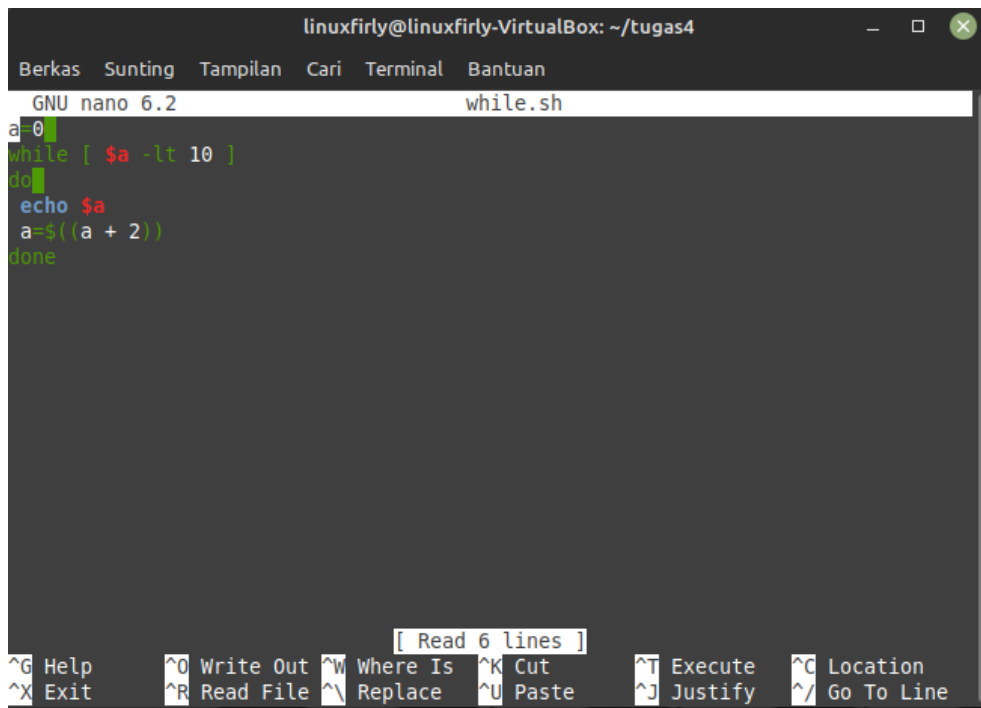


LAPORAN TUGAS 4

WHILE

Perulangan while adalah pernyataan yang mengulangi blok kode sampai kondisi yang ditentukan dievaluasi menjadi benar. Jika kondisinya salah maka perintah di dalam blok while akan dieksekusi dan diulang kembali setelah memeriksa kondisinya. Juga jika kondisinya benar, pernyataan di dalam blok while akan dilewati dan pernyataan setelah blok while dieksekusi.

