

# **LAPORAN PRA KTIKUM**

**Algoritma Pemrograman**

**MODUL 13**

**Repeat-until**



**Disusun oleh:**

**MUHAMMAD TETUKO KEMAL PASHA**

**109082500181**

**S1IF-13-04**

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## LATIHAN KELAS – GUIDED

### 1. Guided 1

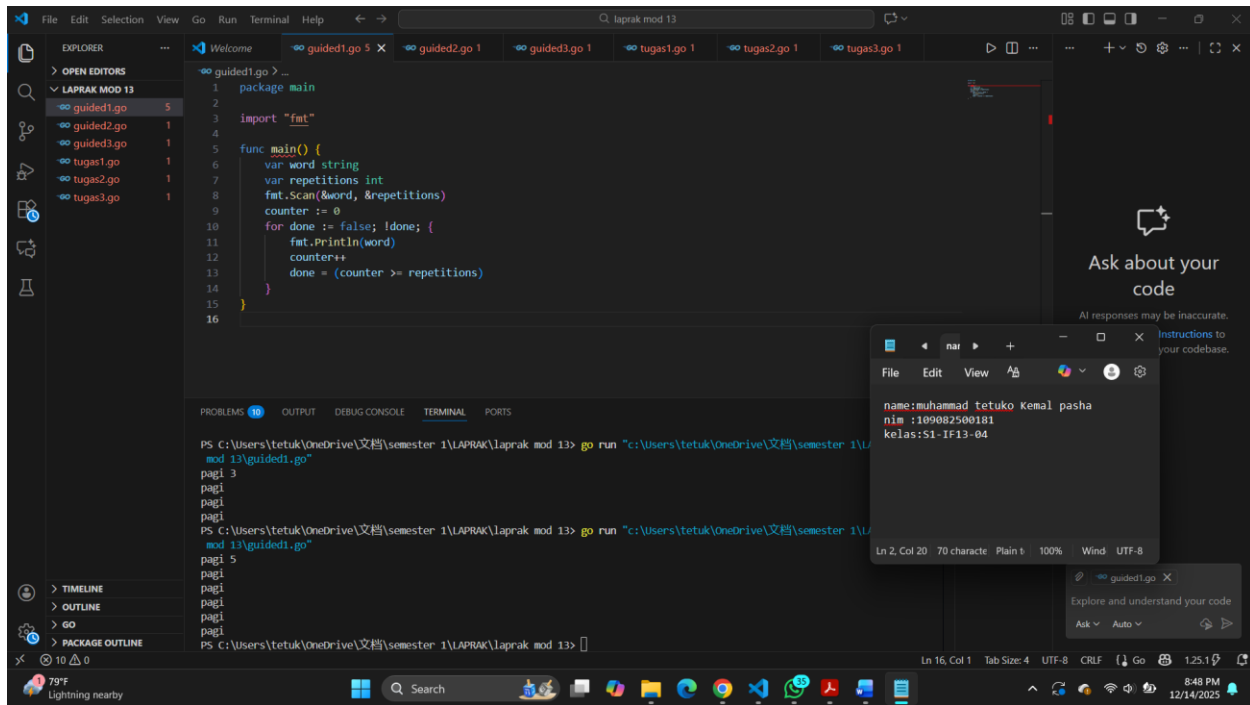
#### Source Code

```
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)
    }
}
```

## Screenshot program



## Deskripsi program

Berdasarkan gambar tangkapan layar, program Go yang sedang ditampilkan adalah *Guided 1* dari Laporan Praktikum Modul 13. Program ini dirancang untuk melakukan perulangan bertipe *repeat-until* yang diimplementasikan menggunakan *loop* for dengan kondisi berhenti yang dikontrol oleh variabel boolean *done*. Program ini menerima dua input dari pengguna: sebuah kata (*word*) dan jumlah pengulangan (*repetitions*). Di dalam *loop*, kata tersebut dicetak (`fmt.Println(word)`), penghitung (*counter*) dinaikkan, dan variabel *done* diatur menjadi *true* ketika *counter* mencapai atau melebihi *repetitions* (`done = (counter >= repetitions)`), memastikan kata tersebut dicetak sebanyak jumlah yang ditentukan.

