

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 13
REPEAT-UNTIL



Disusun oleh:
ALIF RIZKI PANGESTU
109082530004
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 word string
    var repetitions int
    fmt.Scan(&word, &repetitions)
    counter := 0
    for done := false; !done; {
        fmt.Println(word)
        counter++
        done = (counter >= repetitions)
    }
}
```

Screenshoot program

```
laprak2 > go guided.13.1.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     var word string
7     var repetitions int
8     fmt.Scan(&word, &repetitions)
9     counter := 0
10    for done := false; !done; {
11        fmt.Println(word)
12        counter++
13        done = (counter >= repetitions)
14    }
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS D:\laprak2> go run "d:\laprak2\guided.13.1.go"
pagi 3
pagi
pagi
pagi
kursi 5
kursi
kursi
kursi
kursi
kursi
PS D:\laprak2>
```

File Edit View

ALIF RIZKI PANGESTU
109082530004
S1IF-13-04

Ln 1, Col 21 | 44 character Plain text | 130% | Window UTF-8

Deskripsi program

Program Go pada gambar tersebut digunakan untuk menampilkan sebuah kata ke layar sebanyak jumlah pengulangan yang ditentukan oleh pengguna dengan memanfaatkan perulangan for berbasis kondisi logika. Program mendeklarasikan variabel word untuk menyimpan teks dan repetitions untuk menyimpan jumlah pengulangan, kemudian membaca keduanya dari input menggunakan fmt.Scan. Selanjutnya, program menginisialisasi variabel counter sebagai penghitung dan menjalankan perulangan for dengan variabel boolean done sebagai penentu berhenti atau tidaknya perulangan. Di setiap iterasi, program mencetak nilai word, menaikkan nilai counter, dan mengecek apakah jumlah pengulangan yang telah dilakukan sudah sama atau melebihi nilai repetitions; jika sudah terpenuhi, variabel done diatur menjadi true sehingga perulangan berhenti.

2. Guided 2

Source Code

```
package main

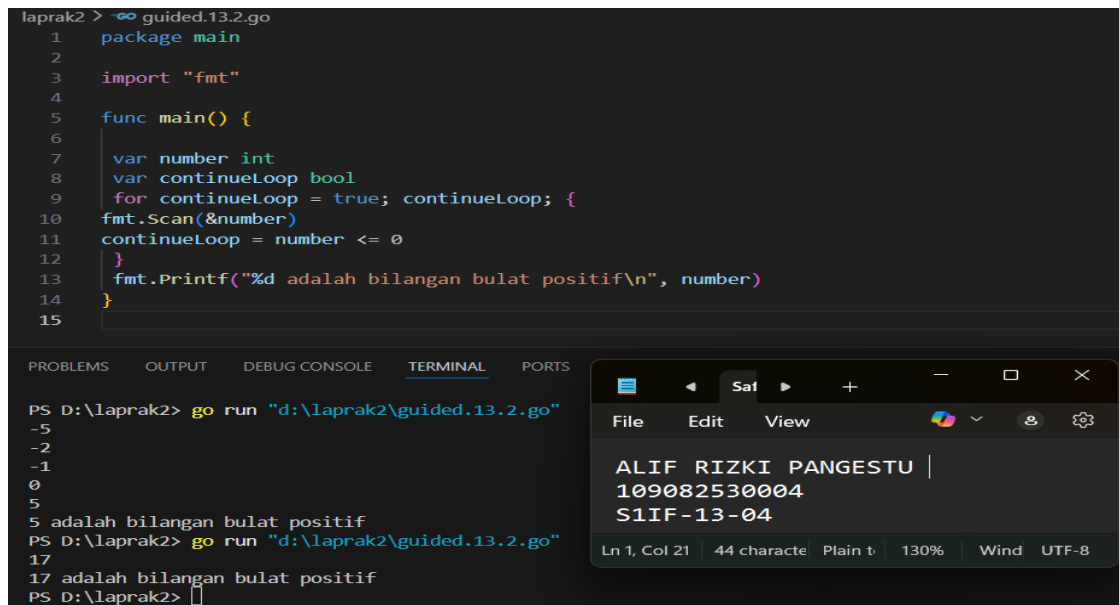
import "fmt"

func main() {

    var number int
    var continueLoop bool
    for continueLoop = true; continueLoop; {
fmt.Scan(&number)
continueLoop = number <= 0
    }

    fmt.Printf("%d adalah bilangan bulat positif\n", number)
}
```

Screenshoot program



