

MODUL PRAKTIKUM PERTEMUAN 5
SISTEM OPERASI
IF-2231

PEMROGRAMAN SHELL II



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
2019

Tujuan dan Pokok Bahasan

Setelah mempelajari materi yang ada pada modul praktikum ini, praktikan diharapkan mampu:

1. Mempelajari struktur dasar perulangan
2. Mempelajari perulangan until, for, select
3. Membuat aplikasi perulangan dengan shell menggunakan konstruksi until, for, select
4. Membuat fungsi dan mengetahui cara memanggil fungsi tersebut
5. Mempelajari dasar array pada shell

Teori Dasar

1. Perulangan/Looping

Perulangan atau dalam istilah lain disebut dengan loop. Perulangan dipakai ketika kita dihadapkan pada suatu masalah dalam jumlah besar yang membutuhkan penyelesaian terkadang sama dengan pola yang telah kita ketahui. Contoh perulangan di kehidupan nyata. Tuliskan "Ini adalah Perulangan" sebanyak 100 kali.

Kita lihat ada 100 masalah dengan 100 penyelesaian sama dengan pola yang sama, disinilah perulangan sangat berguna, mengefisiensikan pekerjaan kita.

2. Fungsi

Fungsi / Function adalah satu blok kode yang melakukan tugas tertentu atau satu blok instruksi yang dieksekusi ketika dipanggil dari bagian lain dalam suatu program.

Tujuan pembuatan fungsi adalah :

1. Memudahkan dalam pembuatan program.
2. Menghemat ukuran program.

Keuntungan memakai fungsi :

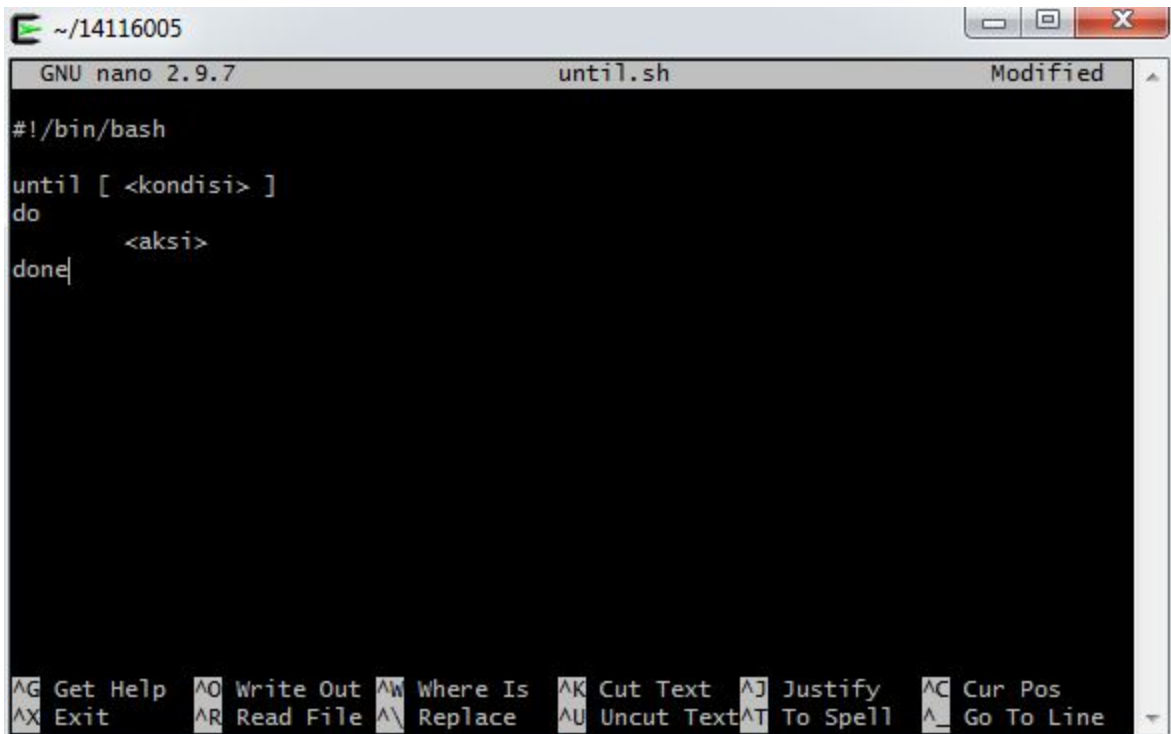
1. Menguraikan tugas pemrograman rumit menjadi langkah-langkah yang lebih sederhana atau kecil.
2. Mengurangi duplikasi kode (kode yang sama ditulis berulang-ulang) dalam program.
3. Dapat menggunakan kode yang ditulis dalam berbagai program yang berbeda.
4. Memecah program besar menjadi kecil sehingga dapat dikerjakan oleh programmer-programmer atau dipecah menjadi beberapa tahap sehingga mempermudah pengerjaan dalam sebuah proyek
5. Menyembunyikan informasi dari user sehingga mencegah adanya perbuatan iseng seperti memodifikasi atau mengubah program yang kita buat
6. Meningkatkan kemampuan pelacakan kesalahan, jika terjadi suatu kesalahan kita tinggal mencari fungsi yang bersangkutan saja dan tak perlu mencari kesalahan tersebut di seluruh program.