## 2. guided 2

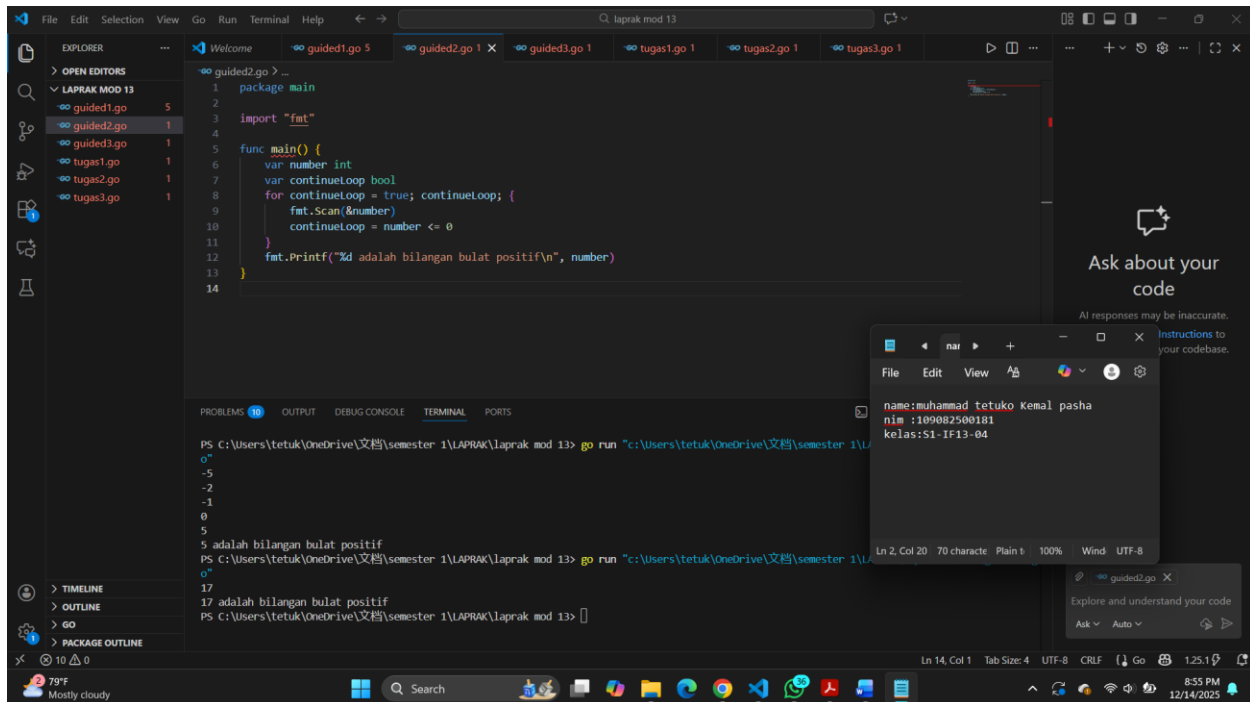
### Source code

```
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)
}
```

## Screenshot program



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

```
PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13> go run ".\guided1.go"
5
-5
-2
-1
0
5
5 adalah bilangan bulat positif
17 adalah bilangan bulat positif
PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13>
```

```
name:muhammad tetuko Kemal pasha
nim :109082500181
kelas:SI-IF13-04
```

## Deskripsi program

Berdasarkan gambar tangkapan layar, program Go yang sedang ditampilkan adalah *Guided 1* dari Laporan Praktikum Modul 13. Program ini dirancang untuk melakukan perulangan bertipe *repeat-until* yang diimplementasikan menggunakan *loop* *for* dengan kondisi berhenti yang dikontrol oleh variabel boolean *done*. Program ini menerima dua input dari pengguna: sebuah kata (*word*) dan jumlah pengulangan (*repetitions*). Di dalam *loop*, kata tersebut dicetak (`fmt.Println(word)`), penghitung (*counter*) dinaikkan, dan variabel *done* diatur menjadi *true* ketika *counter* mencapai atau melebihi *repetitions* (`done = (counter >= repetitions)`), memastikan kata tersebut dicetak sebanyak jumlah yang ditentukan.

