

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 13  
REPEAT - UNTIL**



**Disusun oleh:**

**Andra Dwicki Saputra**

**109082500206**

**S1IF-13-07**

**Asisten Praktikum**

Adithana dharma putra

Apri pandu wicaksono

**PROGRAM STUDI S1 INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## LATIHAN KELAS – GUIDED

### 1. Guided 1 Source Code

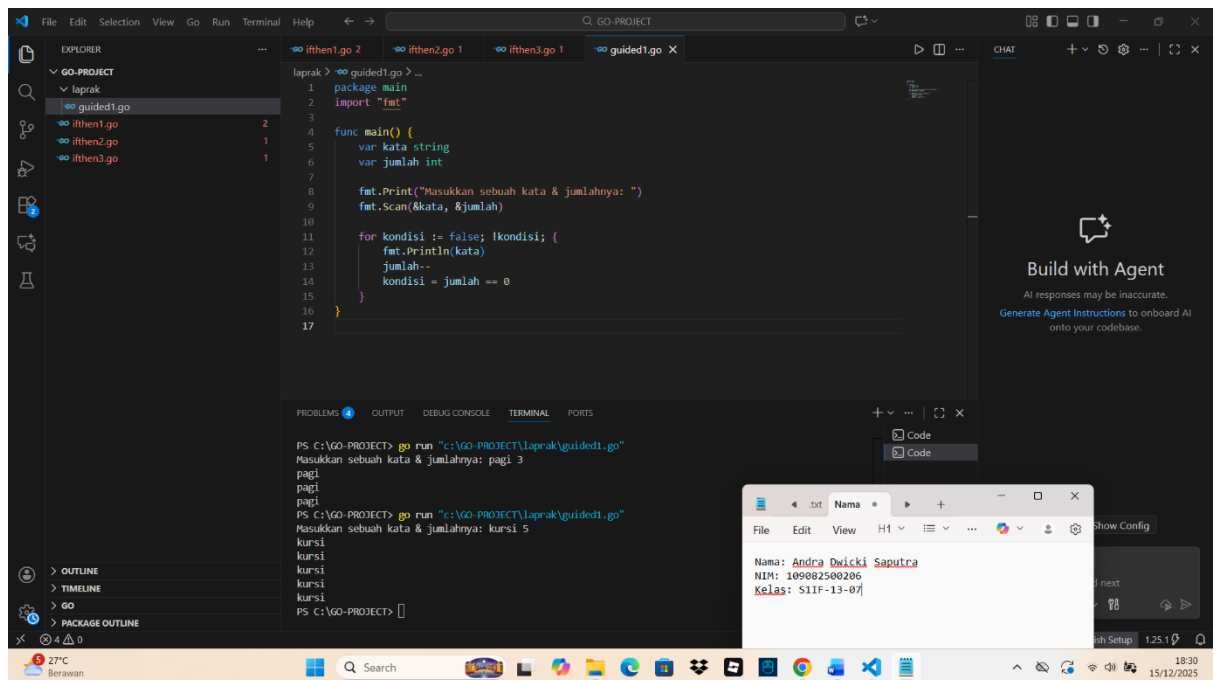
```
package main
import "fmt"

func main() {
    var kata string
    var jumlah int

    fmt.Print("Masukkan sebuah kata & jumlahnya: ")
    fmt.Scan(&kata, &jumlah)

    for kondisi := false; !kondisi; {
        fmt.Println(kata)
        jumlah--
        kondisi = jumlah == 0
    }
}
```

## Screenshoot program



## Deskripsi program

Program ini dibuat untuk menampilkan sebuah kata sebanyak jumlah yang ditentukan oleh pengguna dengan memanfaatkan perulangan dan logika kondisi. Pada awal program, dideklarasikan dua variabel, yaitu kata untuk menyimpan teks yang akan ditampilkan dan jumlah untuk menyimpan berapa kali kata tersebut ingin dicetak. Program kemudian meminta pengguna memasukkan sebuah kata beserta jumlah pengulangannya melalui perintah "Masukkan sebuah kata & jumlahnya:", lalu menyimpan input tersebut ke dalam variabel yang sesuai. Setelah input diterima, program menjalankan perulangan for dengan sebuah variabel kondisi bernama kondisi yang diinisialisasi dengan nilai false. Perulangan ini menggunakan syarat !kondisi, yang berarti perulangan akan terus berjalan selama nilai kondisi masih false. Dengan kata lain, selama jumlah pengulangan belum habis, program akan terus mencetak kata. Di dalam perulangan, program menampilkan isi variabel kata ke layar, kemudian mengurangi nilai jumlah sebanyak satu. Setelah itu, program melakukan pengecekan apakah nilai jumlah sudah sama dengan nol. Jika nilai jumlah belum nol, maka kondisi tetap bernilai false sehingga perulangan berlanjut. Namun, ketika jumlah sudah mencapai nol, nilai kondisi berubah menjadi true. Akibatnya, kondisi perulangan !kondisi menjadi false dan perulangan pun berhenti. Dengan mekanisme tersebut, program memastikan bahwa kata yang dimasukkan oleh pengguna akan ditampilkan tepat sebanyak jumlah yang diinginkan. Program ini juga menunjukkan Penggunaan perulangan for dengan pengendalian kondisi secara manual menggunakan variabel boolean dan operator negasi

