

DOKUMENTASI PRAKTIKUM SISTEM OPERASI WEEK-4

Nama : Novita Anggraini

NPM : 21083010104

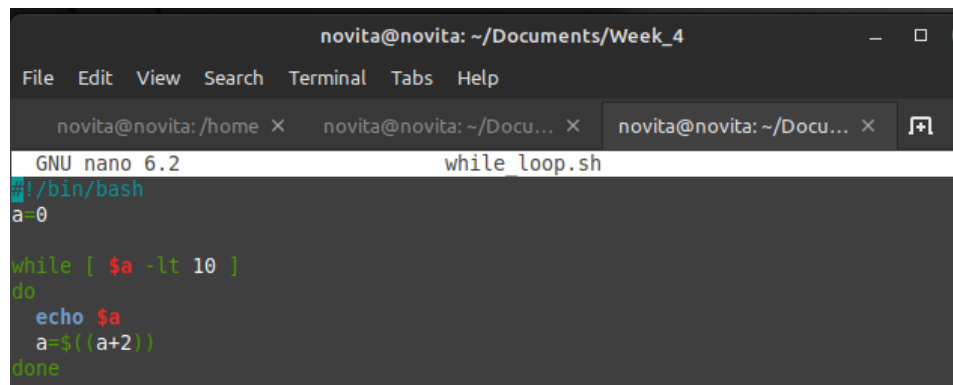
Mata Kuliah : Sistem Operasi A

PERULANGAN/LOOP

Loop atau perulangan adalah sebuah urutan perintah yang secara menerus diulang hingga suatu kondisi tercapai. Terdapat beberapa jenis perulangan, dalam pertemuan kali ini, akan dibahas 4 jenis perulangan.

1. WHILE LOOP

- Menjalankan command `nano nama_file.sh` untuk membuka text editor.
- Membuat script yang memanfaatkan while-loop.

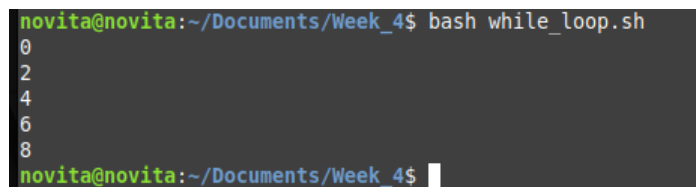
A screenshot of a terminal window with a nano text editor open. The title bar shows 'novita@novita: ~/Documents/Week_4'. The editor has tabs for 'novita@novita: /home', 'novita@novita: ~/Docu...', and 'novita@novita: ~/Docu...'. The current file is 'while loop.sh'. The code inside the editor is:

```
GNU nano 6.2 while loop.sh
#!/bin/bash
a=0

while [ $a -lt 10 ]
do
    echo $a
    a=$((a+2))
done
```

Set nilai `a=0` untuk inisiasi, gunakan command `while [kondisi]`, dalam kasus ini, kondisi yang digunakan adalah program harus mencetak `a` hingga `a` mencapai kurang dari sama dengan 10, lalu menambah `a` dengan 2 di setiap iterasinya.

- Output yang dihasilkan ketika program di-run menggunakan command `bash while_loop.sh`

A screenshot of a terminal window showing the execution of the script. The prompt is 'novita@novita:~/Documents/Week_4\$'. The command entered is 'bash while_loop.sh'. The output is:

```
0
2
4
6
8
novita@novita:~/Documents/Week_4$
```

Jelas bahwa nilai `a` awal terdefiniskan 0, kemudian melalui perulangan dengan penambahan 2 hingga `a` mencapai kurang dari 10.

a = 0
a = 2
a = 4
a = 6
a = 8

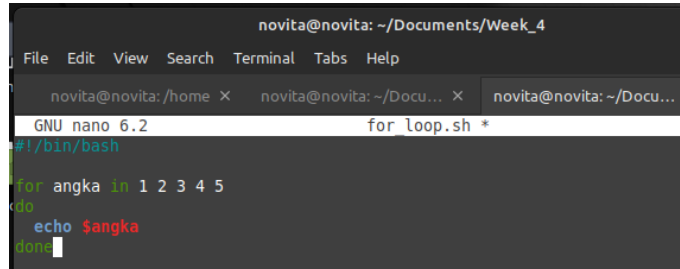
2. FOR LOOP

For Loop 1.

