

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 13  
REPEAT-UNTIL**



**Disusun oleh:**

**Anindya Rahadita Yumnaa**

**109082500138**

**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 word string
    var repetitions int

    fmt.Scan(&word, &repetitions)

    counter := 0

    for done := false; !done; {

        fmt.Println(word)

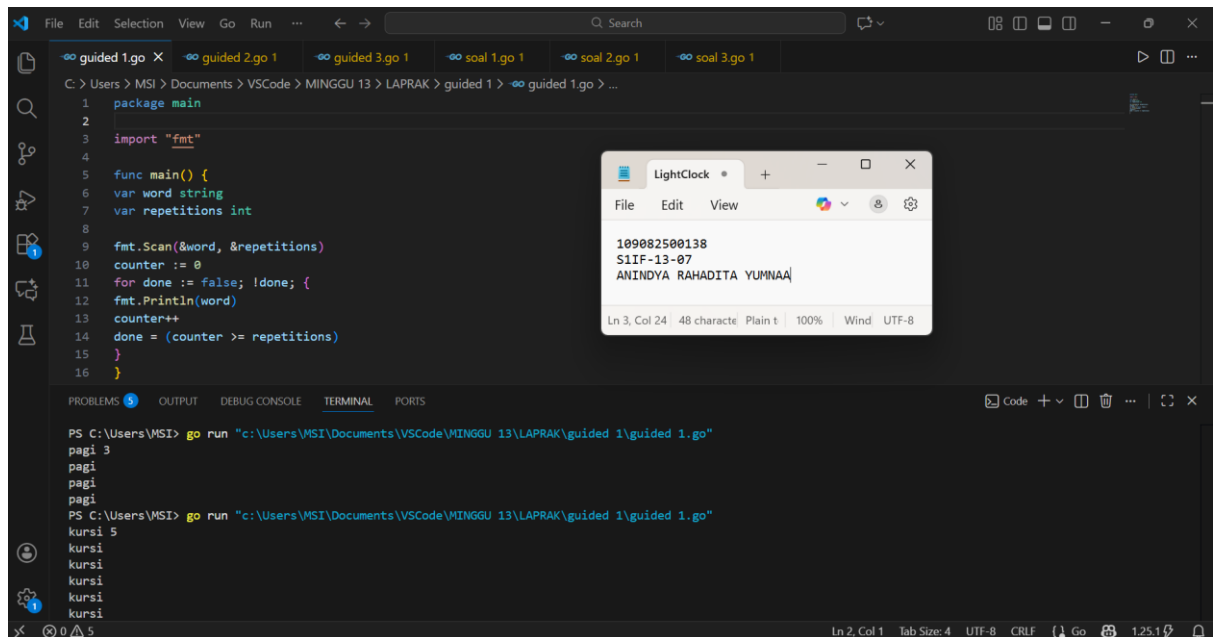
        counter++

        done = (counter >= repetitions)

    }

}
```

#### Screenshoot program



### Deskripsi program

Program ini digunakan untuk menerima sebuah kata (string) dan sebuah bilangan bulat (integer) sebagai input dari pengguna, di mana bilangan bulat tersebut menentukan jumlah pengulangan. Sebuah perulangan yang akan mencetak kata yang diinputkan secara berulang-ulang. Program menggunakan sebuah variabel penghitung (counter) yang terus bertambah di setiap putaran loop, dan perulangan akan dihentikan ketika penghitung tersebut telah mencapai atau melebihi jumlah pengulangan yang diminta, memastikan kata tersebut dicetak tepat sesuai jumlah yang diinginkan.

## 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

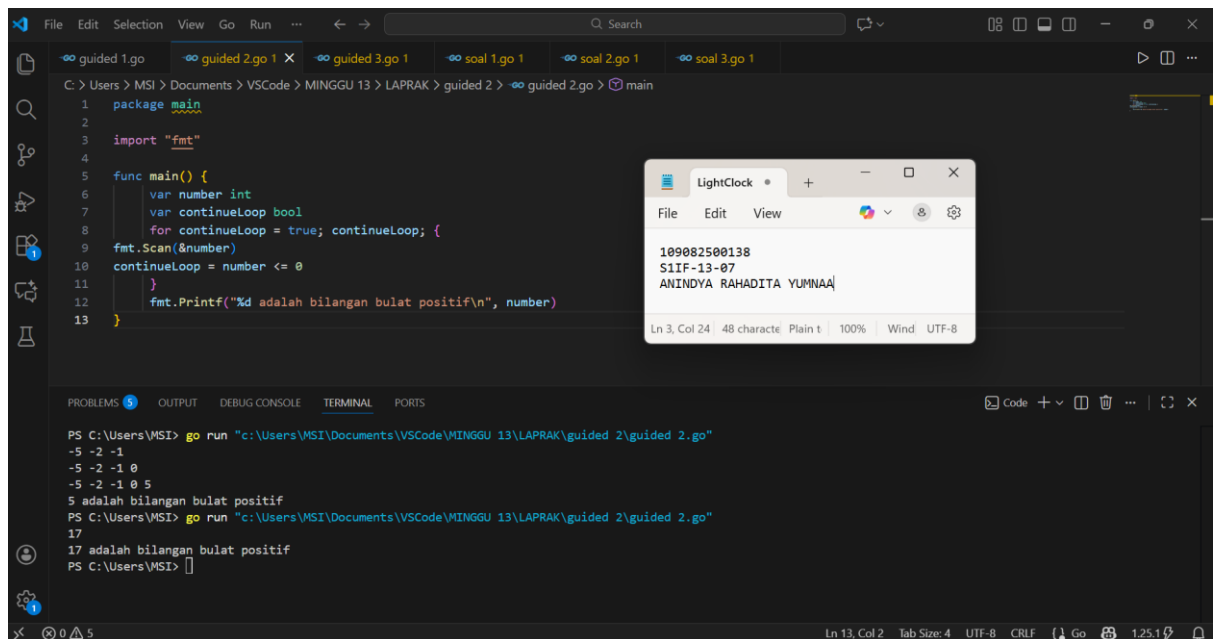
    }

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

}

