

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 13
REPEAT-UNTIL



Disusun oleh:

ZHAFIF YUSUF AL AMIN

109082500137

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {

    var waktu string

    var ulang int

    fmt.Scan(&waktu, &ulang)

    i := 0

    for kondisi := false; !kondisi; {

        fmt.Println(waktu)

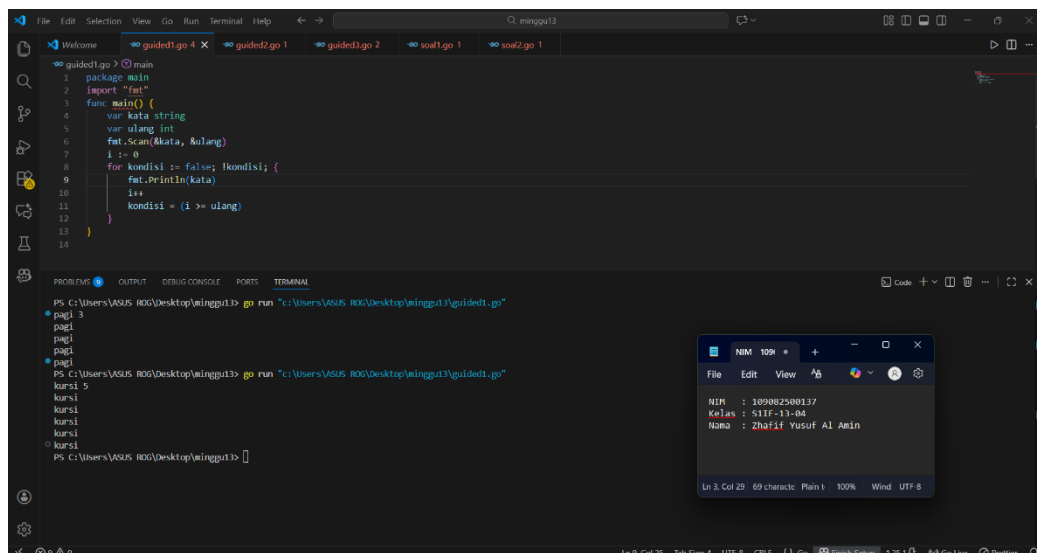
        i++

        kondisi = (i >= ulang)

    }

}
```

Screenshoot program



Deskripsi program

package main: menandakan program utama ini ibaratnya kita akan membangun rumah & package main ini sebagai temboknya.

`import "fmt"`: berfungsi memanggil package "fmt" ini ibarat bangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, ibarat rumah full tembok tanpa ada pintu masuk sama sekali.

`var kata string`: deklarasi variabel kata menggunakan tipe data string.

`var ulang int`: deklarasi variabel ulang menggunakan tipe data integer.

`fmt.Scan(&kata, &ulang)`: fungsi untuk menginput dan menyimpan isi dari variabel kata dan ulang. Ini berfungsi untuk di perulangan nanti.

`i:=0`: deklarasi otomatis variabel i dengan isi 0.

```
for kondisi := false; !kondisi; {  
    fmt.Println(waktu)  
    i++  
    kondisi = (i >= ulang)  
}
```

: perulangan for yang akan di eksekusi selama kondisi bernilai true dan akan berhenti jika kondisi bernilai false, dalam repeat until perulangan harus berhenti jika bernilai false.

Deklarasi variabel kondisi sebagai false lalu di negasikan sehingga bernilai true dengan aksi yang merupakan output dari isi dari variabel waktu. lalu isi dari variabel i akan di tambah 1. Akan dilakukan terus menerus berulang kali sampai i lebih besar atau sama dengan ulang. Sehingga kondisi akan berubah menjadi true lalu dinegasikan menjadi false

2. Guided 2

Source Code