- Menjalankan command `nano nama_file.sh` untuk membuka text editor.

```
novita@novita:~/Documents/Week_4$ nano for_loop.sh
```

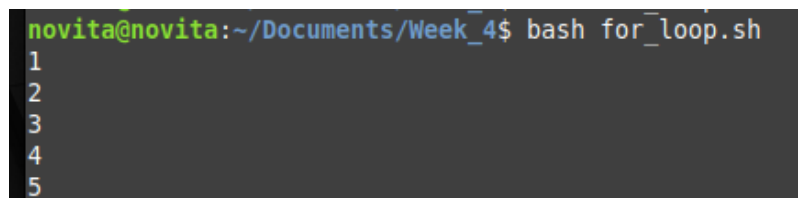
- Membuat script yang memanfaatkan for-loop.



The screenshot shows the nano text editor interface. The title bar indicates the file is `for_loop.sh`. The editor content is as follows:

```
#!/bin/bash
for angka in 1 2 3 4 5
do
    echo $angka
done
```

- Gunakan command `for (nama variabel) in (list objek)` dengan spasi sebagai pemisah antara objek satu dengan objek lainnya. Baris pertama mendeklarasikan kondisi yang harus dicapai selama dilakukan perulangan. Kemudian command `do` digunakan untuk melaksanakan perintah yang harus dilakukan hingga seluruh kondisi `for` terpenuhi.
- Dalam kasus di atas, perulangan akan dilakukan untuk variabel `angka` yang bernilai 1, 2, 3, 4, 5. Perintah yang dilakukan adalah mem-*print out* nilai yang ada di dalam variabel `angka`, yaitu nilai 1, 2, 3, 4, 5.
- Sintaks `done` untuk mengakhiri perulangan.
- Output yang dihasilkan



The screenshot shows the terminal output after running the script with `bash for_loop.sh`. The output is a vertical list of numbers from 1 to 5.

```
novita@novita:~/Documents/Week_4$ bash for_loop.sh
1
2
3
4
5
```

Output yang setelah program dijalankan adalah:

1
2
3
4
5

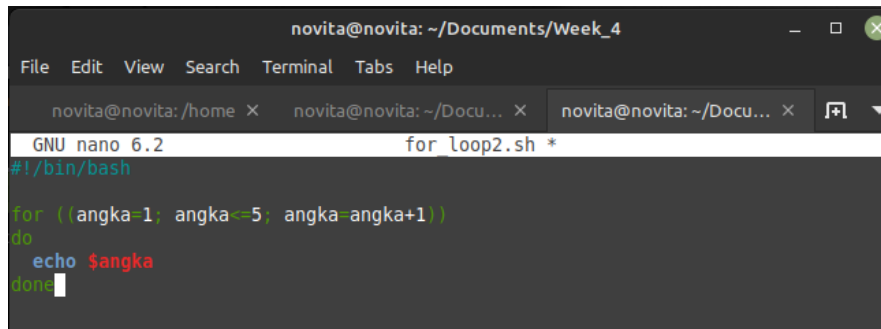
Bisa demikian karena angka-angka di atas adalah nilai yang tersimpan dalam variabel `angka`. Dikarenakan terdapat perintah `echo $angka`, maka nilai yang tersimpan akan keluar sebagai output.

For Loop 2.

- Menjalankan command `nano nama_file.sh` untuk membuka text editor.

```
novita@novita:~/Documents/Week_4$ nano for_loop2.sh
```

- Membuat script yang memanfaatkan perulangan for loop.

A screenshot of a terminal window with the nano text editor open. The window title is 'novita@novita: ~/Documents/Week_4'. The editor shows a script for 'for_loop2.sh' with the following content:

```
GNU nano 6.2 for_loop2.sh *
#!/bin/bash

for ((angka=1; angka<=5; angka=angka+1))
do
    echo $angka
done
```

- Gunakan command `for ((variabel=inisiasi; variabel kondisi; variabel perintah loop))` untuk mendefinisikan for hingga kondisi terpenuhi selama dilakukan perulangan. Kemudian command `do` digunakan untuk melaksanakan perintah yang harus dilakukan.
- Dalam kasus ini, kondisi yang diminta adalah ketika variabel `angka` memiliki nilai awal 1, lalu assign kondisi bahwa perulangan hingga `angka` mencapai nilai kurang dari sama dengan 5, dan di setiap perulangannya, variabel `angka` akan bertambah 1. Perintah yang dilakukan adalah mem-*print out* nilai yang ada di dalam variabel `angka`.
- Output yang dihasilkan:

```
novita@novita:~/Documents/Week_4$ bash for_loop2.sh
1
2
3
4
5
```

Output yang setelah program dijalankan adalah:

1
2
3
4
5

Bisa demikian karena angka-angka di atas adalah nilai yang tersimpan dalam variabel `angka`.