## 2. Guided 2

### Source Code

```
package main
import "fmt"

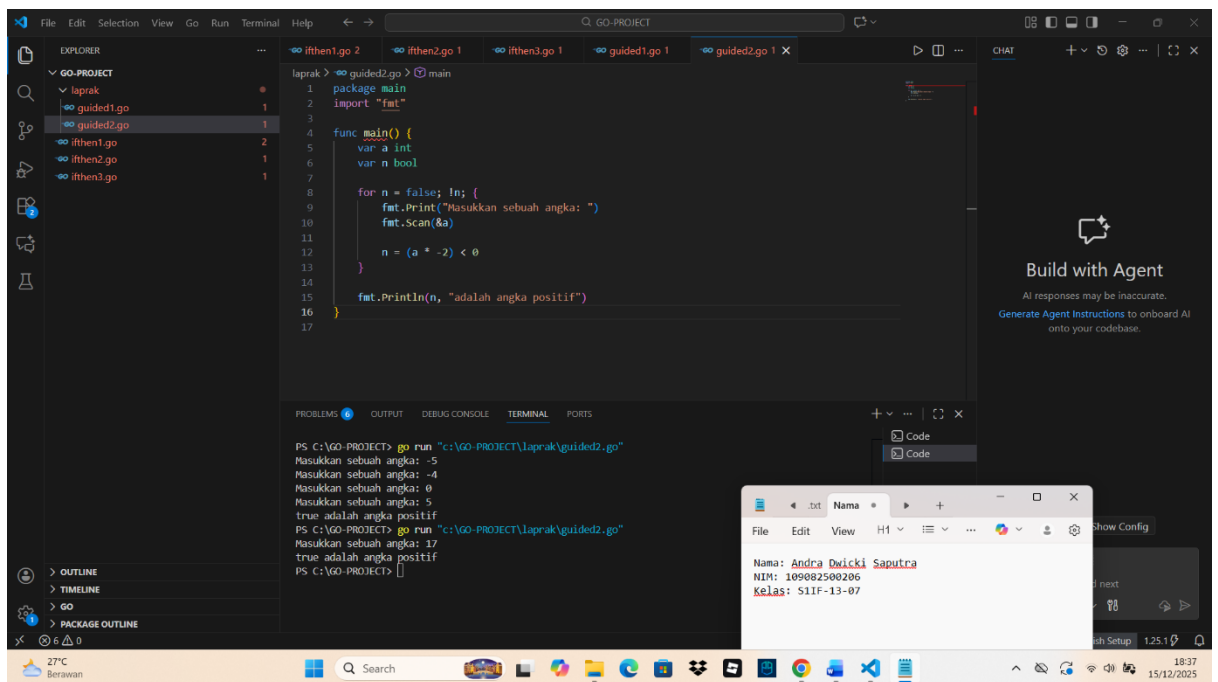
func main() {
    var a int
    var n bool

    for n = false; !n; {
        fmt.Print("Masukkan sebuah angka: ")
        fmt.Scan(&a)

        n = (a * -2) < 0
    }

    fmt.Println(n, "adalah angka positif")
}
```

## Screenshoot program



```
1 package main
2 import "fmt"
3
4 func main() {
5     var a int
6     var n bool
7
8     for n = false; !n; {
9         fmt.Print("Masukkan sebuah angka: ")
10        fmt.Scan(&a)
11
12        n = (a * -2) < 0
13    }
14
15    fmt.Println(n, "adalah angka positif")
16 }
17
```

PS C:\GO-PROJECT> go run "c:\GO-PROJECT\laprak\guided2.go"