```
package main

import "fmt"

func main(){
    var n int

    fmt.Scan(&n)

    for kondisi:=true; kondisi;{

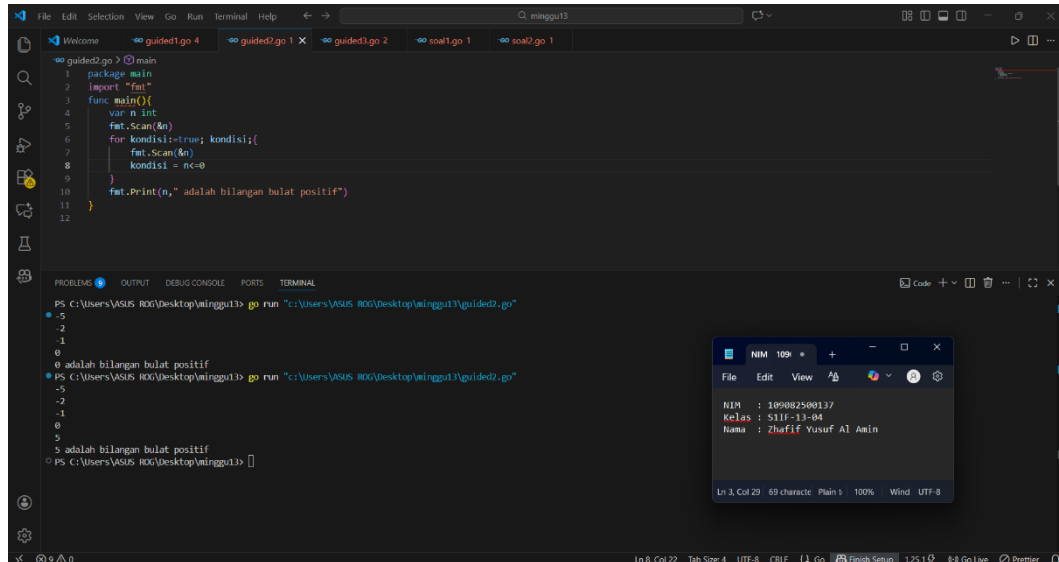
        fmt.Scan(&n)

        kondisi = n<=0

    }

    fmt.Print(n," adalah bilangan bulat positif")
}
```

Screenshoot program



Deskripsi program

package main : menandakan program utama ini ibaratnya kita akan membangun rumah & package main ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var n int`: deklarasi variabel n dengan tipe data integer.

```
for kondisi:=true; kondisi;{  
    fmt.Scan(&n)  
    kondisi = n<=0  
}  
fmt.Print(n," adalah bilangan bulat positif")  
}
```

: perulangan for yang akan di eksekusi selama kondisi bernilai true dan akan berhenti jika kondisi bernilai false, dalam repeat until perulangan harus berhenti jika bernilai false.

Deklarasi variabel kondisi sebagai true. Perulangan bernilai true dengan aksi yang merupakan input isi dari variabel n lagi, begitu seterusnya sampai kondisi bernilai false, kondisi bernilai false jika n yang di input lebih besar dari 0. Jadi kalo input 0 sampai negatif program akan terus terusan meminta inputan sesuai yang di perulangan.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    fmt.Scan(&x, &y)

    for kondisi := false; !kondisi; {

        x -= y

        kondisi = x == 0 || x < 0

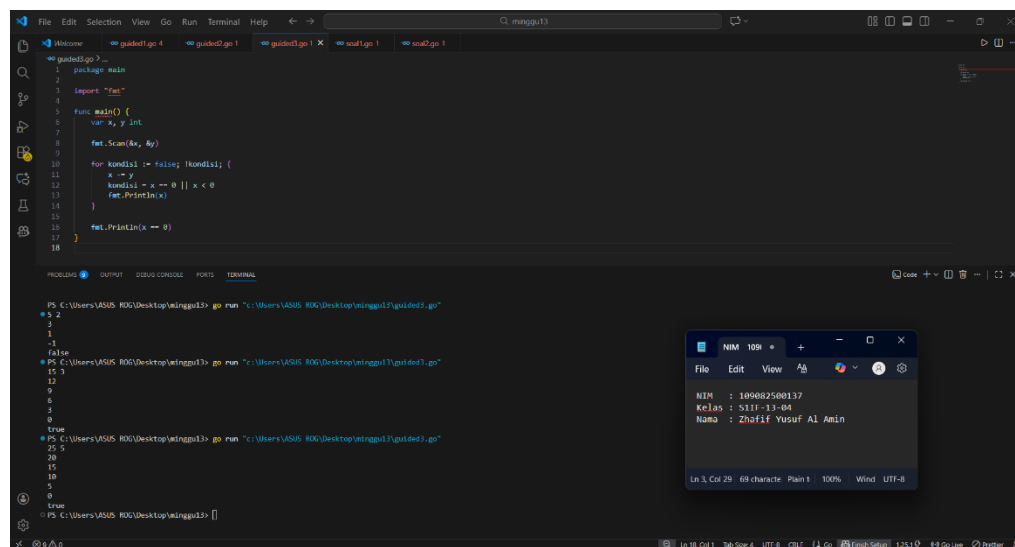
        fmt.Println(x)

    }

    fmt.Println(x == 0)

}
```