Bentuk umum Sebuah fungsi adalah sebagai berikut :

```
Tipe data nama_fungsi(daftar_parameter)
```

Percobaan

3. Struktur Dasar Until



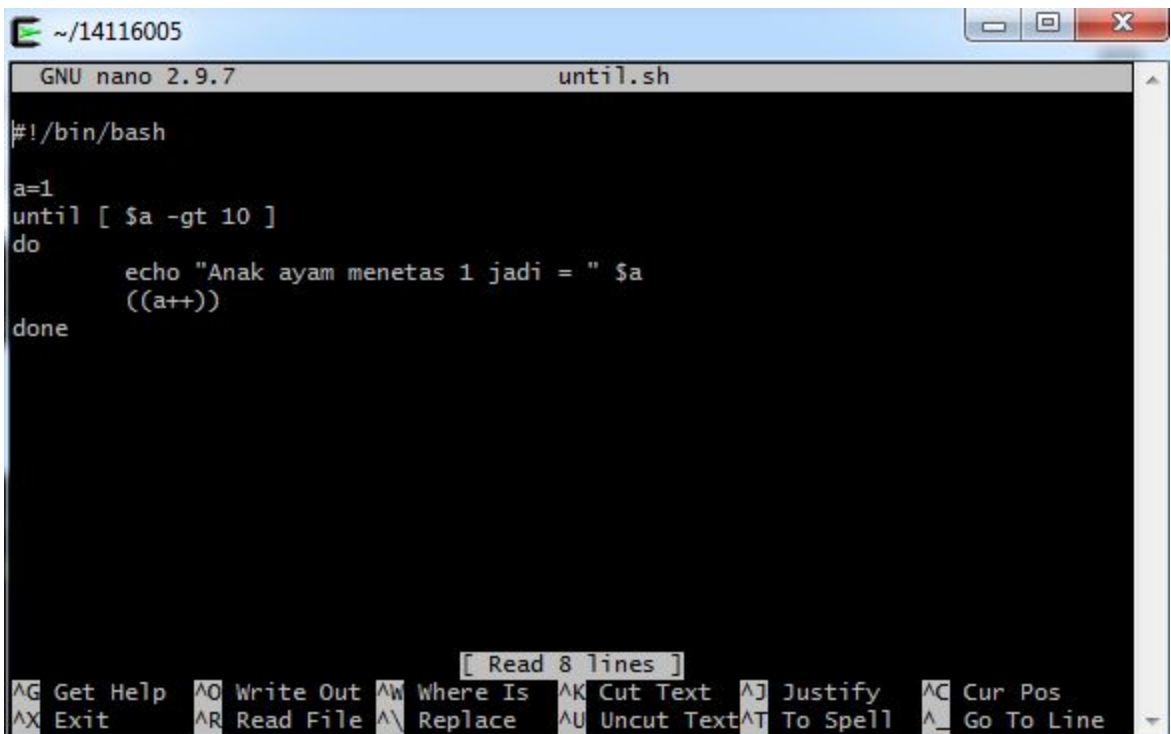
The screenshot shows a nano editor window titled "GNU nano 2.9.7" editing a file named "until.sh". The window title bar also shows the user's home directory "~ / 14116005" and standard window controls. The code inside the editor is a basic until loop structure:

```
#!/bin/bash

until [ <kondisi> ]
do
    <aksi>
done
```

At the bottom of the window, there is a status bar with various keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^\ Replace, ^U Uncut Text, ^T To Spell, and ^_ Go To Line.

Contoh program :



The screenshot shows the same nano editor window with the "until.sh" file. The code now includes a variable 'a' and a loop that prints a message until 'a' is greater than 10:

```
#!/bin/bash

a=1
until [ $a -gt 10 ]
do
    echo "Anak ayam menetas 1 jadi = " $a
    ((a++))
done
```

The status bar at the bottom is the same as in the previous screenshot, but it also includes a "[Read 8 lines]" indicator above the shortcuts.

```
~/14116005

Dicky Hermawan@DickyHermawan ~/14116005
$ ./until.sh
Anak ayam menetas 1 jadi = 1
Anak ayam menetas 1 jadi = 2
Anak ayam menetas 1 jadi = 3
Anak ayam menetas 1 jadi = 4
Anak ayam menetas 1 jadi = 5
Anak ayam menetas 1 jadi = 6
Anak ayam menetas 1 jadi = 7
Anak ayam menetas 1 jadi = 8
Anak ayam menetas 1 jadi = 9
Anak ayam menetas 1 jadi = 10

Dicky Hermawan@DickyHermawan ~/14116005
$ |
```