Masukkan sebuah angka: -5  
Masukkan sebuah angka: -4  
Masukkan sebuah angka: 0  
Masukkan sebuah angka: 5  
true adalah angka positif

PS C:\GO-PROJECT> go run "c:\GO-PROJECT\laprak\guided2.go"

Masukkan sebuah angka: 17  
true adalah angka positif

PS C:\GO-PROJECT>

## Deskripsi program

Program ini dibuat untuk mengecek apakah sebuah angka yang dimasukkan oleh pengguna merupakan angka positif. Pada awal program, dideklarasikan dua variabel, yaitu `a` bertipe integer untuk menyimpan angka yang dimasukkan pengguna, dan `n` bertipe boolean yang berfungsi sebagai penanda kondisi perulangan. Program kemudian menjalankan sebuah perulangan `for` dengan nilai awal `n` adalah `false` dan kondisi perulangan `!n`. Artinya, selama `n` masih bernilai `false`, program akan terus meminta pengguna memasukkan sebuah angka. Di dalam perulangan, program menampilkan pesan "Masukkan sebuah angka:" lalu membaca input pengguna dan menyimpannya ke dalam variabel `a`. Setelah angka dimasukkan, program melakukan pengecekan dengan ekspresi  $(a * -2) < 0$ . Jika hasil perkalian tersebut bernilai negatif, maka variabel `n` akan bernilai `true`. Secara logika, kondisi ini hanya akan terpenuhi jika nilai `a` adalah angka positif. Ketika `n` berubah menjadi `true`, kondisi perulangan `!n` menjadi `false`, sehingga perulangan berhenti. Setelah keluar dari perulangan, program menampilkan hasil pengecekan dengan mencetak nilai `n` beserta pesan "adalah angka positif". Hal ini menunjukkan bahwa program hanya akan berhenti ketika pengguna berhasil memasukkan sebuah angka positif.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var x, y int

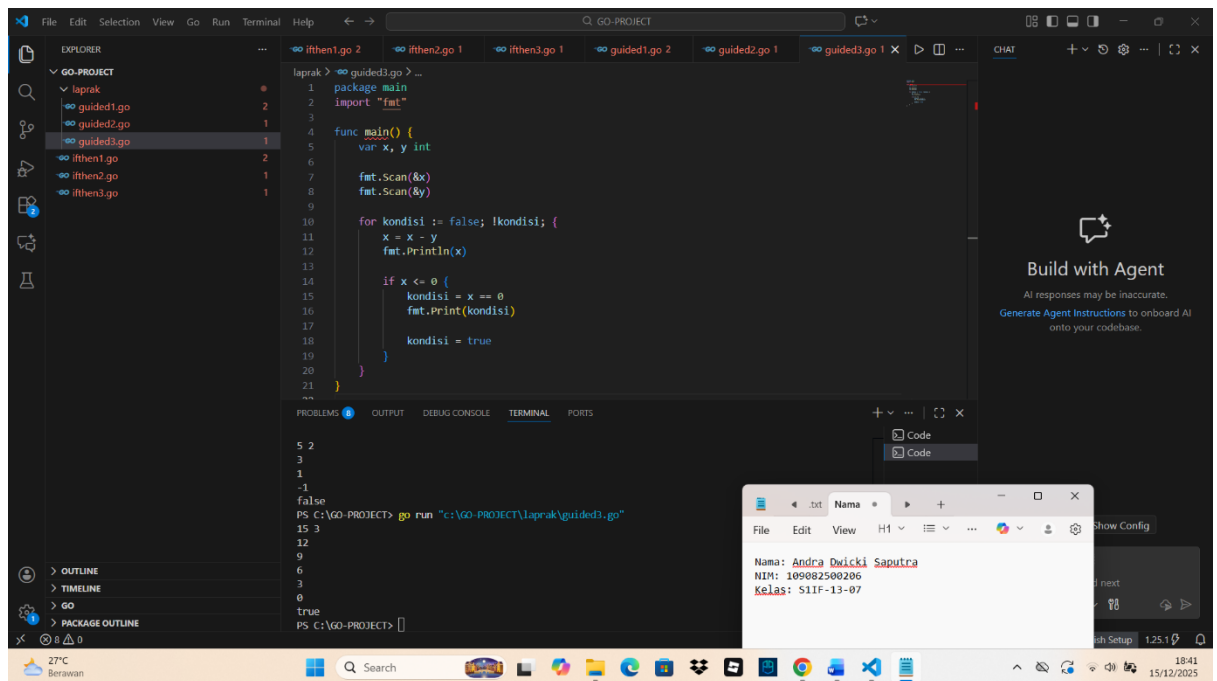
    fmt.Scan(&x)
    fmt.Scan(&y)

