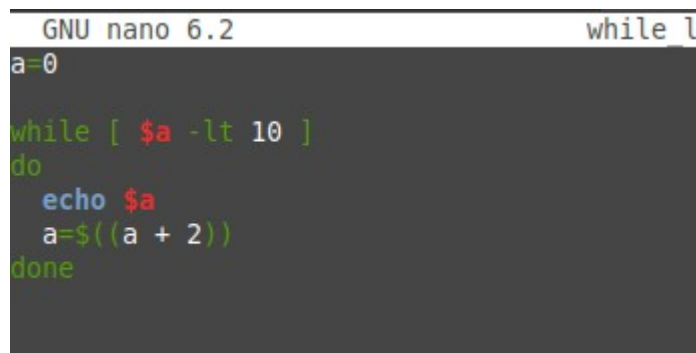


## 1. While Loop

While loop merupakan salah satu metode perulangan yang bisa kita pakai pada command linux. Pada metode ini digunakan saat kita ingin mengeksekusi perintah berulang sampai kondisi yang telah kita tetapkan terpenuhi. Caranya kita bisa membuat file bash terlebih dahulu menggunakan command nano.

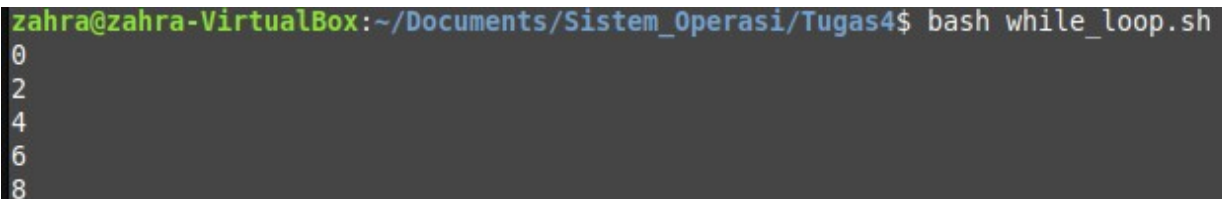
```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ nano while_loop.sh
```

Setelah itu kita bisa mendefinisikan variabel yang kita inginkan lalu kita ketik while dan kondisi yang kita inginkan. Lalu pada baris selanjutnya kita ketik do agar dalam loop tersebut bisa mengeksekusi perintah kita lalu baris selanjutnya perintah yang kita inginkan dan diakhiri dengan baris yang berisi kata done.

A screenshot of a terminal window showing the GNU nano 6.2 editor. The file being edited is named 'while\_l'. The script content is as follows:

```
a=0
while [ $a -lt 10 ]
do
    echo $a
    a=$((a + 2))
done
```

Mari kita buktikan dengan menjalankan kode tersebut. Pada gambar dibawah ini merupakan hasil dari kode yang kita jalankan. Gambar tersebut mengeluarkan output yang kita inginkan yaitu mengeluarkan angka 0 sampai 8 dengan lompat 2.

A screenshot of a terminal window showing the command 'bash while\_loop.sh' being executed. The output of the script is displayed as a list of numbers from 0 to 8, incrementing by 2:

```
0
2
4
6
8
```

## 2. For Loop 1

For Loop merupakan salah satu metode perulangan yang ada pada bash linux. Metode ini digunakan untuk mengulangi perintah yang kita inginkan pada sebuah item item yang terdapat pada variabel tersebut. Terdapat dua cara untuk menuliskan metode ini. Cara yang pertama yaitu untuk melakukannya kita bisa membuat file bash terlebih dahulu dengan command nano.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ nano for_loop1.sh
```

Setelah itu kita tulis for untuk mendefinisikan loop for kemudian nama variabel dilanjut menuliskan in yang berarti pada kemudian menuliskan daftar item. Pada baris selanjutnya kita menuliskan do yang bertujuan agar perintah kita dieksekusi pada baris selanjutnya yaitu perintah yang ingin kita jalankan dan diakhiri dengan done pada baris terakhir.

```
GNU nano 6.2
for angka in 1 2 3 4 5
do
    echo $angka
done
```

Setelah menuliskan kode tersebut pada filebash kita maka akan keluar output yang telah kita perintahkan seperti dibawah ini.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ bash for_loop1.sh
1
2
3
4
5
```

### 3. For Loop 2

Seperti yang telah dijelaskan tadi, terdapat dua cara untuk menuliskan metode perulangan ini. Cara yang kedua yaitu pertama tama kita bisa membuat file bash seperti biasa.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ nano for_loop2.sh
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$
```

Kemudian kita bisa menuliskan for dilanjut nama variabel beserta isinya, kemudian batas yang ingin kita tentukan pada variabel tersebut dan apa yang akan dilakukan setelah perintah pada tiap item telah dijalankan. Setelah itu kita bisa menuliskan kode sama seperti cara yang pertama.

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

Bisa dilihat bahwa output yang ada pada cara pertama dan kedua adalah sama yang berarti bahwa kedua cara tersebut berhasil menunjukkan output yang sebenarnya.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ bash for_loop2.sh
1
2
3
4
5
```