4. Struktur Dasar For

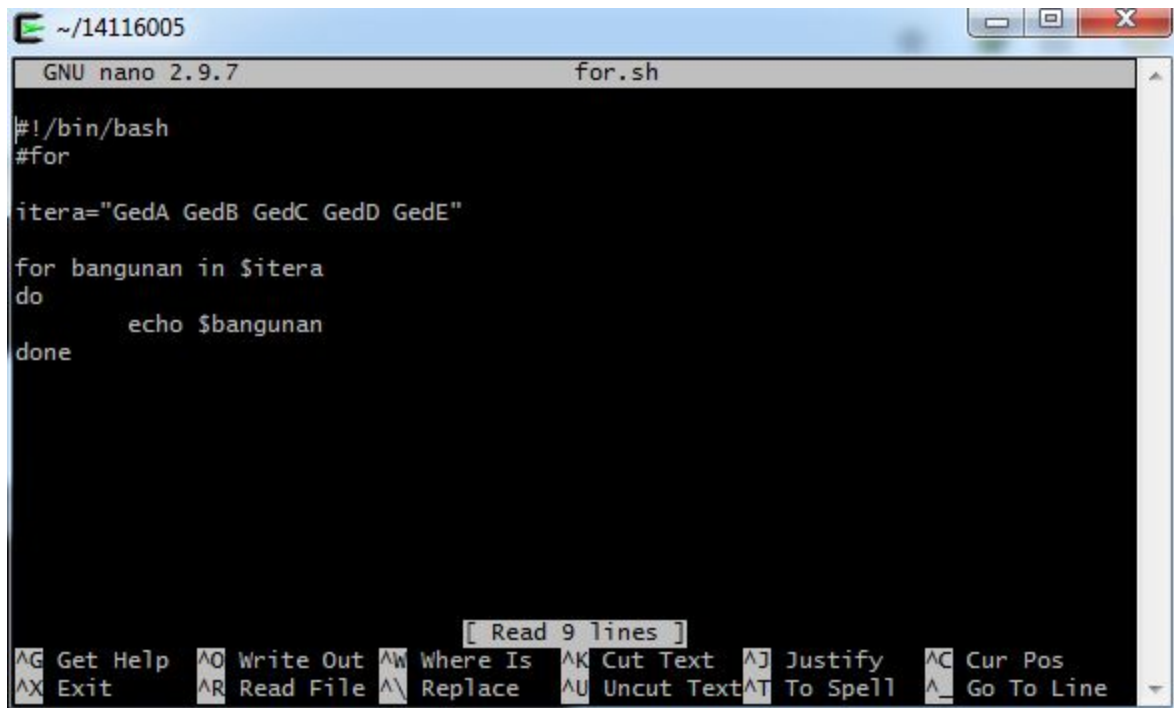
```
~/14116005
GNU nano 2.9.7 for.sh Modified

#!/bin/bash
#for

for var in <list>
do
    <aksi>
done

[ Cancelled ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Contoh program :



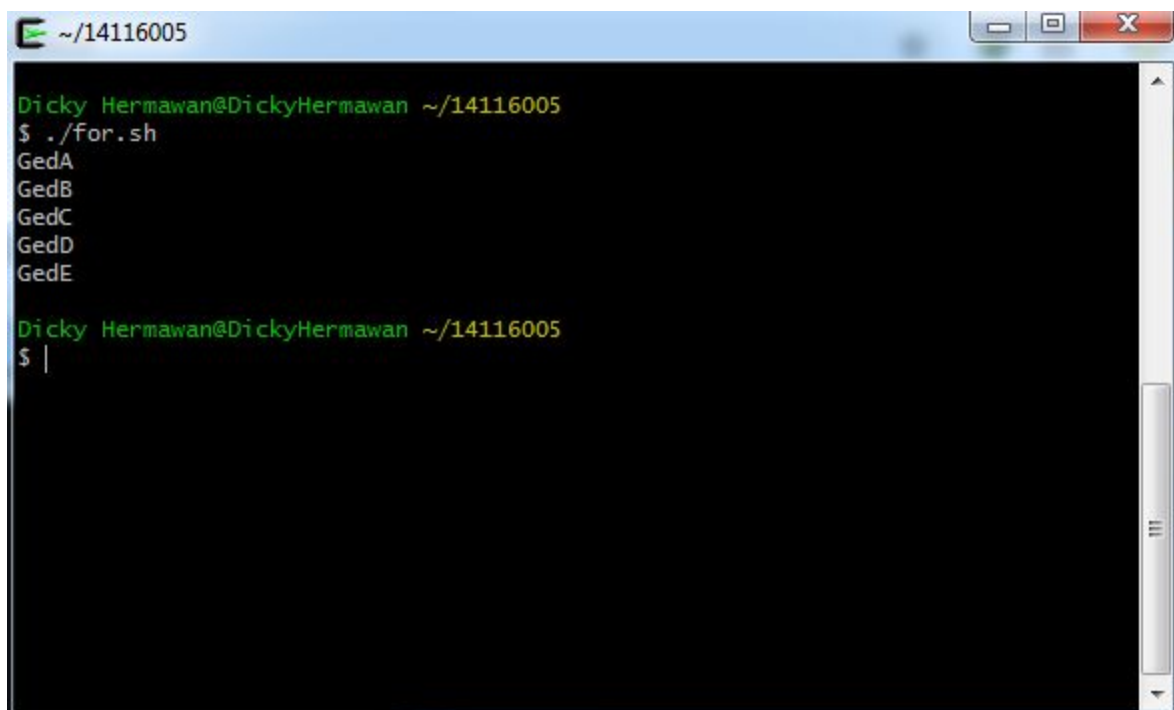
The screenshot shows a terminal window with the nano 2.9.7 editor open. The file being edited is named 'for.sh' and is located at the user's home directory (~). The script content is as follows:

```
#!/bin/bash
#for

itera="GedA GedB GedC GedD GedE"