```

## Screenshoot program



## Deskripsi program

Program ini digunakan untuk menampilkan bilangan bulat positif, melalui mekanisme perulangan bersyarat. Program menginisialisasi variabel **continueLoop** menjadi true untuk memastikan perulangan (for) berjalan. Di setiap putaran, program akan meminta dan menerima input bilangan bulat. Perulangan akan terus diulang selama nilai input tersebut adalah nol atau bilangan negatif. Begitu memasukkan bilangan bulat yang lebih besar dari nol (positif), kondisi loop berubah menjadi false, perulangan dihentikan, dan program kemudian mencetak bilangan positif terakhir yang berhasil didapatkan sebagai output akhir.

## 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

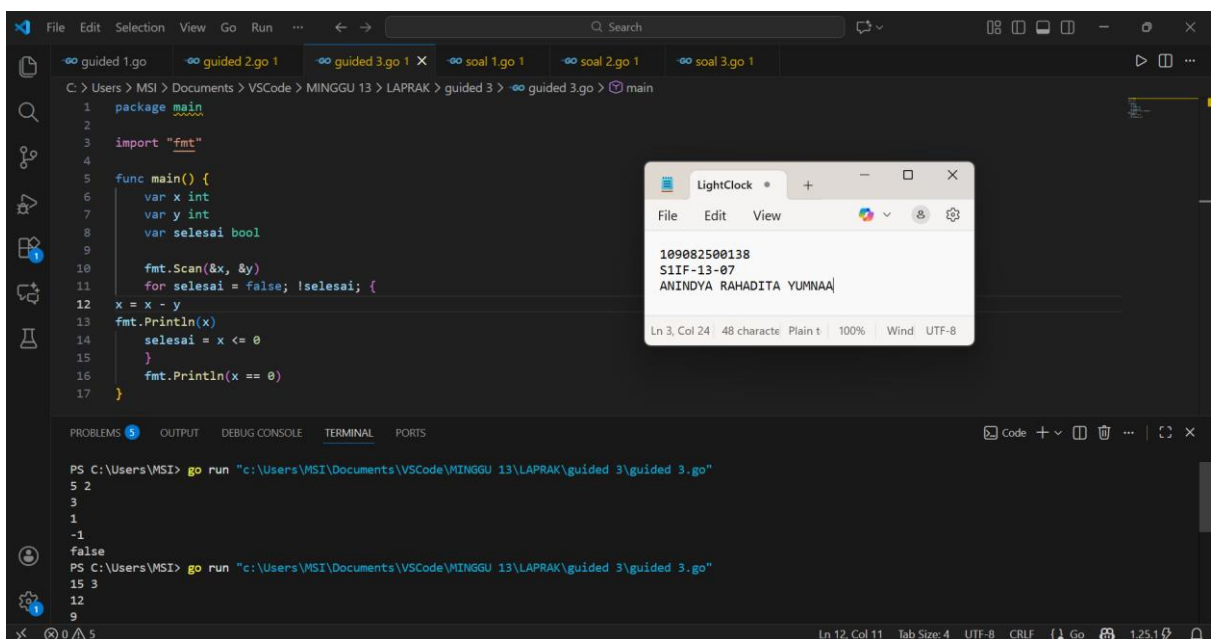
    }

    fmt.Println(x == 0)