### 4. Select Loop

Salah satu metode perulangan salnjutanya adalah select loop. Metode ini digunakan apabila kita ingin membuat sebuah perulangan dengan daftar menu yang bisa dipilih user. Caranya pertama kita bisa membuat file bash terlebih dahulu.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ nano select_loop.sh
```

Setelah itu kita bisa menuliskan select untuk mendefinisikan perulangan tersebut kemudian dilanjut nama variabel menu tersebut dilanjut in (objek objek yang ada pada variabel tersebut). Setelah itu pada baris selanjutnya ditulis do. Pada baris selanjutnya kita bisa menuliskan case nama variabel kemudian in dan objek yang kita inginkan dan pada baris selanjutnya perintah yang dilakukan apabila objek yang ada pada case tersebut terpilih dan diakhiri dengan esac dan done.

```
select minuman in teh kopi air jus susu semua gada
do
  case $minuman in
    teh|kopi|air|semua)
      echo "Maaf, habis"
      ;;
    jus|susu)
      echo "tersedia"
      ;;
    gada)
      break
      ;;
    *) echo "tidak ada didaftar menu"
      ;;
  esac
done
```

Output yang keluar apabila code bash tersebut dijalankan yaitu pertama dia akan mengeluarkan objek objek yang ada pada variabel tersebut dan akan kita pilih. Kemudian kita mengetikkan pilihan kita dan akan mengeluarkan output seperti yang telah kita tuliskan.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ bash select_loop.sh
1) teh
2) kopi
3) air
4) jus
5) susu
6) semua
7) gada
#? teh
tidak ada didaftar menu
#? 7
```

## 5. Until Loop

Pada metode perulangan ini bertujuan untuk mengeksekusi perintah yang kita inginkan sampai kondisi yang telah kita tetapkan terpenuhi. Caranya kita bisa membuat file bash terlebih dahulu.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ nano until_loop.sh
```

Setelah itu kita tulis until untuk mendefinisikan metode tersebut dilanjut dengan kondisi yang kita tetapkan. Pada baris selanjutnya kita tulis do untuk mengeksekusi perintah dan dilanjut pada baris selanjutnya perintah yang kita inginkan dan diakhir dengan done pada baris akhir.

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

Saat kita menjalankan file tersebut, output yang akan dikeluarkan akan mengeluarkan perintah yang kita inginkan.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ bash until_loop.sh
0
2
4
6
8
```

#### Tugas 4

Pada tugas kali ini kita diminta untuk membuat program perulangan yang bisa mencetak bilangan positif berkelipatan ganjil dari bilangan yang kita inputkan saat program dijalankan. Yang harus kita lakukan pertama yaitu kita membuat file bash terlebih dahulu.

```
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/
Tugas4$ nano tugas45.sh
```

Setelah itu kita bisa menuliskan kode nya. Pada kode kali ini saya menggunakan metode perulangan until agar bisa membuat program yang diminta. Disini saya ingin memprint kata kata agar user yang menjalankan program saya bisa mengerti untuk memasukkan bilangan. Setelah itu untuk menyimpan input user saya menggunakan command read pada baris selanjutnya saya memulai metode perulangannya dengan menuliskan until untuk mendefinisikan metode perulangan tersebut dilanjut kondisi yang kita tetapkan. Dilanjut menulis do pada baris selanjutnya dan setelah itu karena saya ingin output yang keluar hanya bilangan ganjil maka saya mensortir bilangan tersebut menggunakan metode percabangan if else. Apabila angka yang diinputkan menghasilkan sisa 0 apabila dibagi 2 maka angka tersebut adalah genap dan harus dikurangi satu baru bisa diprintkan. Apabila sebaliknya maka angka tersebut adalah ganjil dan bisa langsung diprint. Setelah itu bisa diakhiri dengan fi untuk mengakhiri program if else dan done untuk mengakhiri program loop.

```

GNU nano 6.2      tugas45.sh *
printf "halo masukkan bilangan"
read input
until [ ! $input -gt 0 ]
do
    let b=$input%2
    if [ $b == 0 ]
    then
        input=$((input - 1))
    else
        echo $input
        input=$((input - 1))
    fi
done

```

Setelah program dijalankan maka akan mengeluarkan output seperti dibawah ini.

```

zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/
Tugas4$ bash tugas45.sh
halo masukkan bilangan8
7
5
3
1

```

NB :

Izin tidak mengup lwt git mba soalnya ini daritdi gabisa terus

```

zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$ git push -u origin mas
ter
Username for 'https://github.com': FathimaZahrah28
Password for 'https://FathimaZahrah28@github.com':
To https://github.com/FathimaZahrah28/21083010043.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/FathimaZahrah28/210830100
43.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
zahra@zahra-VirtualBox:~/Documents/Sistem_Operasi/Tugas4$

```