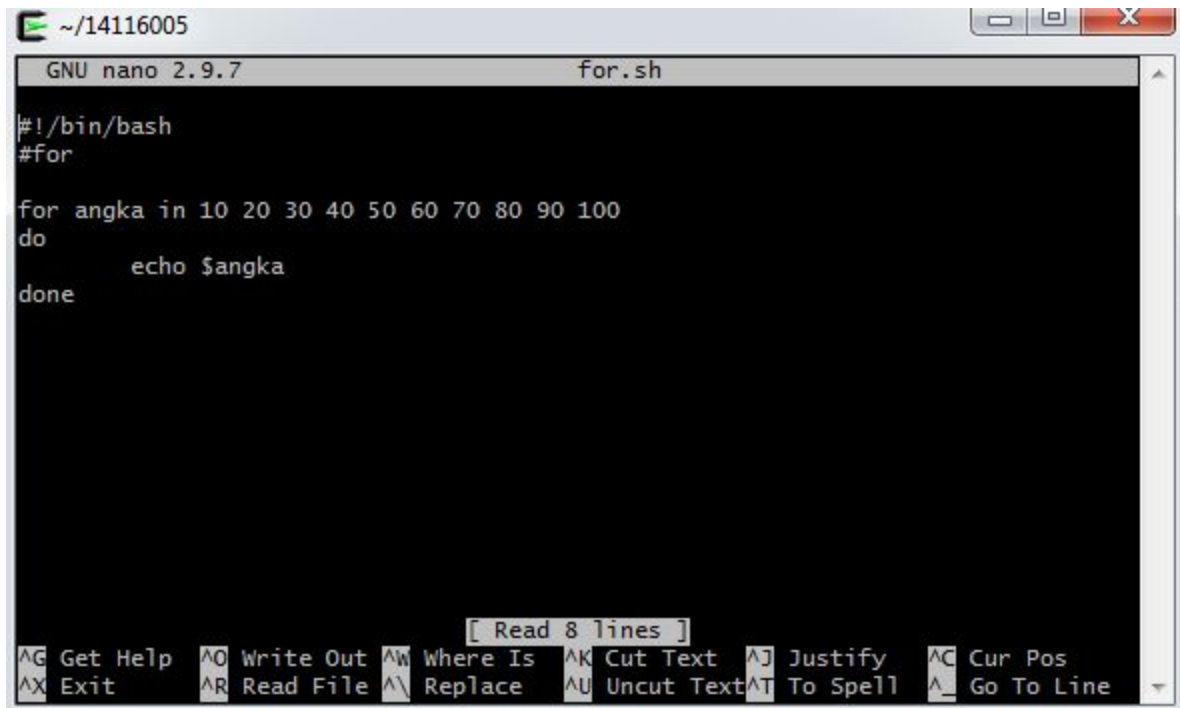
for bangunan in $itera
do
    echo $bangunan
done
```

At the bottom of the editor, a status bar indicates '[Read 9 lines]' and provides a list of keyboard shortcuts for various editor functions.



The screenshot shows a terminal window where the script 'for.sh' has been executed. The prompt is 'Dicky Hermawan@DickyHermawan ~/14116005'. The command entered is './for.sh'. The output of the script is a list of five lines: 'GedA', 'GedB', 'GedC', 'GedD', and 'GedE'. The prompt returns to '\$ |'.

Contoh program menggunakan angka :

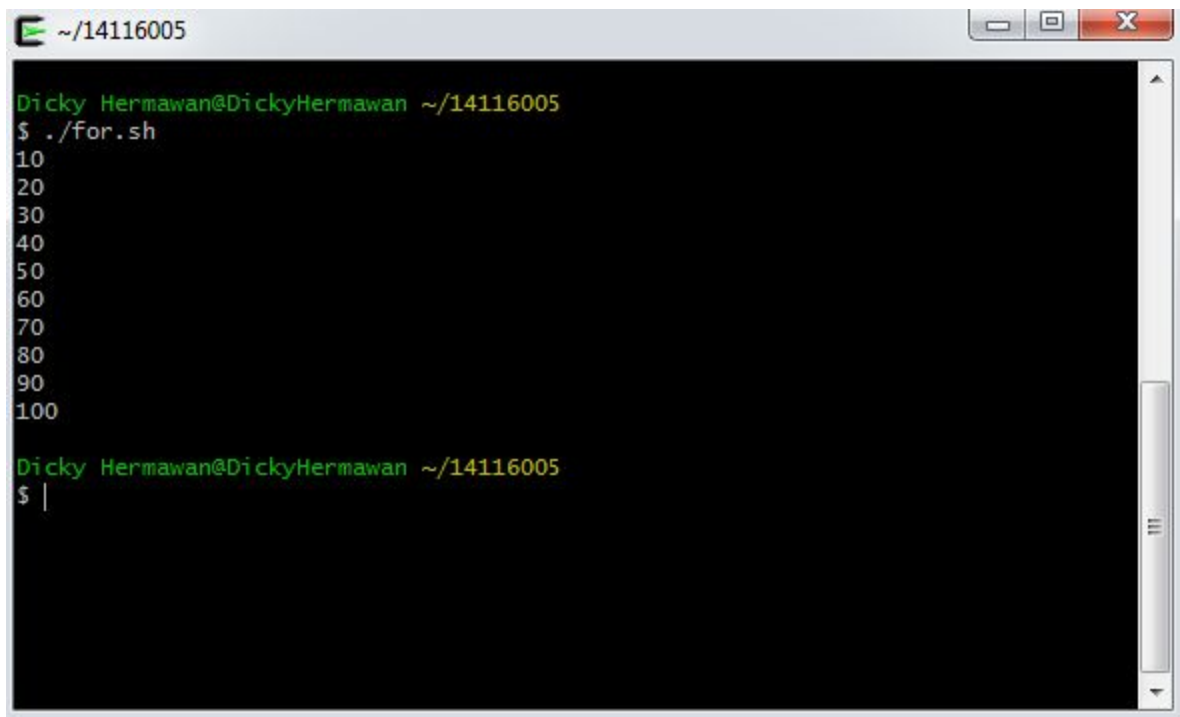


The screenshot shows a terminal window with the nano 2.9.7 text editor. The file being edited is named `for.sh`. The script content is as follows:

```
#!/bin/bash
#for

for angka in 10 20 30 40 50 60 70 80 90 100
do
    echo $angka
done
```

At the bottom of the editor, a status bar displays various keyboard shortcuts. A tooltip `[Read 8 lines]` is visible over the `^R` (Read File) shortcut.

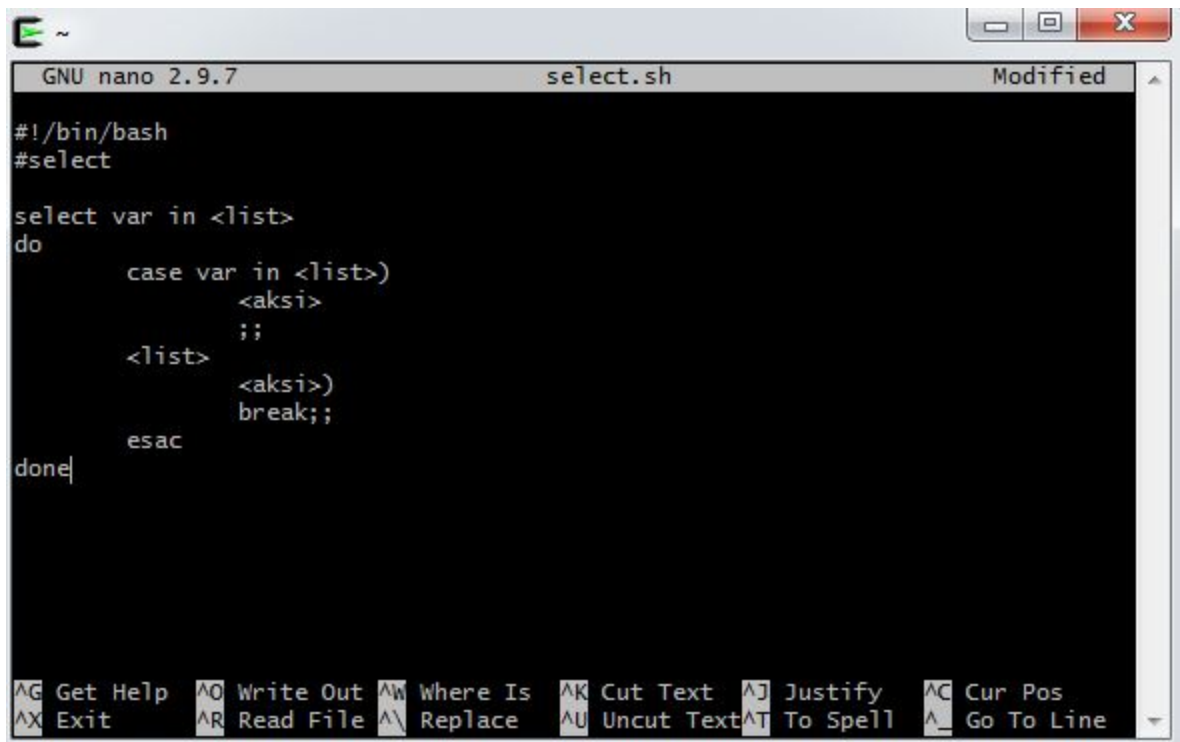


The screenshot shows a terminal window where the script `for.sh` has been executed. The prompt is `Dicky Hermawan@DickyHermawan ~/14116005`. The command `$./for.sh` has been entered, and the output displays the numbers 10 through 100, each on a new line.