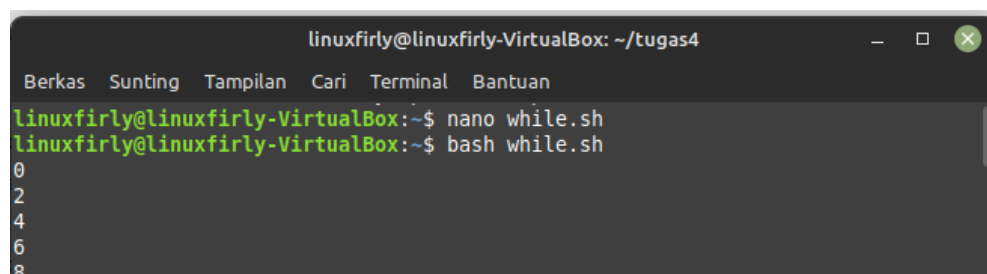


```
linuxfirly@linuxfirly-VirtualBox: ~/tugas4
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 6.2                                while.sh
a=0
while [ $a -lt 10 ]
do
  echo $a
  a=$((a + 2))
done

[ Read 6 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Pernyataan while pada gambar diatas memberikan perintah untuk melewati ke awal blok dan memulai eksekusi perintah dengan melewati perintah berikutnya di dalam blok.

Variabel a yang kurang dari (-lt) 10, sehingga kondisi do memeriksa setiap variabel a ditambah 2, akan diulangi lagi menggunakan do dan menaikkan nilai variabel a.

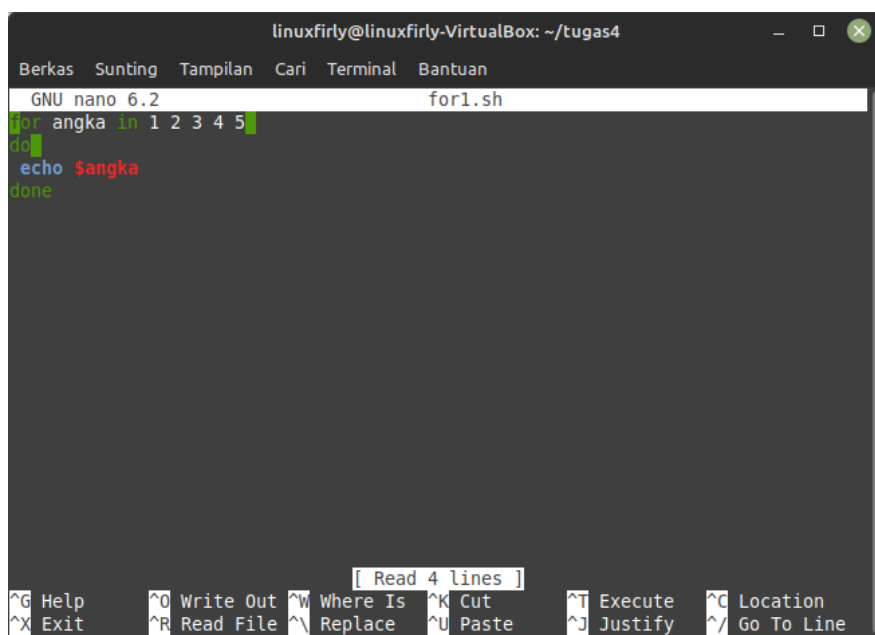