```
laprak2 > go run guided.13.2.go
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var number int
8     var continueLoop bool
9     for continueLoop = true; continueLoop; {
10 fmt.Scan(&number)
11 continueLoop = number <= 0
12     }
13     fmt.Printf("%d adalah bilangan bulat positif\n", number)
14 }
15

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\laprak2> go run "d:\laprak2\guided.13.2.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS D:\laprak2> go run "d:\laprak2\guided.13.2.go"
17
17 adalah bilangan bulat positif
PS D:\laprak2>
```

ALIF RIZKI PANGESTU |
109082530004
S1IF-13-04

Ln 1, Col 21 | 44 character | Plain text | 130% | Wind UTF-8

Deskripsi program

Program Go pada gambar tersebut bertujuan untuk memastikan pengguna memasukkan sebuah bilangan bulat positif dengan memanfaatkan perulangan for berbasis kondisi boolean. Program mendeklarasikan variabel `number` untuk menyimpan nilai input dan `continueLoop` sebagai pengendali perulangan, lalu menjalankan perulangan for selama `continueLoop` bernilai `true`. Di dalam perulangan, program membaca input menggunakan `fmt.Scan` dan mengevaluasi apakah nilai `number` kurang dari atau sama dengan nol; jika iya, `continueLoop` tetap bernilai `true` sehingga pengguna diminta memasukkan ulang angka, dan jika nilai lebih besar dari nol, `continueLoop` diubah menjadi `false` sehingga perulangan berhenti. Setelah keluar dari perulangan, program menampilkan pesan bahwa nilai yang dimasukkan merupakan bilangan bulat positif menggunakan `fmt.Printf`.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var x int

    var y int

    var selesai bool

    fmt.Scan(&x, &y)

    for selesai = false; !selesai; {

x = x - y

    fmt.Println(x)

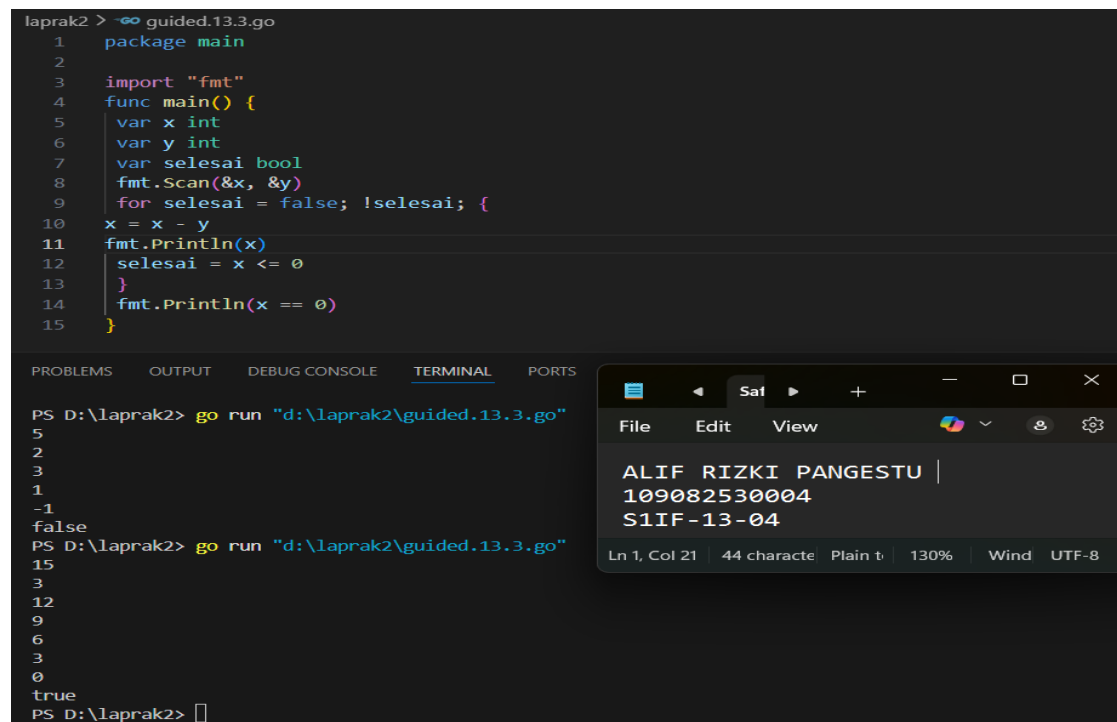
    selesai = x <= 0

    }

    fmt.Println(x == 0)

}
```