    for kondisi := false; !kondisi; {
        x = x - y
        fmt.Println(x)

        if x <= 0 {
            kondisi = x == 0
            fmt.Print(kondisi)

            kondisi = true
        }
    }
}
```

## Screenshoot program



The screenshot shows a Go IDE with a project named 'GO-PROJECT'. The Explorer panel on the left shows a file tree with 'laprak' as the root, containing 'guided1.go', 'guided2.go', and 'guided3.go'. The main editor displays the code for 'guided3.go':

```
1 package main
2 import "fmt"
3
4 func main() {
5     var x, y int
6
7     fmt.Scan(&x)
8     fmt.Scan(&y)
9
10    for kondisi := false; !kondisi; {
11        x = x - y
12        fmt.Println(x)
13
14        if x <= 0 {
15            kondisi = x == 0
16            fmt.Print(kondisi)
17            kondisi = true
18        }
19    }
20 }
21
```

The terminal at the bottom shows the execution of the program:

```
PS C:\GO-PROJECT> go run "c:\GO-PROJECT\laprak\guided3.go"
5 2
3
1
-1
false
PS C:\GO-PROJECT> go run "c:\GO-PROJECT\laprak\guided3.go"
15 3
12
9
6
3
0
true
PS C:\GO-PROJECT>
```

A small window titled 'Nama' is open in the foreground, displaying the following information:

```
Nama: Andra Dwicki Saputra
NIM: 109082500206
Kelas: S11F-13-07
```

## Deskripsi program

Program ini dibuat untuk melakukan pengurangan berulang antara dua bilangan bulat yang dimasukkan oleh pengguna. Pada awal program, dua variabel bertipe integer, yaitu `x` dan `y`, dideklarasikan untuk menyimpan nilai input. Program kemudian membaca nilai `x` dan `y` secara berurutan menggunakan `fmt.Scan`. Setelah kedua nilai diperoleh, program menjalankan sebuah perulangan `for` dengan variabel `kondisi` bertipe boolean yang diinisialisasi dengan nilai `false`. Perulangan ini menggunakan syarat `!kondisi`, yang berarti perulangan akan terus berjalan selama nilai `kondisi` masih bernilai `false`. Di dalam perulangan, nilai `x` akan dikurangi dengan `y`, kemudian hasil pengurangan tersebut langsung ditampilkan ke layar menggunakan `fmt.Println(x)`. Dengan demikian, pengguna dapat melihat hasil pengurangan `x` secara bertahap pada setiap iterasi. Selanjutnya, program melakukan pengecekan menggunakan percabangan `if x <= 0`. Ketika nilai `x` sudah kurang dari atau sama dengan nol, program masuk ke dalam blok `if`. Pada bagian ini, nilai `kondisi` sempat diisi dengan hasil perbandingan `x == 0`, lalu dicetak ke layar. Namun setelah itu, nilai `kondisi` langsung diubah menjadi `true`, sehingga kondisi perulangan `!kondisi` menjadi `false` dan perulangan berhenti.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main
import "fmt"

func main() {
    var bilangan int

    fmt.Print("Masukkan sebuah bilangan: ")
    fmt.Scan(&bilangan)

    jumlahdigit := 0

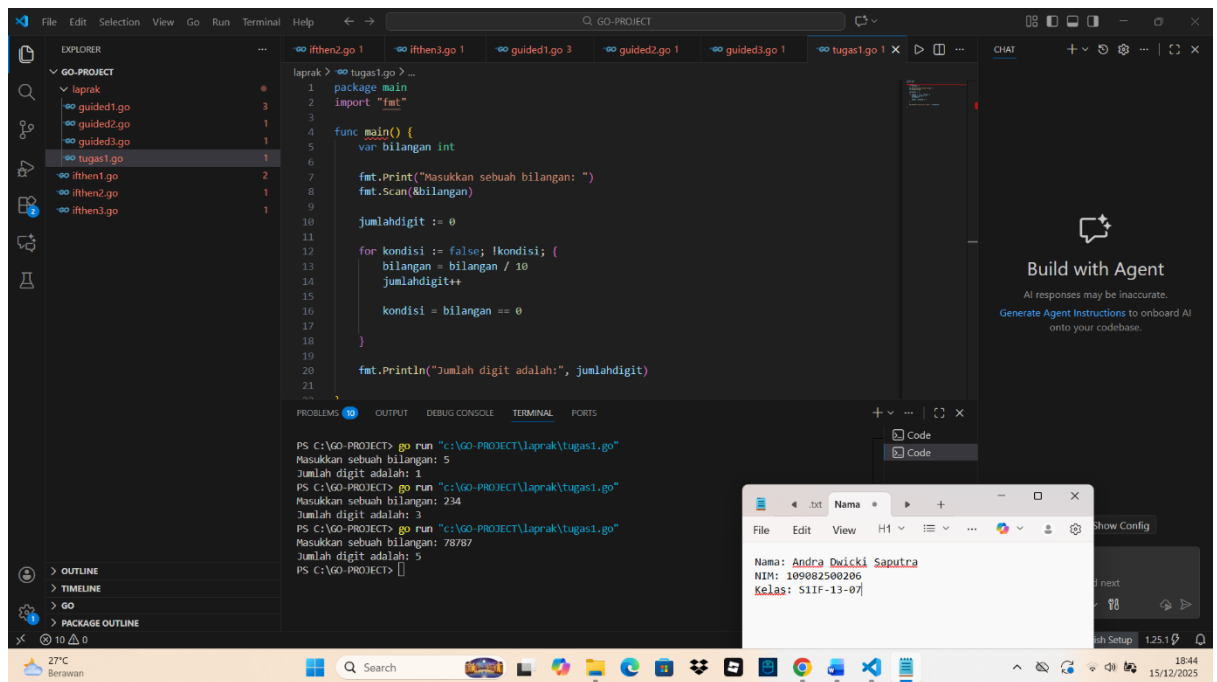
    for kondisi := false; !kondisi; {
        bilangan = bilangan / 10
        jumlahdigit++

        kondisi = bilangan == 0
    }

    fmt.Println("Jumlah digit adalah:", jumlahdigit)
}
```