```
linuxfirly@linuxfirly-VirtualBox: ~/tugas4
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
linuxfirly@linuxfirly-VirtualBox:~$ nano while.sh
linuxfirly@linuxfirly-VirtualBox:~$ bash while.sh
0
2
4
6
8
```

FOR LOOP

Bash for loop merupakan statement iterasi, yang berarti kode tersebut dapat dijalankan berulang-ulang.

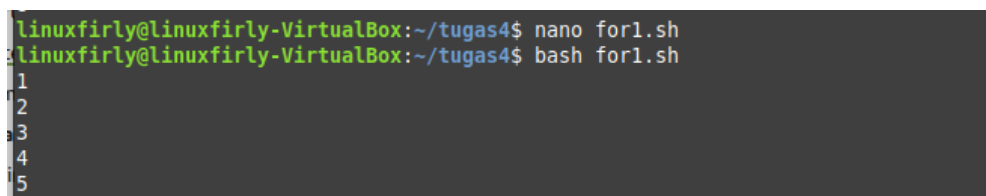
- angka adalah pengganti untuk variabel. Sementara itu, \$angka adalah value individual dari variable.
- In digunakan untuk memisahkan variabel dan item yang mengikuti.
- 1 2 3 4 5 merupakan contoh item yang intruksinta ingin diikuti
- Do merupakan keyword yang menggantikan loop. Elemen ini akan menjalankan intruksi sebanyak n kali, yang mana n merupakan jumlah total itemnya.
- echo \$angka adalah kode yang akan diulang sebanyak n kali.
- Done untuk menghentikan loop



The screenshot shows a terminal window with the nano text editor open. The title bar indicates the user is 'linuxfirly' on a 'linuxfirly-VirtualBox' machine, working in the directory '~/tugas4'. The editor is editing a file named 'for1.sh'. The script content is as follows:

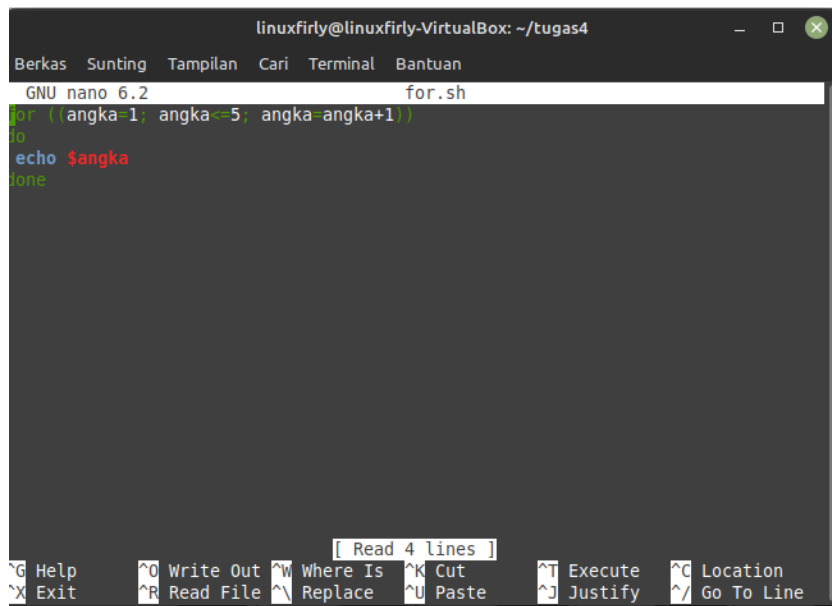
```
GNU nano 6.2 for1.sh
for angka in 1 2 3 4 5
do
  echo $angka
done
```

The bottom status bar of the nano editor shows various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^_ Replace, ^U Paste, ^J Justify, and ^_ Go To Line. A message '[Read 4 lines]' is also visible.



The screenshot shows a terminal window where the script 'for1.sh' has been executed. The prompt is 'linuxfirly@linuxfirly-VirtualBox:~/tugas4\$'. The user has entered 'nano for1.sh' and then 'bash for1.sh'. The output of the script is a list of numbers from 1 to 5, each on a new line:

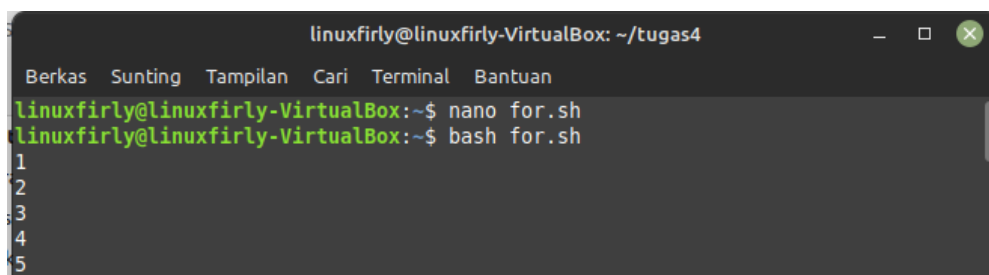
```
linuxfirly@linuxfirly-VirtualBox:~/tugas4$ nano for1.sh
linuxfirly@linuxfirly-VirtualBox:~/tugas4$ bash for1.sh
1
2
3
4
5
```

A screenshot of a nano editor window titled 'linuxfirly@linuxfirly-VirtualBox: ~/tugas4'. The window shows a file named 'for.sh' with the following content:

```
GNU nano 6.2 for.sh
for ((angka=1; angka<=5; angka=angka+1))
do
echo $angka
done
```

 The status bar at the bottom indicates '[Read 4 Lines]' and lists various keyboard shortcuts like ^G Help, ^O Write Out, etc.

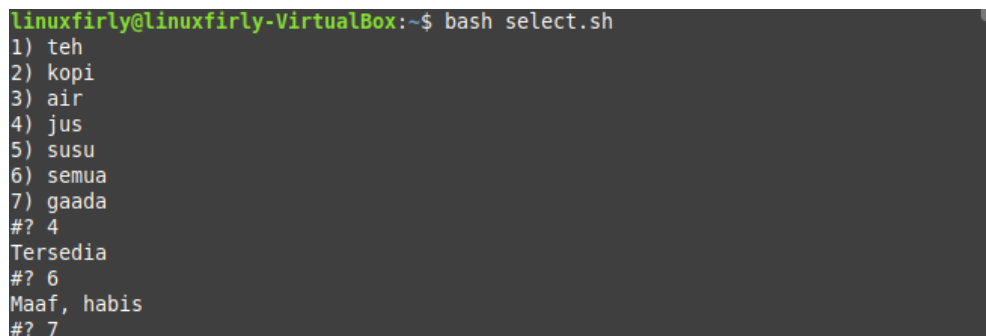
Kode di atas menunjukkan bahwa value awalnya adalah 1. Loop akan dieksekusi selama kondisi EXP2 benar, yang berarti valuenya tidak boleh lebih besar daripada 5. Selain itu, tanda +1 menunjukkan bahwa increment-nya adalah 1.

A screenshot of a terminal window titled 'linuxfirly@linuxfirly-VirtualBox: ~/tugas4'. It shows the user running 'nano for.sh' and then 'bash for.sh'. The output of the script is a list of numbers from 1 to 5, each on a new line:

```
linuxfirly@linuxfirly-VirtualBox:~$ nano for.sh
linuxfirly@linuxfirly-VirtualBox:~$ bash for.sh
1
2
3
4
5
```

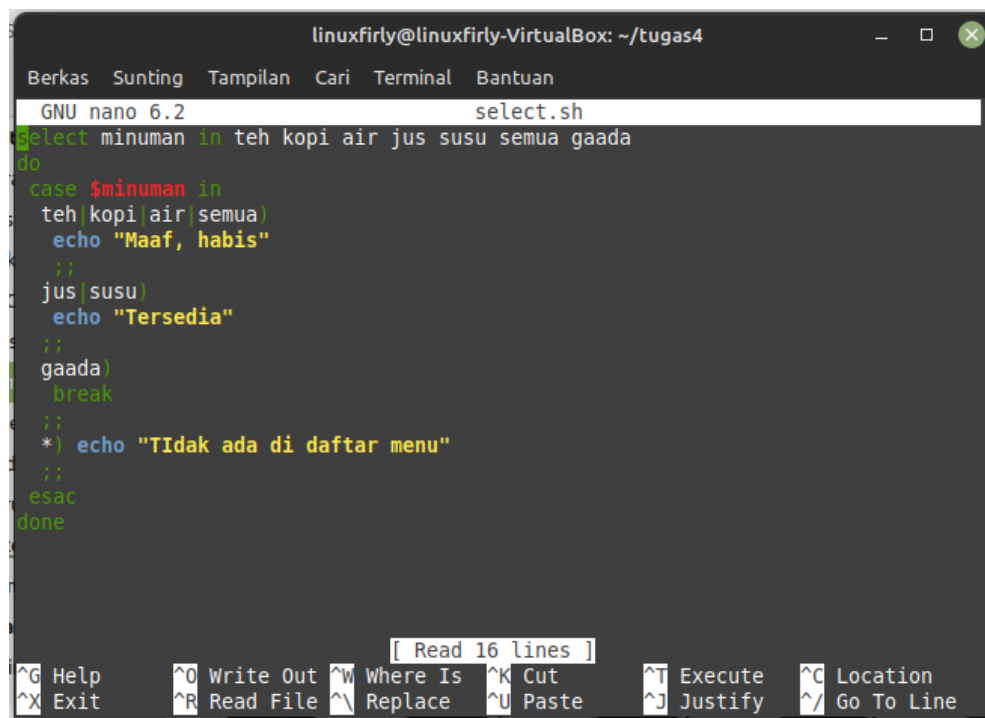
SELECT LOOP

Select-loop di shell dapat dihentikan dalam dua kasus hanya jika ada pernyataan break atau interupsi keyboard. Tujuan utama menggunakan loop pilih adalah bahwa itu mewakili elemen data yang berbeda dalam bentuk daftar bernomor kepada pengguna. Pengguna dapat dengan mudah memilih salah satu opsi seperti yang terdaftar oleh program.

A screenshot of a terminal window titled 'linuxfirly@linuxfirly-VirtualBox:~\$'. It shows the user running 'bash select.sh'. The output of the script is a list of options:

```
linuxfirly@linuxfirly-VirtualBox:~$ bash select.sh
1) teh
2) kopi
3) air
4) jus
5) susu
6) semua
7) gaada
#? 4
Tersedia
#? 6
Maaf, habis
#? 7
```

Dalam program di bawah ini membuat menu bernomor untuk memungkinkan user untuk memilih minuman yang tersedia.

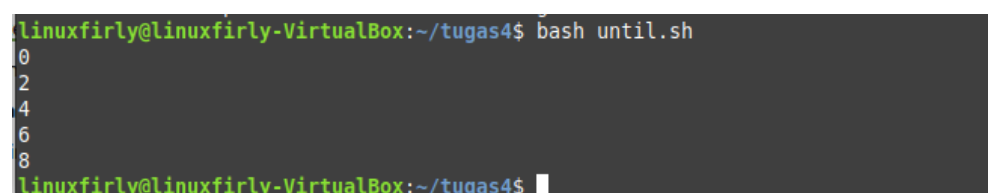


```
linuxfiry@linuxfiry-VirtualBox: ~/tugas4
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 6.2      select.sh
select minuman in teh kopi air jus susu semua gaada
do
case $minuman in
teh|kopi|air|semua)
    echo "Maaf, habis"
    ;;
jus|susu)
    echo "Tersedia"
    ;;
gaada)
    break
    ;;
*) echo "Tidak ada di daftar menu"
    ;;
esac
done

[ Read 16 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

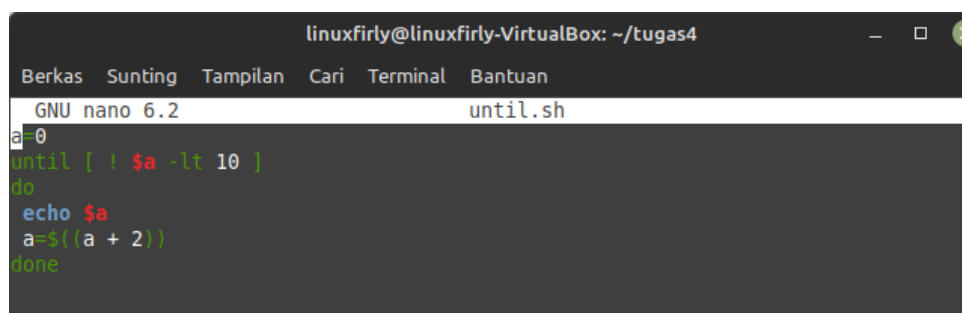
UNTIL

Loop until digunakan untuk mengeksekusi serangkaian perintah tertentu selama kondisi yang diberikan bernilai false. Kondisi dievaluasi sebelum menjalankan perintah. Jika kondisi bernilai false, perintah dijalankan. Jika tidak, jika kondisi bernilai true, loop akan dihentikan dan kontrol program akan diteruskan ke perintah.



```
linuxfiry@linuxfiry-VirtualBox:~/tugas4$ bash until.sh
0
2
4
6
8
linuxfiry@linuxfiry-VirtualBox:~/tugas4$
```