Screenshoot program



```
laprak2 > go guided.13.3.go
1 package main
2
3 import "fmt"
4 func main() {
5     var x int
6     var y int
7     var selesai bool
8     fmt.Scan(&x, &y)
9     for selesai = false; !selesai; {
10 x = x - y
11     fmt.Println(x)
12     selesai = x <= 0
13     }
14     fmt.Println(x == 0)
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS D:\laprak2> go run "d:\laprak2\guided.13.3.go"
5
2
3
1
-1
false
15
3
12
9
6
3
0
true
PS D:\laprak2>
```

File Edit View

ALIF RIZKI PANGESTU |
109082530004
S1IF-13-04

Ln 1, Col 21 | 44 character Plain text | 130% | Window UTF-8

Deskripsi program

Program Go pada gambar tersebut berfungsi untuk melakukan pengurangan berulang antara dua bilangan bulat hingga nilai hasilnya kurang dari atau sama dengan nol dengan menggunakan perulangan for berbasis kondisi boolean. Program menerima dua input bilangan bulat, yaitu x dan y, kemudian menjalankan perulangan selama variabel selesai bernilai false. Di dalam perulangan, nilai x dikurangi dengan y pada setiap iterasi, lalu hasil sementara x ditampilkan ke layar menggunakan `fmt.Println`. Setelah itu, program mengecek apakah nilai x sudah kurang dari atau sama dengan nol; jika kondisi tersebut terpenuhi, variabel selesai diubah menjadi true sehingga perulangan berhenti. Setelah keluar dari perulangan, program menampilkan hasil pengecekan logika `x == 0` untuk menunjukkan apakah nilai akhir x tepat sama dengan nol atau tidak.

TUGAS

1. Tugas 1

Source code

```
package main

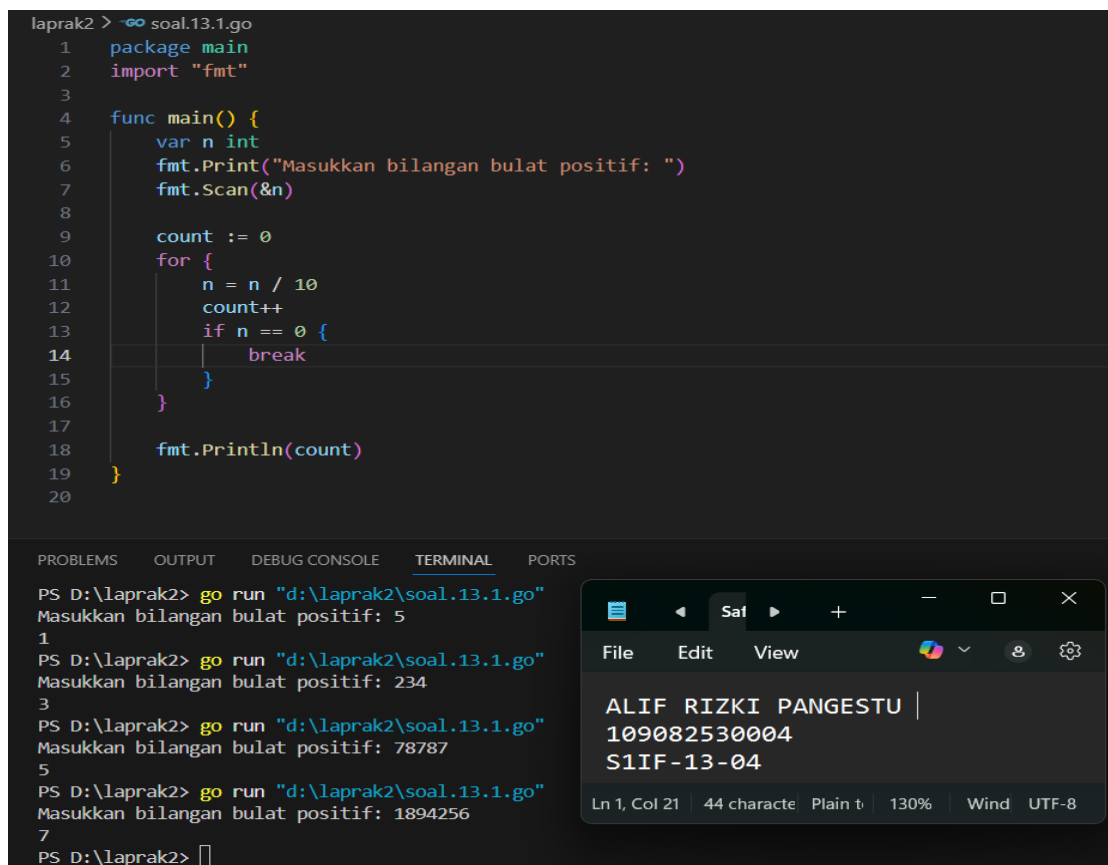
import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif: ")
    fmt.Scan(&n)

    count := 0
    for {
        n = n / 10
        count++
        if n == 0 {
            break
        }
    }

    fmt.Println(count)
}
```