Screenshoot program



Deskripsi program

package main : menandakan program utama ini ibaratnya kita akan membangun rumah & package main ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var x, y int` : deklarasi variabel dengan tipe data int.

`fmt.Scan(&x, &y)` : fungsi untuk menginput dan menyimpan isi dari variabel x dan y.

```
for kondisi := false; !kondisi; {  
    x -= y  
    kondisi = x == 0 || x < 0  
    fmt.Println(x)  
}  
  
fmt.Println(x == 0)  
}
```

: perulangan for yang akan di eksekusi selama kondisi bernilai true dan akan berhenti jika kondisi bernilai false, dalam repeat until perulangan harus berhenti jika bernilai false.

Deklarasi variabel kondisi sebagai false lalu di negasikan sehingga bernilai true dengan aksi yang merupakan pengurangan x terhadap y disetiap perulangan sampai kondisi x sama dengan 0 atau x kurang dari 0 lalu x dioutputkan sesuai berapa nilai akhirnya. Diluar perulangan x di cek sebagai boolean apakah x bernilai 0, nanti outputnya akan berupa true atau false.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var u, p string

    fmt.Scan(&u, &p)

    ux := "Admin"

    px := "Admin"

    gagal := 0

    for u != ux || p != px {

        gagal++

        fmt.Scan(&u, &p)