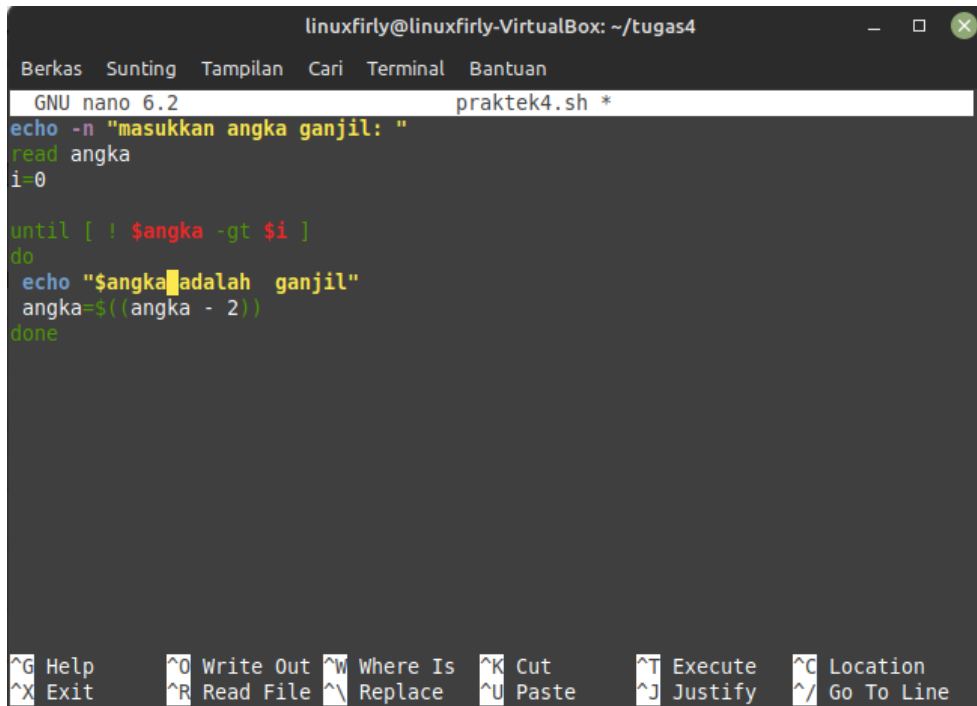
Pada program dibawah menunjukkan bahwa variabel ditambah 2 yang memiliki batas angka 10.



```
linuxfiry@linuxfiry-VirtualBox: ~/tugas4
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 6.2      until.sh
a=0
until [ ! $a -lt 10 ]
do
    echo $a
    a=$((a + 2))
done
```

TUGAS 4

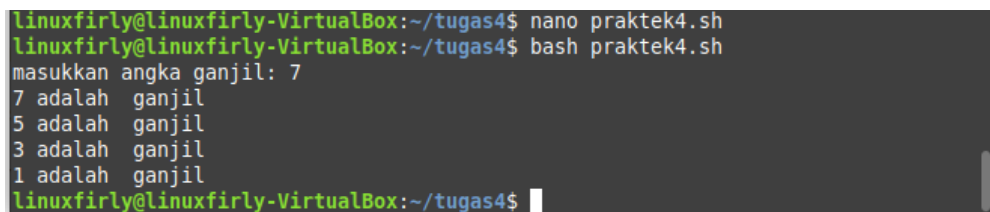
- Menggunakan metode until
- Menggunakan perintah echo untuk memasukkan angka ganjil
- Angka -gt | menunjukkan eksekusi angka lebih besar dari i
- Pada baris ke 7 membaca input dan akan diurutkan dari yang terbesar dengan selisih 2 angka.



```
linuxfirly@linuxfirly-VirtualBox: ~/tugas4
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 6.2  praktek4.sh *
echo -n "masukkan angka ganjil: "
read angka
i=0

until [ ! $angka -gt $i ]
do
    echo "$angka adalah ganjil"
    angka=$((angka - 2))
done

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify  ^/ Go To Line
```



```
linuxfirly@linuxfirly-VirtualBox:~/tugas4$ nano praktek4.sh
linuxfirly@linuxfirly-VirtualBox:~/tugas4$ bash praktek4.sh
masukkan angka ganjil: 7
7 adalah ganjil
5 adalah ganjil
3 adalah ganjil
1 adalah ganjil
linuxfirly@linuxfirly-VirtualBox:~/tugas4$
```