}

```

## Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x int
7     var y int
8
9     // Read input
10    fmt.Scan(&x, &y)
11
12    // Check if y is 0
13    if y == 0 {
14        fmt.Println("Error: Division by zero")
15        return
16    }
17
18    // Perform division using repeated subtraction
19    var hasil int
20    for x > 0 {
21        x = x - y
22        hasil++
23    }
24
25    // Print result
26    fmt.Println("Hasil pembagian: ", hasil)
27
28    // Check if remainder is 0
29    if x == 0 {
30        fmt.Println("Sisa pembagian: 0")
31    } else {
32        fmt.Println("Sisa pembagian: ", x)
33    }
34
35    // Print 'selesai' status
36    if x == 0 {
37        fmt.Println("selesai")
38    } else {
39        fmt.Println("tidak selesai")
40    }
41}
```

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\guided 3\guided 3.go"
109082500138
13
-1
false
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\guided 3\guided 3.go"
15 3
12
9
6
3
0
true
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\guided 3\guided 3.go"
25 5
20
15
10
5
0
true
```

## Deskripsi program

Program ini merupakan implementasi dari algoritma pengurangan berulang untuk mensimulasikan proses pembagian. Program diawali dengan menerima dua input bilangan bulat (x sebagai bilangan yang dikurangi dan y sebagai pengurang) dari pengguna. Inti eksekusi terletak pada sebuah perulangan yang dikendalikan oleh **variabel boolean** (selesai). Perulangan akan terus berjalan dan secara konsisten mengurangi nilai x dengan y, sambil mencetak nilai x yang terbaru di setiap langkah, hingga kondisi  $x \leq 0$  terpenuhi. Setelah nilai x mencapai atau melewati nol, variabel kontrol selesai diatur menjadi true, yang menghentikan perulangan. Sebagai hasil akhir, program mencetak **nilai boolean** (true atau false) untuk menunjukkan apakah sisa pengurangan terakhir (nilai akhir x) sama dengan nol, yang secara efektif menguji apakah bilangan awal x habis dibagi oleh y.

## 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 > 0 {

    count++

    n = n / 10

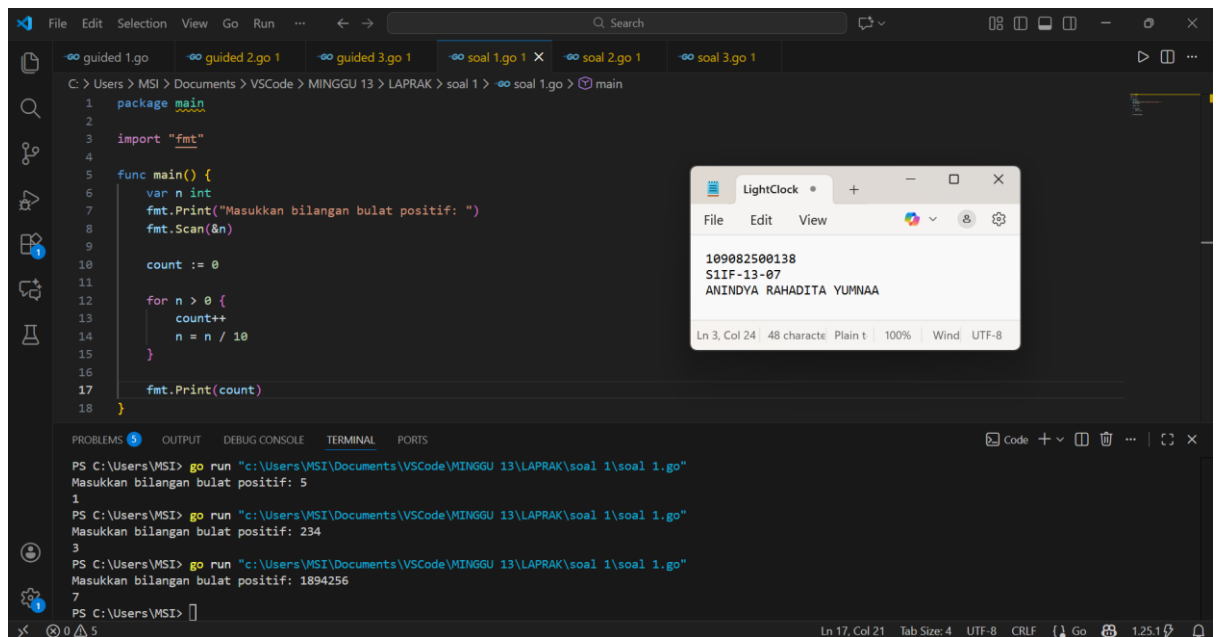
}

fmt.Print(count)

}