### 3.guided 3

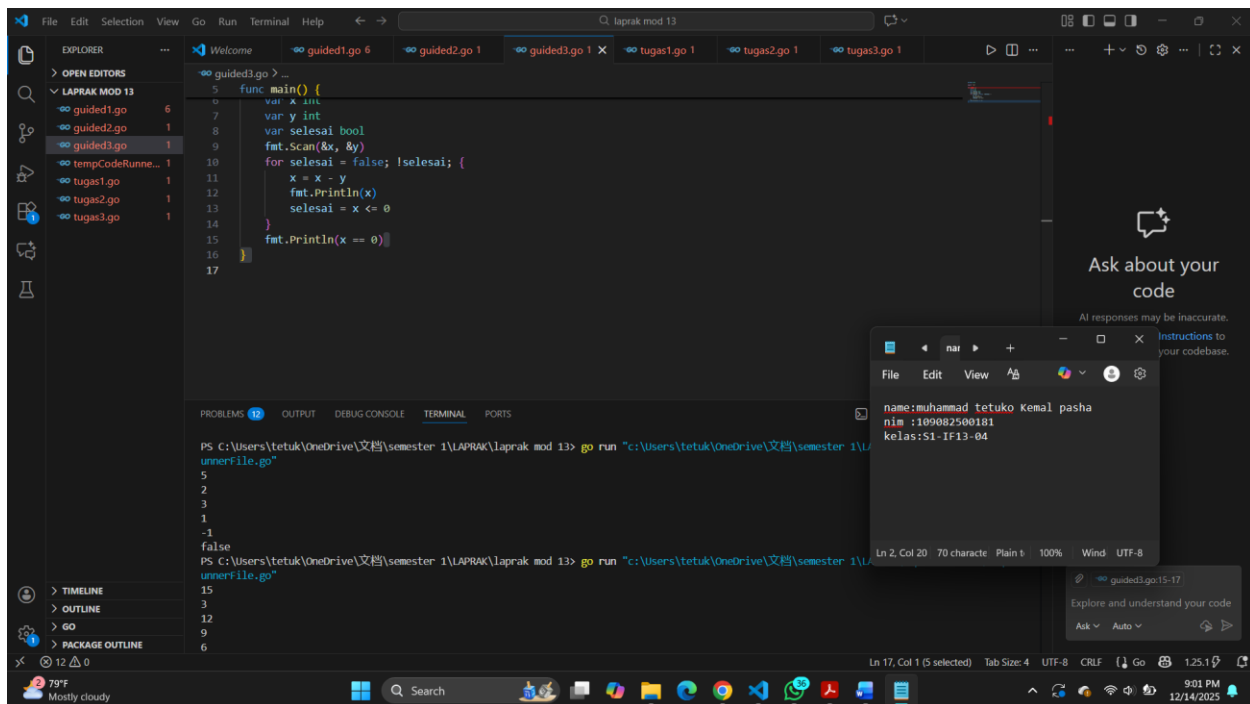
#### Source code

```
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)
}
```