Screenshoot program



The screenshot shows a Go program being executed in a terminal window. The program prompts the user to enter a positive integer, and the user enters 5. The program then calculates the number of digits in the integer by repeatedly dividing it by 10 until it reaches 0. The output shows the count of digits for three different inputs: 5, 234, and 78787.

```
laprak2 > go run soal.13.1.go
1 package main
2 import "fmt"
3
4 func main() {
5     var n int
6     fmt.Print("Masukkan bilangan bulat positif: ")
7     fmt.Scan(&n)
8
9     count := 0
10    for {
11        n = n / 10
12        count++
13        if n == 0 {
14            break
15        }
16    }
17
18    fmt.Println(count)
19 }
20
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS D:\laprak2> go run "d:\laprak2\soal.13.1.go"
Masukkan bilangan bulat positif: 5
PS D:\laprak2> go run "d:\laprak2\soal.13.1.go"
Masukkan bilangan bulat positif: 234
PS D:\laprak2> go run "d:\laprak2\soal.13.1.go"
Masukkan bilangan bulat positif: 78787
PS D:\laprak2> go run "d:\laprak2\soal.13.1.go"
Masukkan bilangan bulat positif: 1894256
PS D:\laprak2>
```

File Edit View

ALIF RIZKI PANGESTU |
109082530004
S1IF-13-04

Ln 1, Col 21 | 44 character Plain text | 130% | Window UTF-8

Deskripsi program

Program Go tersebut berfungsi untuk menghitung jumlah digit dari sebuah bilangan bulat positif yang dimasukkan oleh pengguna melalui keyboard. Program dimulai dengan mendeklarasikan paket main dan mengimpor paket fmt untuk keperluan input dan output. Di dalam fungsi main, variabel `n` bertipe integer digunakan untuk menyimpan bilangan input, kemudian program menampilkan pesan permintaan input dan membaca nilai `n` menggunakan `fmt.Scan`. Selanjutnya, variabel `count` diinisialisasi dengan nilai 0 sebagai penghitung jumlah digit. Program menggunakan perulangan `for` tanpa kondisi (infinite loop) yang di dalamnya nilai `n` dibagi 10 setiap iterasi untuk menghilangkan satu digit terakhir, lalu `count` ditambah satu. Jika setelah pembagian nilai `n` menjadi 0, perulangan dihentikan dengan perintah `break`. Setelah perulangan selesai, program menampilkan nilai `count` yang merepresentasikan jumlah digit dari bilangan bulat positif yang dimasukkan.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var x float64