## Screenshoot program



## Deskripsi program

Program ini dibuat untuk menghitung jumlah digit dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Pada awal program, dideklarasikan sebuah variabel bilangan bertipe integer untuk menyimpan nilai yang dimasukkan oleh pengguna. Program kemudian menampilkan pesan "Masukkan sebuah bilangan:" dan membaca input tersebut menggunakan `fmt.Scan`. Setelah nilai bilangan diperoleh, program menyiapkan variabel `jumlahdigit` yang berfungsi sebagai penghitung banyaknya digit pada bilangan tersebut. Nilai awal variabel ini adalah 0. Proses penghitungan jumlah digit dilakukan menggunakan perulangan `for` dengan variabel kondisi bernama `kondisi` yang diinisialisasi dengan nilai `false`. Perulangan ini menggunakan syarat `!kondisi`, yang berarti perulangan akan terus berjalan selama kondisi masih bernilai `false`. Di dalam perulangan, nilai bilangan dibagi dengan 10 menggunakan pembagian bilangan bulat. Operasi ini bertujuan untuk menghilangkan satu digit terakhir dari bilangan pada setiap iterasi. Setiap kali pembagian dilakukan, nilai `jumlahdigit` akan bertambah satu. Setelah itu, program mengecek apakah nilai bilangan sudah menjadi 0. Jika sudah, variabel kondisi diubah menjadi `true` sehingga perulangan akan berhenti. Setelah perulangan selesai, program menampilkan hasil perhitungan berupa jumlah digit dari bilangan yang dimasukkan dengan pesan "Jumlah digit adalah:" diikuti oleh nilai `jumlahdigit`.

## 2. Tugas 2

### Source code

```
package main
import "fmt"

func main() {
    var n float64

    fmt.Print("Masukkan sebuah bilangan desimal: ")
    fmt.Scan(&n)

    bilBul := int(n)
    target := bilBul

    if n != float64(bilBul) {
        target = bilBul + 1
    }

    for kondisi:= false; !kondisi; {
        n = n + 0.1
        fmt.Printf("%.1f\n", n)

        kondisi = n >= float64(target) - 0.000001
    }
}
```

## Screenshoot program

```
1 package main
2 import "fmt"
3
4 func main() {
5     var n float64
6
7     fmt.Print("Masukkan sebuah bilangan desimal: ")
8     fmt.Scan(&n)
9
10    bilBul := int(n)
11    target := bilBul
12
13    if n != float64(bilBul) {
14        target = bilBul + 1
15    }
16
17    for kondisi := false; !kondisi; {
18        n = n + 0.1
19        fmt.Printf("%.1f\n", n)
20
21        kondisi = n >= float64(target) - 0.000001
22    }
```