## Screenshot program



## Deskripsi program

Berdasarkan gambar tangkapan layar, ketiga program Go yang ditampilkan (Guided 1, Guided 2, dan Guided 3) semuanya mengilustrasikan implementasi perulangan REPEAT-UNTIL menggunakan struktur for pada bahasa pemrograman Go. Guided 1 mencetak sebuah kata berulang kali sampai jumlah cetakan mencapai target yang ditentukan, menggunakan variabel boolean done untuk mengontrol kondisi berhenti. Guided 2 terus-menerus meminta input angka dari pengguna dan akan mengulang permintaan selama angka yang dimasukkan kurang dari atau sama dengan nol, baru berhenti setelah mendapatkan bilangan bulat positif. Sementara itu, Guided 3 melakukan pengurangan berulang pada variabel  $x$  oleh variabel  $y$  hingga  $x \leq 0$ , yang kondisi berhentinya juga diatur oleh variabel boolean selesai. Secara keseluruhan, ketiga contoh ini menunjukkan bagaimana konsep *repeat-until* — di mana aksi dieksekusi minimal satu kali dan perulangan berlanjut *sampai* kondisi terpenuhi — diimplementasikan dalam Go.

## 1. tugas 1

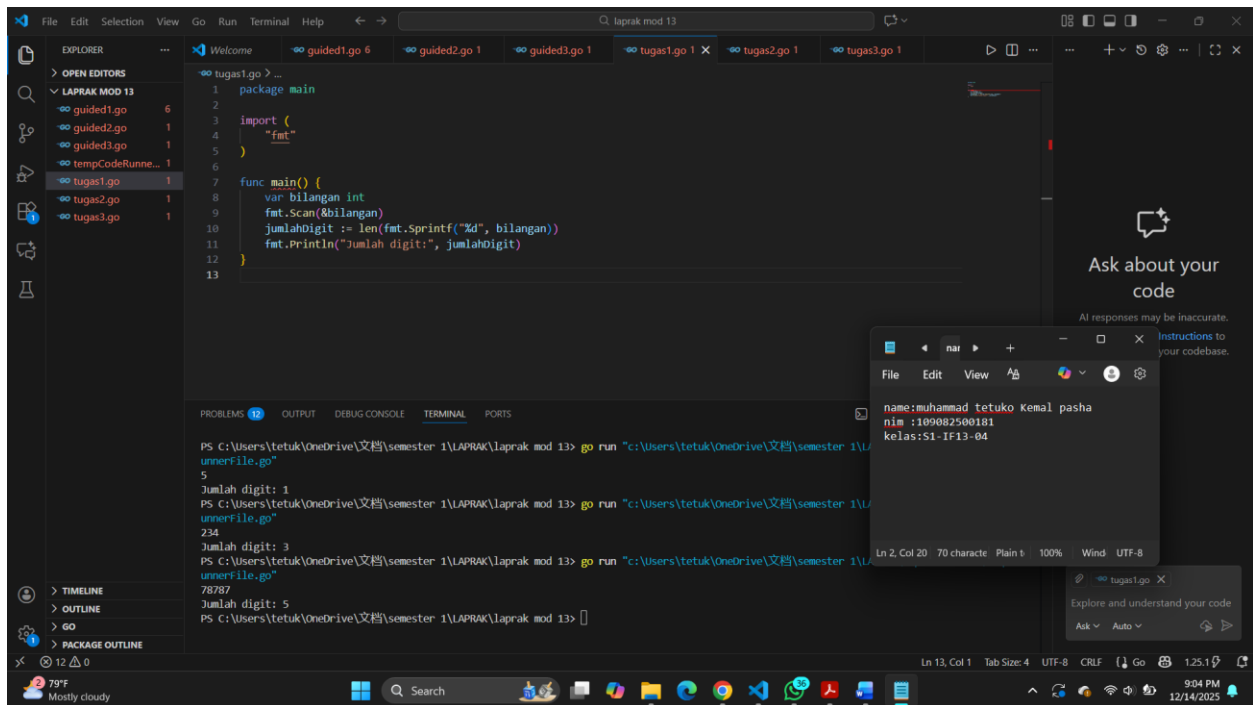
### Source code

```
package main

import (
    "fmt"
)

func main() {
    var bilangan int
    fmt.Scan(&bilangan)
    jumlahDigit := len(fmt.Sprintf("%d", bilangan))
    fmt.Println("Jumlah digit:", jumlahDigit)
}
```