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

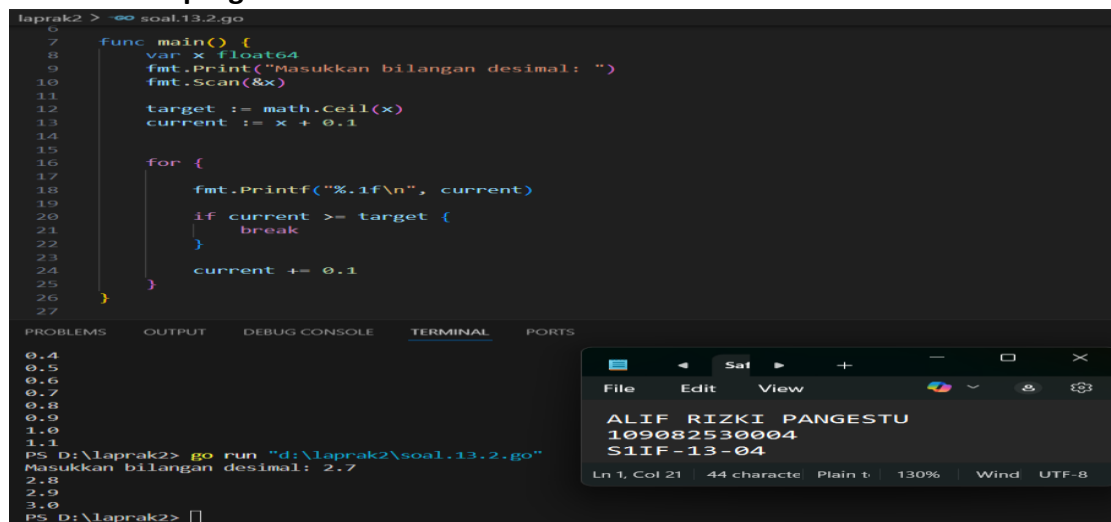
    target := math.Ceil(x)
    current := x + 0.1

    for {
        fmt.Printf("%.1f\n", current)

        if current >= target {
            break
        }

        current += 0.1
    }
}
```

Screenshoot program



```
laprak2 > go run soal.13.2.go
0
1  func main() {
2      var x float64
3      fmt.Print("Masukkan bilangan desimal: ")
4      fmt.Scan(&x)
5
6      target := math.Ceil(x)
7      current := x + 0.1
8
9      for {
10         fmt.Printf("%.1f\n", current)
11
12         if current >= target {
13             break
14         }
15
16         current += 0.1
17     }
18 }
19
20 0.4
21 0.5
22 0.6
23 0.7
24 0.8
25 0.9
26 1.0
27 1.1
28 ps D:\laprak2> go run "d:\laprak2\soal.13.2.go"
29 Masukkan bilangan desimal: 2.7
30 2.8
31 2.9
32 3.0
33 ps D:\laprak2>
```

Deskripsi program

Program Go tersebut berfungsi untuk menampilkan deretan bilangan desimal dengan kenaikan 0,1 yang dimulai dari nilai input pengguna ditambah 0,1 hingga mencapai atau melewati bilangan bulat terdekat di atasnya. Program meminta pengguna memasukkan sebuah bilangan desimal bertipe float64, kemudian nilai batas akhir ditentukan menggunakan fungsi `math.Ceil(x)` yang menghasilkan pembulatan ke atas dari nilai input. Variabel `current` diinisialisasi dengan nilai `x + 0.1` sebagai angka awal yang akan ditampilkan. Selanjutnya, program menggunakan perulangan `for` tanpa kondisi yang setiap iterasinya mencetak nilai `current` dengan satu angka di belakang koma menggunakan `fmt.Printf("%.1f\n", current)`. Jika nilai `current` sudah lebih besar atau sama dengan `target`, perulangan dihentikan dengan perintah `break`; jika belum, nilai `current` ditambah 0,1 dan perulangan dilanjutkan. Dengan demikian, program menghasilkan urutan bilangan desimal bertahap hingga mencapai batas pembulatan ke atas dari input pengguna.

3. Tugas 3

Source code