PS C:\GO-PROJECT> go run "c:\GO-PROJECT\laprak\tugas2.go"

Masukkan sebuah bilangan desimal: 0.2

0.3  
0.4  
0.5  
0.6  
0.7  
0.8  
0.9  
1.0  
2.7  
2.8  
2.9

PS C:\GO-PROJECT> go run "c:\GO-PROJECT\laprak\tugas2.go"

Masukkan sebuah bilangan desimal: 2.7

2.8  
2.9

## Deskripsi program

Program ini digunakan untuk menaikkan sebuah bilangan desimal secara bertahap hingga mencapai bilangan bulat berikutnya. Pengguna diminta memasukkan sebuah bilangan desimal yang kemudian disimpan dalam variabel `n`. Nilai desimal tersebut diubah ke bilangan bulat untuk menentukan batas akhir (`target`). Jika bilangan yang dimasukkan bukan bilangan bulat, maka `target` dinaikkan satu agar program berhenti di bilangan bulat terdekat di atasnya. Selanjutnya, program menggunakan perulangan untuk menambah nilai `n` sebesar 0.1 pada setiap iterasi dan menampilkannya ke layar. Perulangan akan berhenti ketika nilai `n` sudah mencapai atau melewati `target` yang ditentukan. Program ini menunjukkan penggunaan perulangan, operasi bilangan desimal, dan pengecekan kondisi untuk mengendalikan proses perhitungan.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {

    var target, jumlah, donasi int

    fmt.Print("Masukkan target donasi: ")

    fmt.Scan(&target)

    jumlah = 0

    donatur := 0

    for kondisi := false; !kondisi; {

        fmt.Print("Masukkan donasi: ")

        fmt.Scan(&donasi)

        jumlah += donasi

        donatur++

        fmt.Printf("Donatur %d: Menyumbang %d. Total  
Terkumpul: %d\n", donatur, donasi, jumlah)
```

```

        kondisi = jumlah >= target
    }

    fmt.Printf("Target tercapai total donasi: %d dari %d
donatur\n", jumlah, donatur)
}

```

## Screenshoot program

```

4 func main() {
5     var target, jumlah, donasi int
6
7     fmt.Print("Masukkan target donasi: ")
8     fmt.Scan(&target)
9
10    jumlah = 0
11    donatur := 0
12
13    for kondisi := false; !kondisi; {
14        fmt.Print("Masukkan donasi: ")
15        fmt.Scan(&donasi)
16
17        jumlah += donasi
18        donatur++
19
20        fmt.Printf("Donatur %d: Menyumbang %d. Total Terkumpul: %d\n", donatur, donasi, jumlah)
21
22        kondisi = jumlah >= target
23    }
24
25    fmt.Printf("Target tercapai total donasi: %d dari %d donatur\n", jumlah, donatur)
26
27 }

```

```

PS C:\GO-PROJECT> go run "c:\GO-PROJECT\laprak\tugas3.go"
Masukkan target donasi: 300
Masukkan donasi: 100
Donatur 1: Menyumbang 100. Total Terkumpul: 100
Masukkan donasi: 50
Donatur 2: Menyumbang 50. Total Terkumpul: 150
Masukkan donasi: 200
Donatur 3: Menyumbang 200. Total Terkumpul: 350
Target tercapai total donasi: 350 dari 3 donatur
PS C:\GO-PROJECT>

```

## Deskripsi program

Program ini dibuat untuk mencatat dan memantau pengumpulan donasi hingga mencapai target yang ditentukan. Di awal program, pengguna diminta memasukkan target donasi yang ingin dicapai, lalu program menyiapkan variabel jumlah untuk menyimpan total donasi dan donatur untuk menghitung jumlah penyumbang. Selanjutnya, program menjalankan perulangan yang akan terus meminta input donasi dari pengguna. Setiap donasi yang masuk akan ditambahkan ke total donasi, jumlah donatur akan bertambah satu, dan informasi donasi langsung ditampilkan, mulai dari nomor donatur, jumlah sumbangan, hingga total dana yang sudah terkumpul. Perulangan akan berhenti ketika total donasi sudah sama dengan atau melebihi target yang ditentukan. Setelah itu, program menampilkan pesan akhir bahwa target donasi telah tercapai beserta total dana dan jumlah donatur yang berpartisipasi