### Screenshot program





## Deskripsi program

Berdasarkan kode dan *screenshot* yang tersedia (Guided 1, Guided 2, Guided 3, dan Tugas 1), program-program ini mengilustrasikan berbagai implementasi perulangan dalam bahasa Go. Guided 1 menunjukkan perulangan *repeat-until* yang mencetak sebuah kata berulang kali hingga mencapai jumlah pengulangan yang ditentukan, dikontrol oleh variabel boolean `done`. Guided 2 mengimplementasikan perulangan yang akan terus meminta input angka dari pengguna dan hanya akan berhenti ketika pengguna memasukkan bilangan bulat positif, menggunakan variabel `continueLoop` sebagai kondisi terminasi. Guided 3 menggunakan perulangan untuk melakukan pengurangan berulang ( $x = x - y$ ) dan berhenti saat  $x \leq 0$ , yang menguji apakah  $x$  habis dibagi oleh  $y$ . Sementara itu, Tugas 1 menunjukkan cara menghitung jumlah digit sebuah bilangan bulat dengan mengubah bilangan tersebut menjadi *string* lalu menghitung panjangnya, tanpa menggunakan konsep perulangan *repeat-until* yang merupakan fokus modul ini. Secara keseluruhan, program-program ini berfungsi sebagai latihan dasar untuk memahami mekanisme kontrol alur perulangan.

## 2. tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var x float64

    fmt.Scan(&x)

    batasAtas := float64(int(x)) + 1

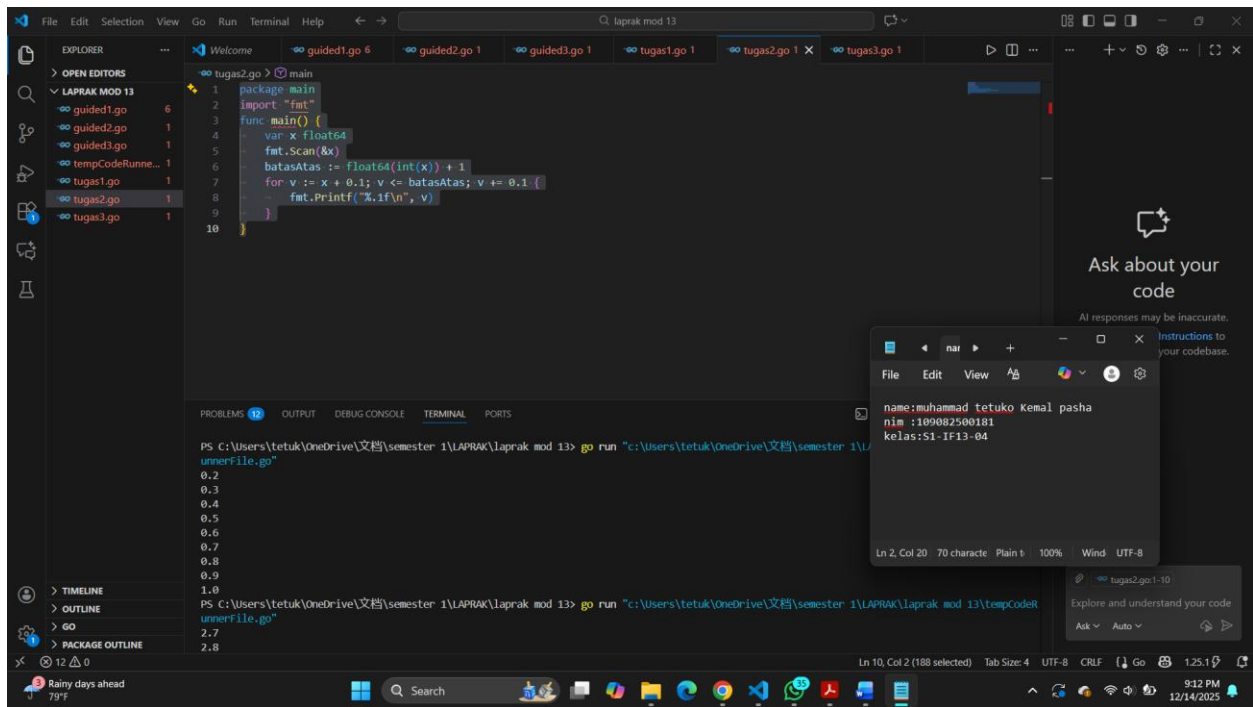
    for v := x + 0.1; v <= batasAtas; v += 0.1 {