```
Dicky Hermawan@DickyHermawan ~/14116005
$ ./for.sh
10
20
30
40
50
60
70
80
90
100

Dicky Hermawan@DickyHermawan ~/14116005
$ |
```

5. Struktur Dasar Select



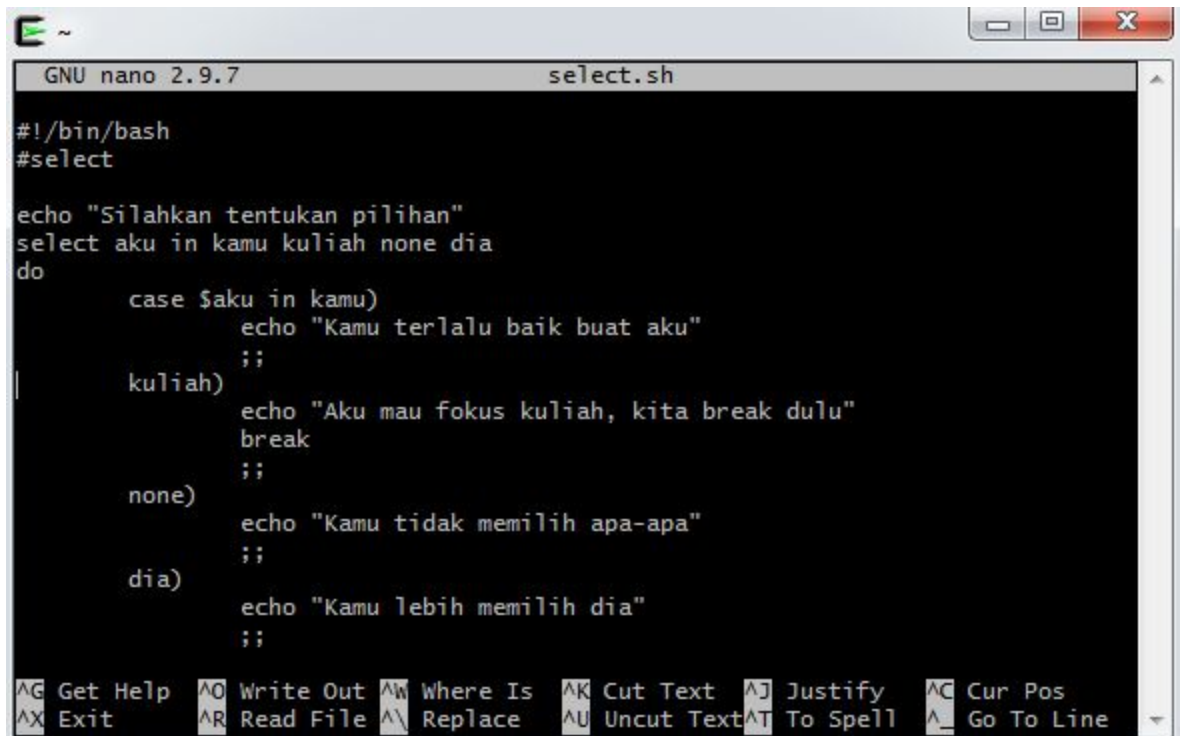
```
GNU nano 2.9.7      select.sh      Modified

#!/bin/bash
#select

select var in <list>
do
    case var in <list>)
        <aksi>
        ;;
    <list>
        <aksi>)
        break;;
    esac
done
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Contoh program :



```
GNU nano 2.9.7      select.sh

#!/bin/bash
#select

echo "Silahkan tentukan pilihan"
select aku in kamu kuliah none dia
do
    case $aku in kamu)
        echo "Kamu terlalu baik buat aku"
        ;;
    kuliah)
        echo "Aku mau fokus kuliah, kita break dulu"
        break
        ;;
    none)
        echo "Kamu tidak memilih apa-apa"
        ;;
    dia)
        echo "Kamu lebih memilih dia"
        ;;
done
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line


```
Dicky Hermawan@DickyHermawan ~
$ ./select.sh
Silahkan tentukan pilihan
1) kamu
2) kuliah
3) none
4) dia
#? 3
Kamu tidak memilih apa-apa
#? 2
Aku mau fokus kuliah, kita break dulu

Dicky Hermawan@DickyHermawan ~
$ |
```

Contoh lain

```
GNU nano 2.9.7 select.sh

#!/bin/bash
#select

karakter='Dudung Adit Sopo Jarwo Denis PakHaji'
echo "Pilih karakter : "
select nama in $karakter
do
    if [ $nama == 'Jarwo' ]
    then
        echo "Bang Jarwo makin deket"
        break
    fi
    echo "Hallo $nama"
done

[ Read 14 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

```
Dicky Hermawan@DickyHermawan ~
$ ./select.sh
Pilih karakter :
1) Dudung
2) Adit
3) Sopo
4) Jarwo
5) Denis
6) PakHaji
#? 2
Hallo Adit
#? 1
Hallo Dudung
#? 6
Hallo PakHaji
#? 4
Bang Jarwo makin dekat

Dicky Hermawan@DickyHermawan ~
$ |
```