`angka = 1` = 1 (nilai inisiasi)

`angka = 1 + 1` = 2

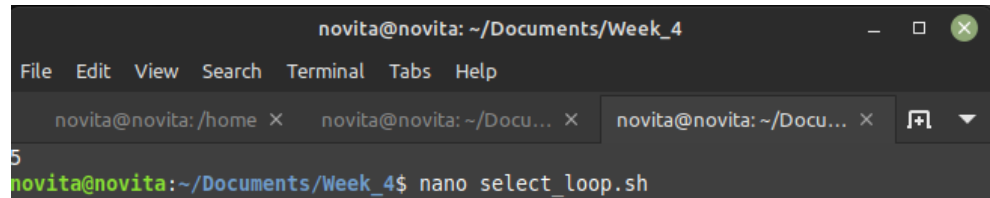
`angka = 2 + 1` = 3

`angka = 3 + 1` = 4

`angka = 4 + 1` = 5 (nilai maksimal)

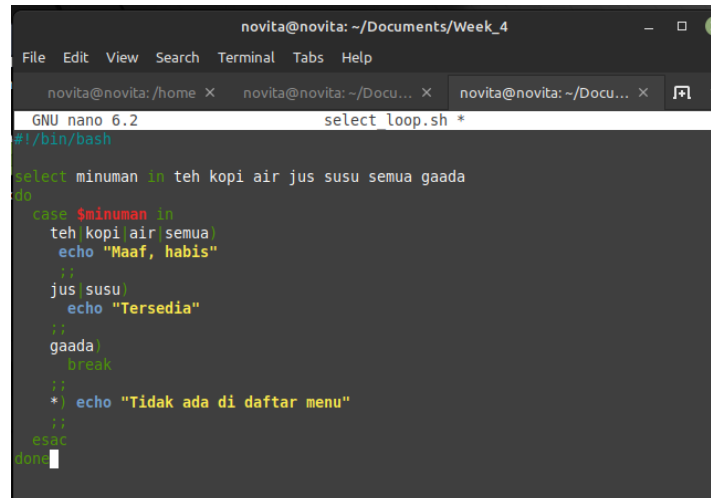
3. SELECT LOOP

- Menjalankan command `nano nama_file.sh` untuk membuka text editor.



```
novita@novita: ~/Documents/Week_4
File Edit View Search Terminal Tabs Help
novita@novita: /home X novita@novita: ~/Docu... X novita@novita: ~/Docu... X
5
novita@novita:~/Documents/Week_4$ nano select_loop.sh
```

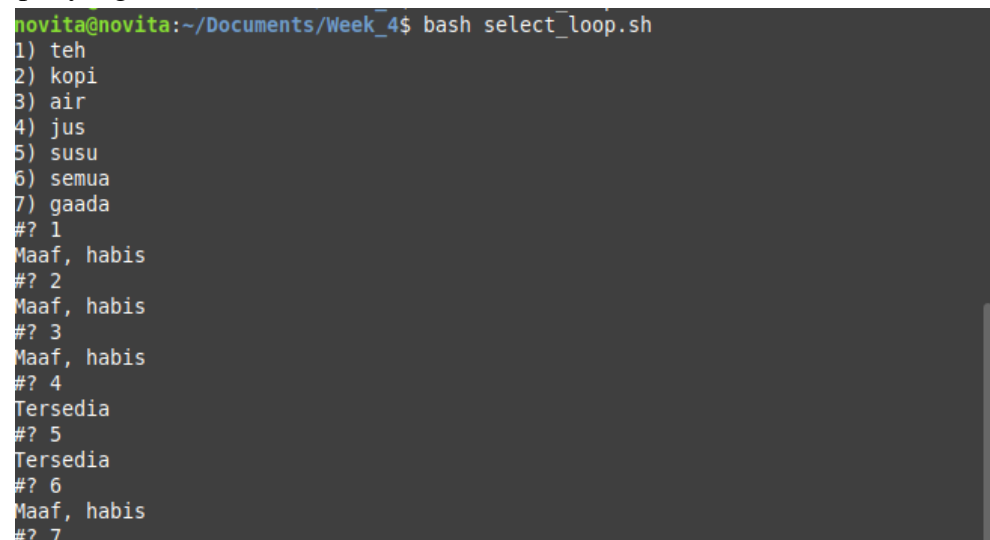
- Membuat script yang memanfaatkan perulangan select loop.



```
novita@novita: ~/Documents/Week_4
File Edit View Search Terminal Tabs Help
novita@novita: /home X novita@novita: ~/Docu... X novita@novita: ~/Docu... X
GNU nano 6.2 select_loop.sh *
#!/bin/bash

select minuman in teh kopi air jus susu semua gaada
do
    case $minuman in
        teh kopi air semua
            echo "Maaf, habis"
            ;;
        jus susu)
            echo "Tersedia"
            ;;
        gaada)
            break
            ;;
        *) echo "Tidak ada di daftar menu"
            ;;
    esac
done
```