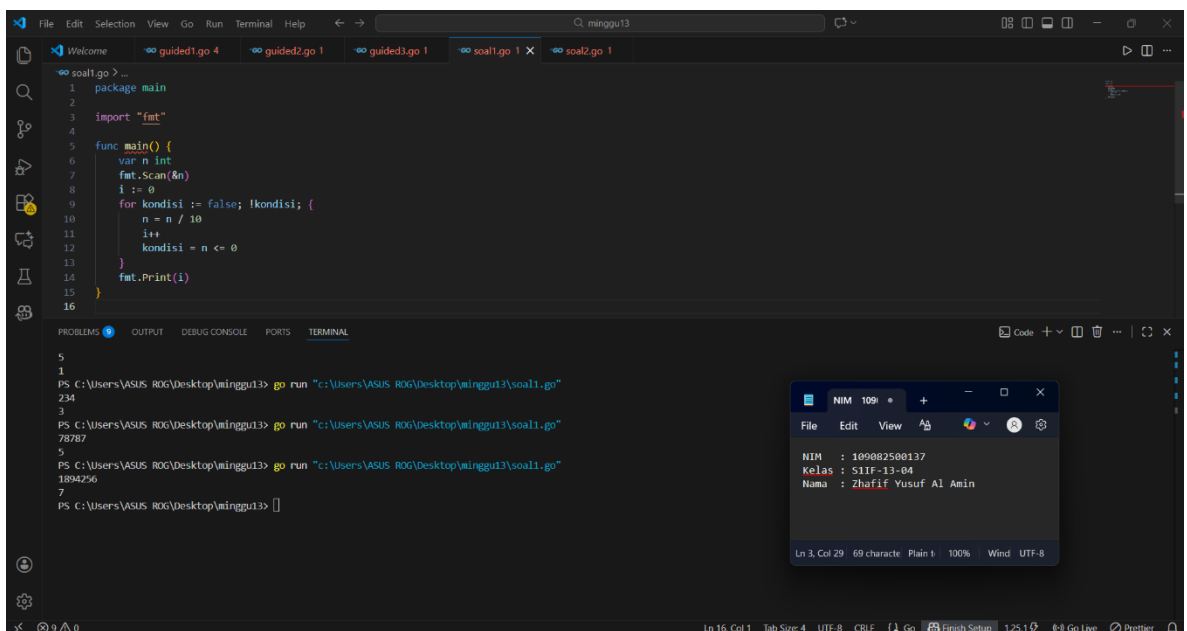
    }

    fmt.Println(gagal)

    fmt.Print("percobaan gagal login ")

}
```

Screenshoot program



Deskripsi program

`package main` : menandakan program utama ini ibaratnya kita akan membangun rumah & `package main` ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}` : ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var n int` : Pendeklarasian variabel `n` menggunakan tipe data `int`.

`fmt.Scan(&n)` : fungsi untuk menginput dan setelah di input isi dari variabel akan disimpan di `n`.

`i:=0`

: deklarasi otomatis variabel `i` itu isi variabelnya 0.

```
    for kondisi := false; !kondisi; {
        n = n / 10
        i++
        kondisi = n <= 0
    }
    fmt.Print(i)
}
```

: perulangan `for` yang akan di eksekusi selama kondisi bernilai `true` dan akan berhenti jika kondisi bernilai `false`, dalam `repeat until` perulangan harus berhenti jika bernilai `false`.

Deklarasi variabel kondisi sebagai `false` lalu di negasikan sehingga bernilai `true` dengan aksi yang merupakan pembagian `n` kepada 10 secara terus menerus. Nilai dari `i` akan ditambah 1 disetiap perulangan sampai kondisi bernilai `true` yaitu jika `n` lebih besar dari 0. Nilai `true` lalu di negasikan sehingga `false` lalu perulangan akan berhenti.

2. Tugas 2

Source code

```
package main

import "fmt"

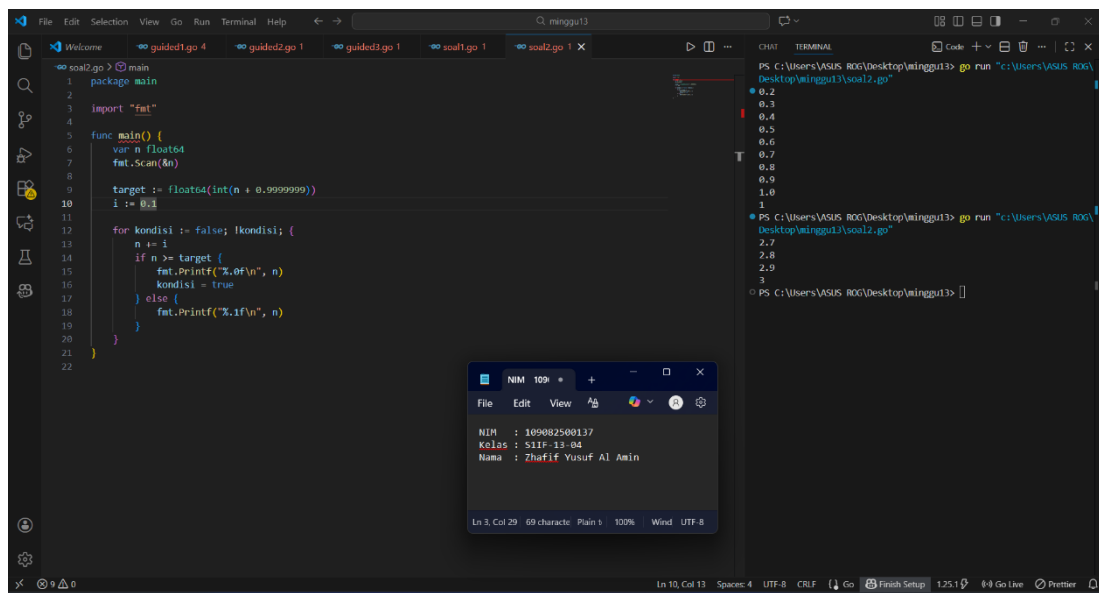
func main() {
    var n float64

    fmt.Scan(&n)