        fmt.Printf("%.1f\n", v)

    }

}
```

### Screenshot program



## Deskripsi program

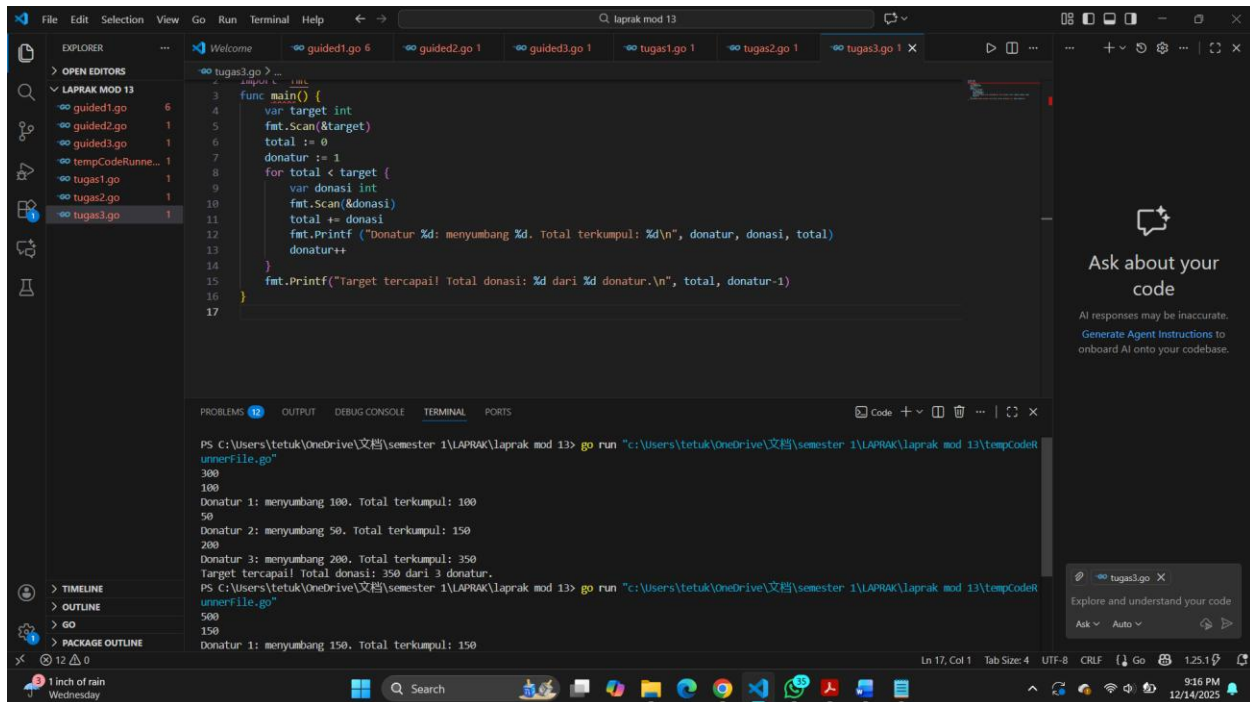
Kelima tangkapan layar ini menampilkan kode program Go dan hasil eksekusinya dari Modul 13 tentang perulangan. Program-program ini meliputi Guided 1 (mencetak kata 'pagi' berulang kali dengan jumlah pengulangan 5), Guided 2 (terus meminta input angka hingga mendapatkan bilangan bulat positif, seperti 5, -1, 0, lalu 17), Guided 3 (melakukan pengurangan berulang ( $x = x - y$ ) dan menampilkan hasil  $x == 0$  untuk menguji pembagian), Tugas 1 (menghitung jumlah digit sebuah bilangan dengan mengubahnya menjadi *string*), dan Tugas 2 (mencetak deretan angka desimal mulai dari  $x + 0.1$  hingga batas atas terdekat dengan kenaikan 0.1). Secara keseluruhan, file-file ini mendokumentasikan praktik implementasi perulangan, khususnya pola *repeat-until*, dan manipulasi data dasar menggunakan bahasa pemrograman Go.

### 3. tugas 3

#### Source code