- Gunakan command `select (variabel) in (list objek)` untuk membuat kondisi select. Dalam kasus ini, variabel yang akan digunakan adalah minuman dengan list objek teh, kopi, air, jus, susu, semua, dan gaada; dikodekan dengan angka.
- Perintah yang akan dilakukan adalah dengan menampilkan output sesuai dengan kondisi yang telah kita tentukan. Jika user memilih teh/kopi/air/semua, maka sistem akan mem-print out string “Maaf, habis”. Jika user memilih jus/susu, maka sistem akan mem-print out “Tersedia”. Dan jika user memilih gaada, maka sistem akan mem-print out string “Tidak ada di daftar menu”.
- Output yang dihasilkan.



```
novita@novita:~/Documents/Week_4$ bash select_loop.sh
1) teh
2) kopi
3) air
4) jus
5) susu
6) semua
7) gaada
#? 1
Maaf, habis
#? 2
Maaf, habis
#? 3
Maaf, habis
#? 4
Tersedia
#? 5
Tersedia
#? 6
Maaf, habis
#? 7
```

4. UNTIL LOOP

- Menjalankan command `nano nama_file.sh` untuk membuka text editor.

```
novita@novita:~/Documents/Week_4$ nano until_loop.sh
```

- Membuat script yang memanfaatkan perulangan until loop.

```
GNU nano 6.2          until_loop.sh *
#!/bin/bash

a=0

until [ ! $a -lt 10 ]
do
    echo $a
    a=$((a+2))
done
```

- Set nilai `a = 0` untuk menginisiasi nilai. Gunakan command `until [kondisi]` untuk membuat kondisi perulangan.
- Dalam menggunakan command `until`, harus terdapat nilai awal yang dideklarasikan sebelum masuk ke baris kondisi.
- Kondisi yang digunakan dalam kasus ini adalah menyimpan nilai `a` hingga mencapai `-lt` (less than/kurang dari) 10.
- Dengan perintah yang harus dilakukan adalah mem-*print out* nilai `a`, dengan penambahan nilai `a` oleh 2 di setiap perulangannya.
- Output yang dihasilkan.

```
novita@novita:~/Documents/Week_4$ bash until_loop.sh
0
2
4
6
8
```

Demikian karena jelas bahwa nilai `a` awal terdefiniskan 0, kemudian melalui perulangan dengan penambahan 2 hingga `a` mencapai 10.

$a = 0 = 0$ (nilai inisiasi)
 $a = 0 + 2 = 2$
 $a = 2 + 2 = 4$
 $a = 4 + 2 = 6$
 $a = 6 + 2 = 8$ (nilai maksimal)