    target := float64(int(n + 0.9999999))
    i := 0.1

    for kondisi := false; !kondisi; {
        n += i
        if n >= target {
            fmt.Printf("%.0f\n", n)
            kondisi = true
        } else {
            fmt.Printf("%.1f\n", n)
        }
    }
}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution output in a terminal. The program is a simple loop that prints numbers from 0.2 to 3.0 in increments of 0.1. The terminal output shows the program running successfully and printing the expected sequence of numbers.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n float64
7     fmt.Scan(&n)
8
9     target := float64(int(n + 0.9999999))
10    i := 0.1
11
12    for kondisi := false; !kondisi; {
13        n += i
14        if n >= target {
15            fmt.Printf("%.0f\n", n)
16            kondisi = true
17        } else {
18            fmt.Printf("%.1f\n", n)
19        }
20    }
21 }
22
```

```
PS C:\Users\ASUS ROG\Desktop\minggu13> go run "c:\Users\ASUS ROG\Desktop\minggu13\soal2.go"
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1
2.7
2.8
2.9
3
PS C:\Users\ASUS ROG\Desktop\minggu13>
```

Deskripsi program

`package main` : menandakan program utama ini ibaratnya kita akan membangun rumah & `package main` ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package “fmt” ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}` : ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var n float64` : Deklarasi variabel n menggunakan tipe data float.

`fmt.Scan(&n)` : digunakan untuk membaca inputan dari variabel n lalu disimpan.

`target := float64(int(n + 0.9999999))` : deklarasi otomatis variabel target dengan nilai `n + 0.9999999` yang dimana sekaligus berfungsi untuk membulatkan ke atas dari n. Nilai n terlebih dahulu ditambah dengan 0.9999999 agar naik mendekati bilangan bulat berikutnya, kemudian dikonversi ke tipe int sehingga bagian desimalnya dibuang. Setelah itu hasilnya dikembalikan lagi ke tipe float64 agar dapat dibandingkan dengan variabel n yang juga bertipe float64.

`i := 0.1` : deklarasi otomatis variabel h dengan nilai 0.1.