```
package main

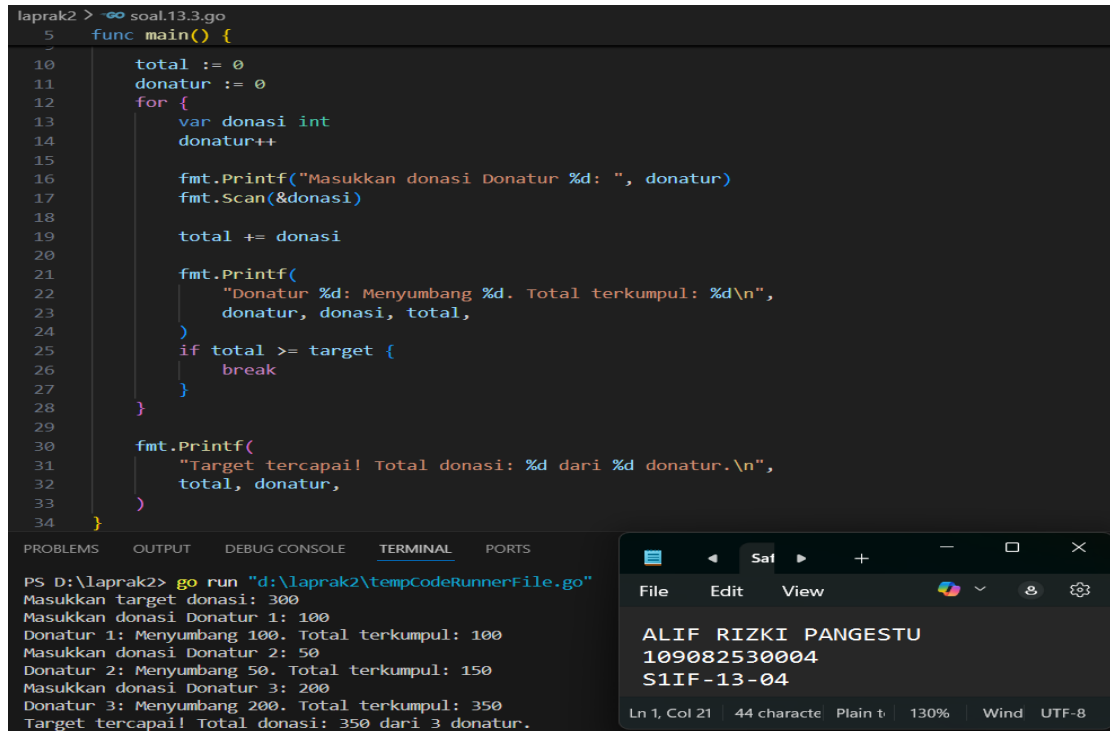
import "fmt"

func main() {
    var target int
    fmt.Print("Masukkan target donasi: ")
    fmt.Scan(&target)

    total := 0
    donatur := 0
    for {
        var donasi int
        donatur++
        fmt.Printf("Masukkan donasi Donatur %d: ", donatur)
        fmt.Scan(&donasi)
        total += donasi
        fmt.Printf(
            "Donatur %d: Menyumbang %d. Total terkumpul: %d\n",
            donatur, donasi, total,
        )
        if total >= target {
            break
        }
    }

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

Screenshoot program



```
laprak2 > go run soal.13.3.go
5 func main() {
10     total := 0
11     donatur := 0
12     for {
13         var donasi int
14         donatur++
15
16         fmt.Printf("Masukkan donasi Donatur %d: ", donatur)
17         fmt.Scan(&donasi)
18
19         total += donasi
20
21         fmt.Printf(
22             "Donatur %d: Menyumbang %d. Total terkumpul: %d\n",
23             donatur, donasi, total,
24         )
25         if total >= target {
26             break
27         }
28     }
29
30     fmt.Printf(
31         "Target tercapai! Total donasi: %d dari %d donatur.\n",
32         total, donatur,
33     )
34 }
```

PS D:\laprak2> go run "d:\laprak2\tempCodeRunnerFile.go"

Masukkan target donasi: 300
Masukkan donasi Donatur 1: 100
Donatur 1: Menyumbang 100. Total terkumpul: 100
Masukkan donasi Donatur 2: 50
Donatur 2: Menyumbang 50. Total terkumpul: 150
Masukkan donasi Donatur 3: 200
Donatur 3: Menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.

ALIF RIZKI PANGESTU
109082530004
S1IF-13-04

Ln 1, Col 21 | 44 character | Plain text | 130% | Window | UTF-8

Deskripsi program

Program Go tersebut berfungsi untuk mensimulasikan pengumpulan donasi dari beberapa donatur hingga mencapai atau melampaui target donasi yang telah ditentukan. Di dalam fungsi main, program menginisialisasi variabel total untuk menyimpan total donasi yang terkumpul dan donatur untuk menghitung jumlah donatur. Program kemudian menjalankan perulangan for tanpa kondisi, di mana pada setiap iterasi jumlah donatur ditambah satu, lalu pengguna diminta memasukkan nilai donasi dari donatur ke-n. Nilai donasi tersebut ditambahkan ke variabel total, dan program menampilkan informasi sementara berupa nomor donatur, jumlah donasi yang disumbangkan, serta total donasi yang telah terkumpul. Perulangan akan terus berjalan sampai total donasi mencapai atau melebihi nilai target, kemudian dihentikan dengan perintah break. Setelah perulangan selesai, program menampilkan pesan akhir yang menyatakan bahwa target donasi telah tercapai, disertai total donasi yang terkumpul dan jumlah donatur yang berpartisipasi.