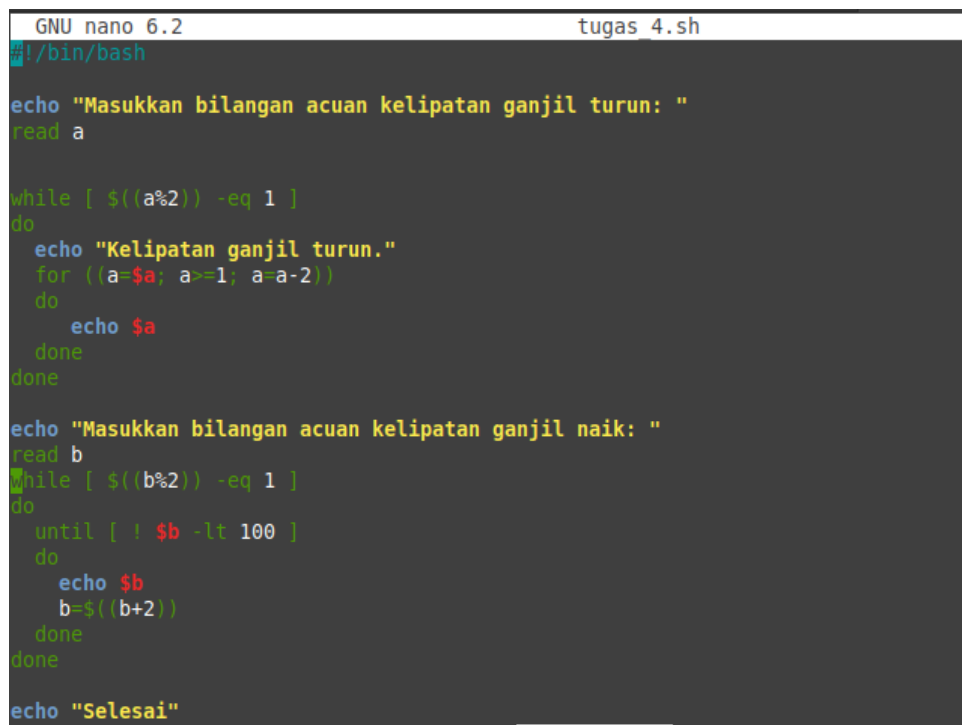
SOAL LATIHAN

Buatlah program perulangan yang menerapkan beberapa konsep pemrograman bash seperti di atas dengan ketentuan sbb:

- User menginputkan sebuah bilangan
- Mencetak bilangan positif kelipatan ganjil 3, 5, 7, 9 dari bilangan acuan
- Jalankan command `nano nama_file.sh` untuk membuat file baru dan membuka text editor.

```
novita@novita:~/Documents/Week_4$ nano tugas_4.sh
```

- Membuat script bash dengan memanfaatkan konsep perulangan.



```
GNU nano 6.2          tugas_4.sh
#!/bin/bash

echo "Masukkan bilangan acuan kelipatan ganjil turun: "
read a

while [  $((a\%2)) -eq 1$  ]
do
    echo "Kelipatan ganjil turun."
    for ((a=$a; a>=1; a=a-2))
    do
        echo $a
    done
done

echo "Masukkan bilangan acuan kelipatan ganjil naik: "
read b
while [  $((b\%2)) -eq 1$  ]
do
    until [  $\$b -lt 100$  ]
    do
        echo $b
        b=$((b+2))
    done
done

echo "Selesai"
```

- Dalam tugas ini, digunakan 3 konsep loop; while loop, for loop, dan until loop.
- Terdapat dua kelompok kode program. Kode pertama untuk membuat kelipatan ganjil dengan nilai yang semakin kecil hingga angka ganjil positif yaitu 1. Kode kedua untuk membuat kelipatan ganjil dengan nilai yang semakin besar hingga mencapai nilai maksimal 100.
- While loop mengatur kondisi agar program berjalan hanya ketika bilangan acuan bernilai negatif. Hal ini bisa diketahui jika modulo bilangan dengan angka 2 menyisakan nilai 1.
- For loop mengatur kondisi agar variabel a berkurang 2 setiap perulangannya hingga mencapai angka 1.
- Until loop mengatur kondisi agar variabel b bertambah 2 setiap perulangannya hingga mencapai angka 100.

- Output yang dihasilkan:

```
novita@novita:~/Documents/Week_4$ bash tugas_4.sh
Masukkan bilangan acuan kelipatan ganjil turun:
```

Masukkan bilangan ganjil untuk melakukan proses kelipatan ganjil turun. Di sini akan di-assign angka 21 sebagai bilangan acuan.

```
novita@novita:~/Documents/Week_4$ bash tugas_4.sh
Masukkan bilangan acuan kelipatan ganjil turun:
21
Kelipatan ganjil turun.
21
19
17
15
13
11
9
7
5
3
1
```

Demikian karena perulangan dimulai dari angka 21.

$a = 21$

$a = 21 - 2 = 19$

$a = 19 - 2 = 17$

...

$a = 5 - 2 = 3$

$a = 3 - 2 = 1$

Masukkan bilangan ganjil lainnya untuk melakukan proses kelipatan ganjil naik. Di sini akan di-assign angka 91 sebagai bilangan acuan.

```
Masukkan bilangan acuan kelipatan ganjil naik:
91
91
93
95
97
99
```

Demikian karena perulangan dimulai dari angka 91.

$b = 91$

$b = 91 + 2 = 93$

$b = 93 + 2 = 95$

$b = 95 + 2 = 97$

$b = 97 + 2 = 99$

Program tidak akan berjalan jika kita memasukkan angka selain angka ganjil.

```
novita@novita:~/Documents/Week_4$ bash tugas_4.sh
Masukkan bilangan acuan kelipatan ganjil turun:
22
Masukkan bilangan acuan kelipatan ganjil naik:
88
Selesai
```

Demikian karena dalam kondisi while hanya memperbolehkan nilai modulo bersisa satu, dalam kata lain bilangan ganjil saja yang bisa diproses oleh program.