.bak)

contoh hasil running program adalah sebagai berikut:

```

Copying 2005_Q1.doc to 2005_Q1.doc.bak
Copying 2005_Q2.doc to 2005_Q2.doc.bak
Copying business_plan.doc to business_plan.doc.bak
dst

```

2. Buatlah bash script dari running program dengan tampilan sebagai berikut:

```

Row 1: 1 2 3 4 5 6 7 8 9 10
Row 2: 1 2 3 4 5 6 7 8 9 10
Row 3: 1 2 3 4 5 6 7 8 9 10
Row 4: 1 2 3 4 5 6 7 8 9 10
Row 5: 1 2 3 4 5 6 7 8 9 10
Row 6: 1 2 3 4 5 6 7 8 9 10

```