```
package main
import "fmt"
func main() {
    var target int
    fmt.Scan(&target)
    total := 0
    donatur := 1
    for total < target {
        var donasi int
        fmt.Scan(&donasi)
        total += donasi
        fmt.Printf ("Donatur %d: menyumbang %d. Total terkumpul:
%d\n", donatur, donasi, total)
        donatur++
    }
    fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n",
total, donatur-1)
}
```

## Screenshot program



The screenshot shows a Go program in VS Code. The Explorer pane on the left shows a project named 'LAPRAK MOD 13' with files 'guided1.go', 'guided2.go', 'guided3.go', 'tugas1.go', 'tugas2.go', and 'tugas3.go'. The main editor displays the code for 'tugas3.go', which implements a loop for collecting donations. The code is as follows:

```
1 package main
2 import "fmt"
3 func main() {
4     var target int
5     fmt.Scan(&target)
6     total := 0
7     donatur := 1
8     for total < target {
9         var donasi int
10        fmt.Scan(&donasi)
11        total += donasi
12        fmt.Printf("Donatur %d: menyumbang %d. Total terkumpul: %d\n", donatur, donasi, total)
13        donatur++
14    }
15    fmt.Printf("Target tercapai Total donasi: %d dari %d donatur.\n", total, donatur-1)
16 }
17
```

The Output pane at the bottom shows the execution results of the program:

```
PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13> go run "c:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13\tempCodeRunnerFile.go"
300
100
Donatur 1: menyumbang 100. Total terkumpul: 100
50
Donatur 2: menyumbang 50. Total terkumpul: 150
200
Donatur 3: menyumbang 200. Total terkumpul: 350
Target tercapai Total donasi: 350 dari 3 donatur.
PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13> go run "c:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13\tempCodeRunnerFile.go"
500
150
Donatur 1: menyumbang 150. Total terkumpul: 150
```

## Deskripsi program

Berdasarkan kumpulan *screenshot* kode Go dan hasil eksekusinya, program-program ini mengilustrasikan berbagai penggunaan perulangan. Guided 1 dan Guided 2 menunjukkan implementasi pola *repeat-until* menggunakan perulangan `for` di Go; Guided 1 mengulang pencetakan kata, sementara Guided 2 mengulang permintaan input hingga mendapatkan bilangan bulat positif. Guided 3 menggunakan perulangan untuk melakukan pengurangan berulang ( $\$x = x - y\$$ ) dan berhenti ketika  $\$x \leq 0\$$ , yang pada dasarnya menguji sisa pembagian. Untuk bagian Tugas, Tugas 1 menghitung jumlah digit dengan mengubah bilangan menjadi *string* dan mengukur panjangnya, Tugas 2 mencetak deretan angka *float* dengan kenaikan 0.1 dari  $\$x + 0.1\$$  hingga batas atas, dan Tugas 3 mensimulasikan pengumpulan donasi berulang hingga total mencapai atau melebihi target. Secara keseluruhan, semua kode ini berfokus pada kontrol alur dan manipulasi data dasar menggunakan perulangan `for` di Go.