```

for kondisi := false; !kondisi; {
    n += i
    if n >= target {
        fmt.Printf("%.0f\n", n)
        kondisi = true
    } else {
        fmt.Printf("%.1f\n", n)
    }
}
}

```

: perulangan for yang akan di eksekusi selama kondisi bernilai true dan akan berhenti jika kondisi bernilai false, dalam repeat until perulangan harus berhenti jika bernilai false.

Deklarasi variabel kondisi sebagai false lalu di negasikan sehingga bernilai true dengan aksi yang merupakan penambahan nilai n sebanyak i disetiap perulangan sampai n memenuhi atau lebih besar dari target barulah berhenti. Disetiap perulangan nilai n akan di tampilkan sampai memenuhi target. Target sudah kita deklarasikan di atas tadi.

3. Tugas 3

Source code

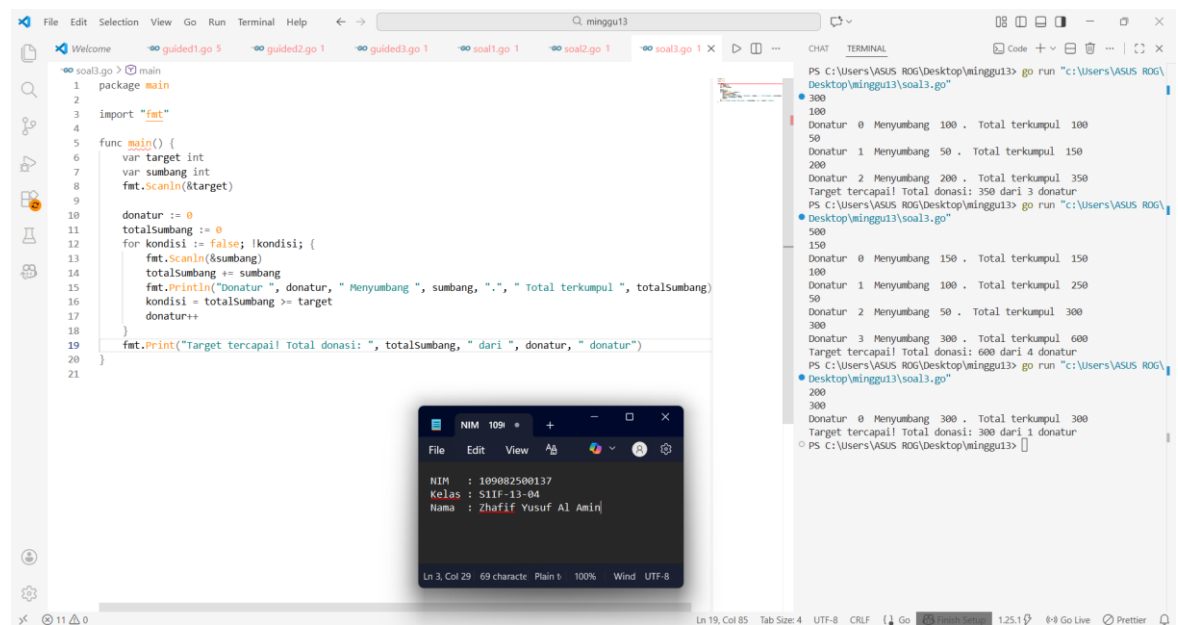
```
package main

import "fmt"

func main() {
    var target int
    var sumbang int
    fmt.Scanln(&target)

    donatur := 0
    totalSumbang := 0
    for kondisi := false; !kondisi; {
        fmt.Scanln(&sumbang)
        totalSumbang += sumbang
        fmt.Println("Donatur ", donatur, " Menyumbang ",
sumbang, ".", " Total terkumpul ", totalSumbang)
        kondisi = totalSumbang >= target
        donatur++
    }
    fmt.Print("Target tercapai! Total donasi: ",
totalSumbang, " dari ", donatur, " donatur")
}
```

Screenshoot program



Deskripsi program

package main : menandakan program utama ini ibaratnya kita akan membangun rumah & package main ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var target int`: deklarasi variabel target menggunakan tipe data integer.

`var sumbang int`: deklarasi variabel sumbang menggunakan tipe data integer.

`fmt.Scanln(&target)` : berfungsi untuk membaca inputan yang telah kita input lalu disimpan ke variabel `&target`.

`donatur:= 0` : deklarasi otomatis variabel donatur dengan nilai 0

`totalSumbang:= 0` : deklarasi otomatis variabel totalSumbang dengan nilai 0.

```
for kondisi := false; !kondisi; {  
    fmt.Scanln(&sumbang)  
    totalSumbang += sumbang  
    fmt.Println("Donatur ", donatur, " Menyumbang  
", sumbang, ".", " Total terkumpul ", totalSumbang)  
    kondisi = totalSumbang >= target  
    donatur++  
}  
  
fmt.Print("Target tercapai! Total donasi: ",  
totalSumbang, " dari ", donatur, " donatur")  
}
```

: perulangan for yang akan di eksekusi selama kondisi bernilai true dan akan berhenti jika kondisi bernilai false, dalam repeat until perulangan harus berhenti jika bernilai false.

Deklarasi variabel kondisi sebagai false lalu di negasikan sehingga bernilai true dengan aksi menginput nilai dari variabel sumbang lalu totalSumbang akan di jumlahkan disetiap perulangannya dengan variabel sumbang. Menampilkan output sesuai yang tertera lalu akan berhenti jika kondisinya totalSumbang lebih besar atau sama dengan target. Setiap perulangan nilai dari donatur akan ditambah 1. Jika perulangan sudah selesai maka hasil akhirnya adalah sesuai pada kode.