6. Fungsi

Contoh program :

```
GNU nano 2.9.7 fungsi.sh

#!/bin/bash

hallo(){
    echo "Hello World"
}

hallo

[ Read 7 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

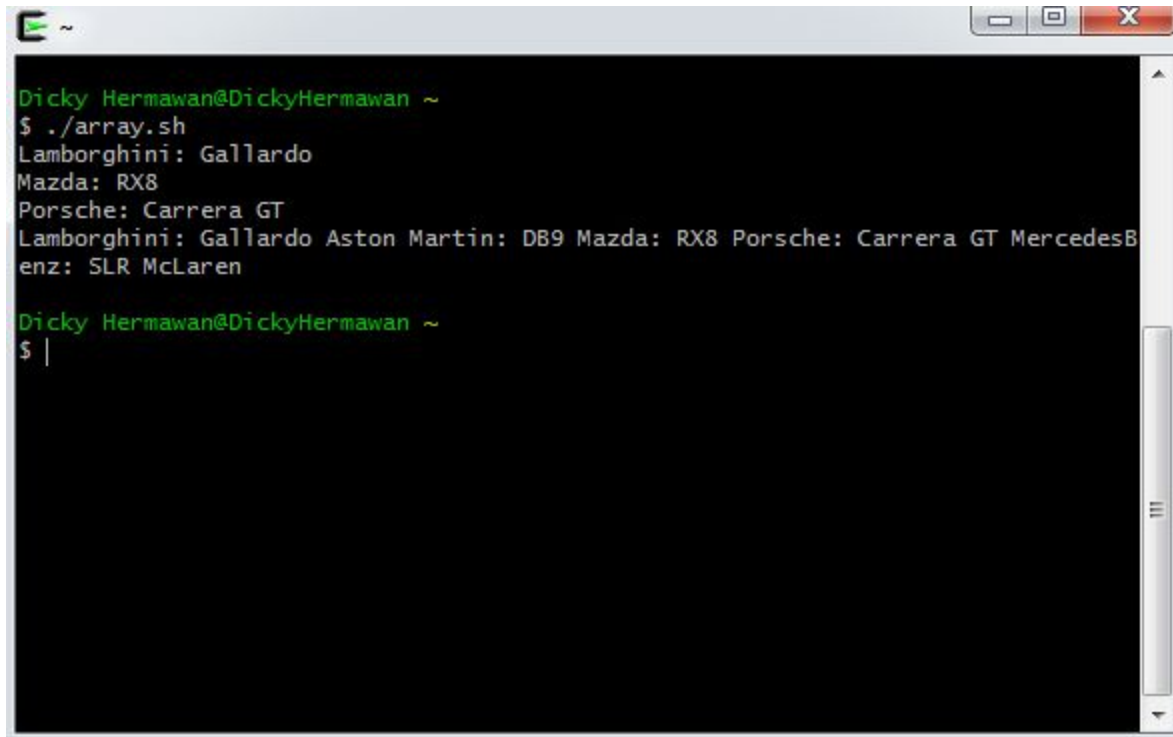
```
Dicky Hermawan@DickyHermawan ~  
$ ./fungsi.sh  
Hello World  
  
Dicky Hermawan@DickyHermawan ~  
$ |
```

7. Array

Array adalah kumpulan variabel dengan tipe sejenis

Contoh program :

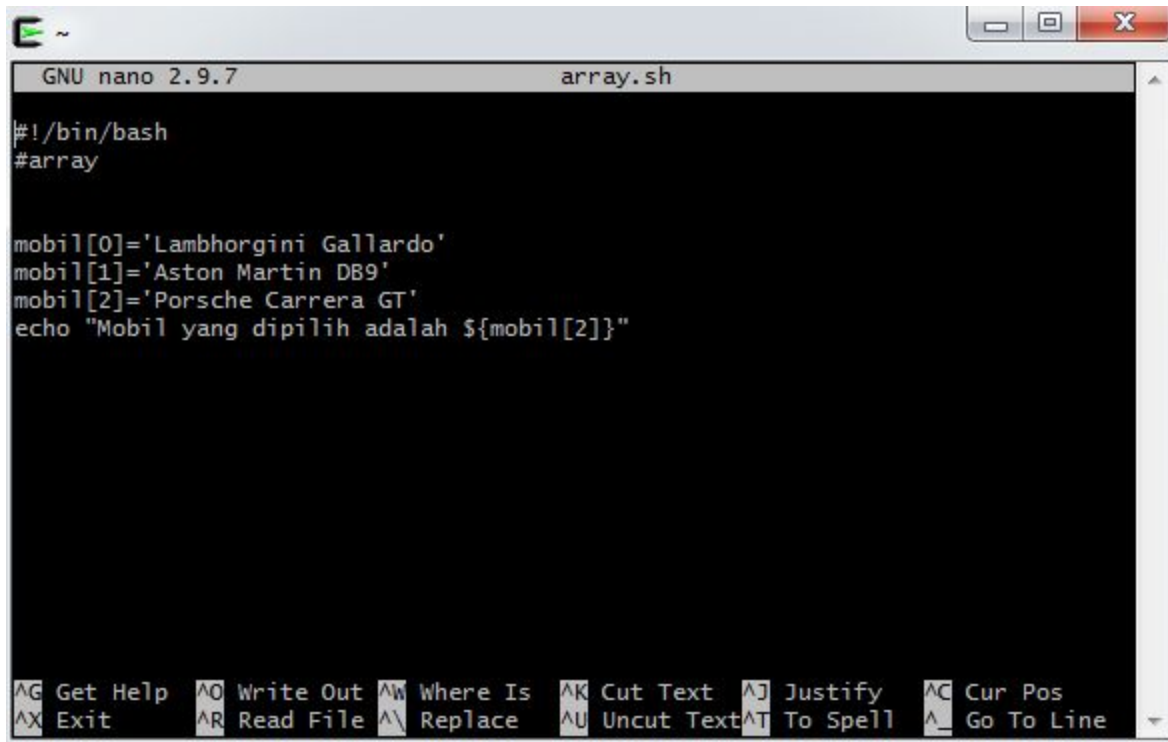
```
GNU nano 2.9.7 array.sh  
#!/bin/bash  
#array  
  
mobil=('Lamborghini: Gallardo' 'Aston Martin: DB9' 'Mazda: RX8'  
      'Porsche: Carrera GT' 'MercedesBenz: SLR McLaren')  
  
echo ${mobil[0]}  
echo ${mobil[2]}  
echo ${mobil[3]}  
echo ${mobil[*]}  
  
[ Read 11 lines ]  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^_ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

A terminal window with a black background and green text. The prompt is 'Dicky Hermawan@DickyHermawan ~'. The user enters './array.sh'. The script outputs four lines: 'Lamborghini: Gallardo', 'Mazda: RX8', 'Porsche: Carrera GT', and 'Lamborghini: Gallardo Aston Martin: DB9 Mazda: RX8 Porsche: Carrera GT MercedesBenz: SLR McLaren'. The prompt returns, and the user enters a vertical bar '|'.

```
Dicky Hermawan@DickyHermawan ~  
$ ./array.sh  
Lamborghini: Gallardo  
Mazda: RX8  
Porsche: Carrera GT  
Lamborghini: Gallardo Aston Martin: DB9 Mazda: RX8 Porsche: Carrera GT MercedesBenz: SLR McLaren  
  
Dicky Hermawan@DickyHermawan ~  
$ |
```

Echo \${mobil[0]} memanggil indeks array ke-0
Echo \${mobil[2]} memanggil indeks array ke-2
Echo \${mobil[3]} memanggil indeks array ke-3
Echo \${mobil[*]} memanggil seluruh indeks array

Contoh lain :

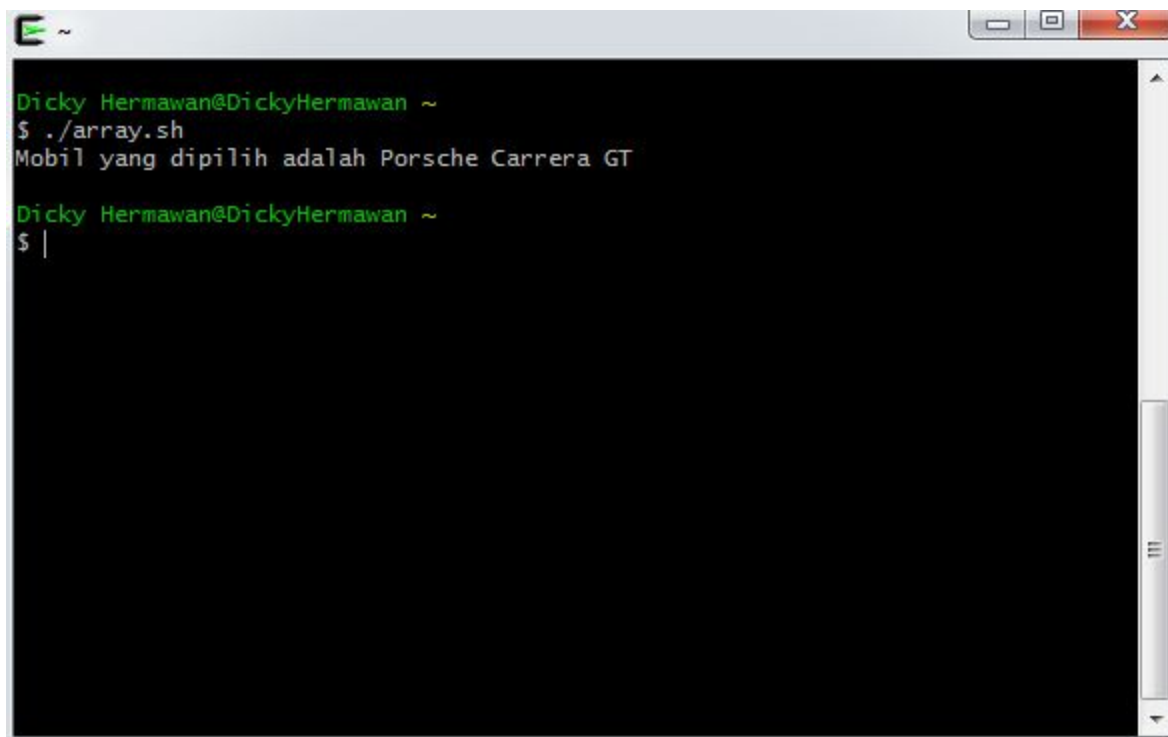


```
GNU nano 2.9.7 array.sh

#!/bin/bash
#array

mobil[0]='Lamborghini Gallardo'
mobil[1]='Aston Martin DB9'
mobil[2]='Porsche Carrera GT'
echo "Mobil yang dipilih adalah ${mobil[2]}"
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line



```
Dicky Hermawan@DickyHermawan ~
$ ./array.sh
Mobil yang dipilih adalah Porsche Carrera GT

Dicky Hermawan@DickyHermawan ~
$ |
```

TUGAS

1. Buatlah sebuah program menerima inputan sebanyak N tentukan :
 - a. Maksimum
 - b. Minimum

INPUT	OUTPUT
5 1 3 2 -1 5	Maks : 5 Min : -1

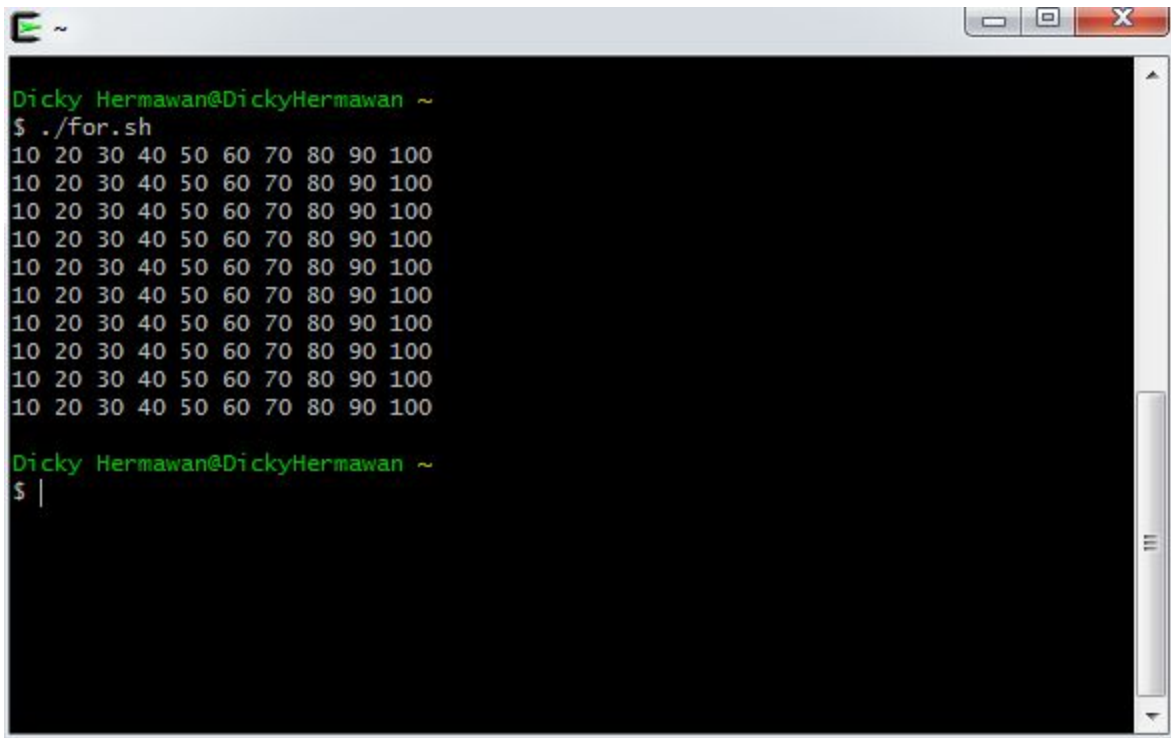
2. Jika pak mimin mempunyai anak sebanyak A dan permen sebanyak B, jika setiap anak harus mendapatkan C permen. Tentukan, apakah setiap anak mendapatkan permen? Outputkan Ya / Tidak

INPUT	OUTPUT
5 10 2 5 5 2	Ya Tidak

3. Perusahaan Kapitalis mempunyai karyawan bernama mawan, di Perusahaan Kapitalis mempunyai aturan yaitu, setiap karyawan harus bekerja selama 12 jam setiap harinya, dan dikarenakan Perusahaan Kapitalis memiliki bos yang pelit, sehingga karyawan mendapatkan gaji Rp.50.000/hari, jika karyawan telat setiap jamnya gaji dikurangi Rp.5.000. Jika lembur setiap jamnya mendapatkan gaji Rp.500. Tentukan gaji yang diterima mawan hari ini.

INPUT	OUTPUT
12 1 13	50000 -5000 50500

4. Buatlah program yang outputnya seperti ini

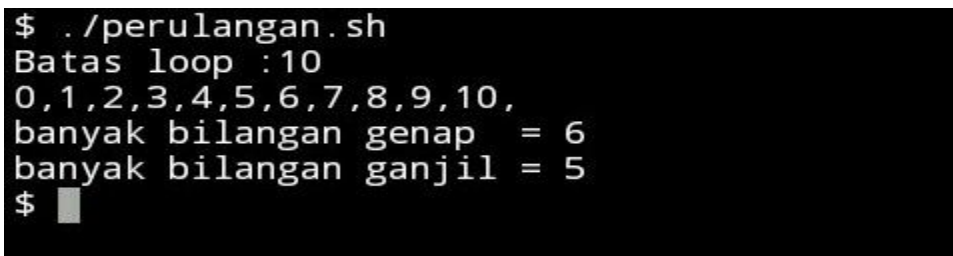


```
Dicky Hermawan@DickyHermawan ~  
$ ./for.sh  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
10 20 30 40 50 60 70 80 90 100  
Dicky Hermawan@DickyHermawan ~  
$ |
```

Gunakan perulangan for.

5. Buatlah pemrograman shell yang menerima inputan berupa angka dan melakukan keluaran urutan banyaknya angka tersebut dan jumlah bilangan ganjil serta banyaknya bilangan genap yang terdapat pada angka yang diinputkan sebelumnya, menggunakan perulangan serta percabangan.

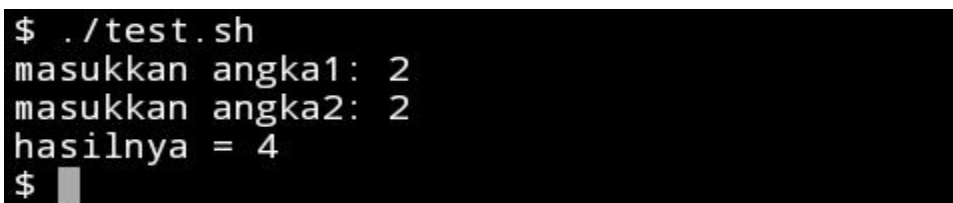
Contoh hasil program:



```
$ ./perulangan.sh  
Batas loop :10  
0,1,2,3,4,5,6,7,8,9,10,  
banyak bilangan genap = 6  
banyak bilangan ganjil = 5  
$ █
```

6. Buatlah suatu fungsi untuk menghitung jumlah 2 buah bilangan yang diinputkan oleh user dan hasilnya akan ditampilkan di layar.

Contoh hasil program:



```
$ ./test.sh  
masukkan angka1: 2  
masukkan angka2: 2  
hasilnya = 4  
$ █
```

Referensi

1. <https://musaamin.web.id/bash-script-perulangan-while-until-for-select/>
2. <https://itconsultantproject.wordpress.com/2018/04/18/perulangan-pada-pemrograman-shell-linux-bash/>
3. https://id.wikibooks.org/wiki/Pemrograman_C/Perulangan