```

## Screenshoot program



## Deskripsi program

Program berfungsi sebagai alat untuk menghitung jumlah digit yang terdapat dalam sebuah bilangan bulat positif yang dimasukkan oleh pengguna. Setelah menerima

input dan menginisialisasi variabel penghitung (count) dengan nol, program menjalankan sebuah perulangan (for) yang berlanjut selama nilai bilangan masukan (n) masih lebih besar dari nol. Di setiap putaran loop, variabel count dinaikkan satu, yang menandakan telah diprosesnya satu digit. Secara simultan, nilai n terus diperbarui melalui operasi **pembagian integer** oleh 10 ( $n = n / 10$ ), yang secara efektif menghilangkan digit paling kanan. Perulangan akan berhenti ketika nilai n menjadi nol, menunjukkan bahwa semua digit telah dipangkas dan dihitung. Hasil akhirnya, yaitu jumlah total digit, kemudian dicetak ke konsol melalui nilai akhir variabel count.

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var x float64
    fmt.Scan(&x)

    limit := math.Ceil(x)

    for x < limit {
        x += 0.1

        x = math.Round(x*10) / 10
    }
}
```



```
        if x == limit {

            fmt.Printf("%.0f\n", x)

        } else {

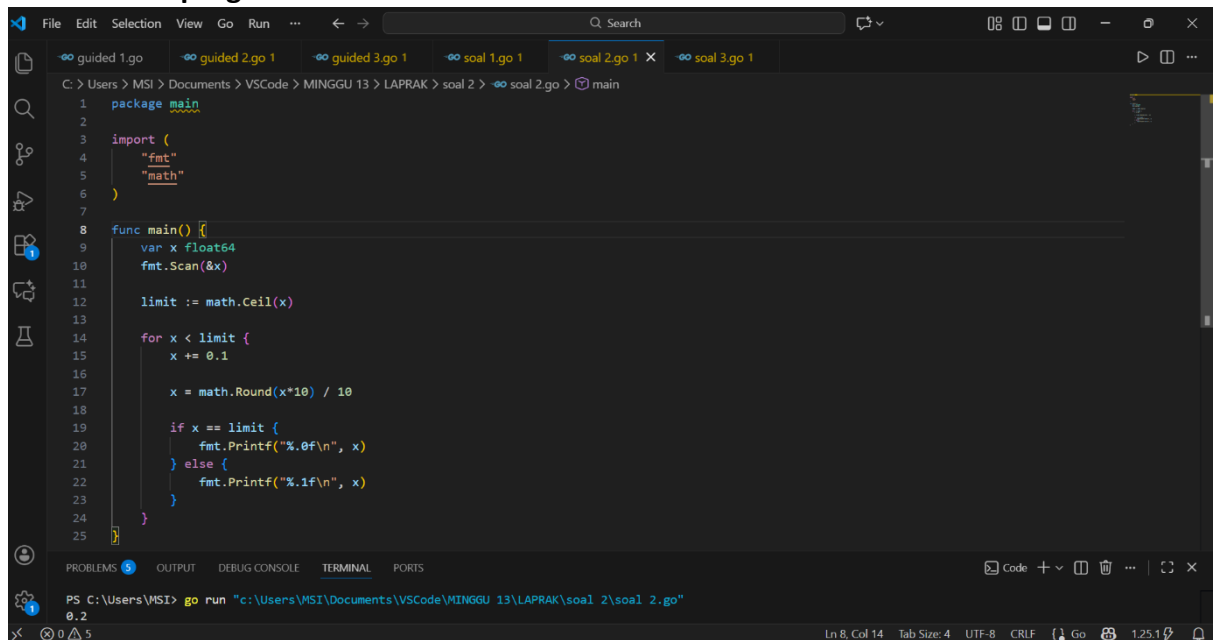
            fmt.Printf("%.1f\n", x)

        }

    }

}
```

## Screenshoot program



```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var x float64
10    fmt.Scan(&x)
11
12    limit := math.Ceil(x)
13
14    for x < limit {
15        x += 0.1
16
17        x = math.Round(x*10) / 10
18
19        if x == limit {
20            fmt.Printf("%.0f\n", x)
21        } else {
22            fmt.Printf("%.1f\n", x)
23        }
24    }
25 }
```

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 2\soal 2.go"

0.2

```
func main() {  
    x = math.Round(x*10) / 10  
  
    if x == limit {  
        fmt.Printf("%.0f\n", x)  
    } else {  
        fmt.Printf("%.1f\n", x)  
    }  
}
```

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 2\soal 2.go"  
0.2  
0.3  
0.4  
0.5  
0.6  
0.7  
0.8  
0.9  
0.9  
1  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 2\soal 2.go"  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 2\soal 2.go"  
2.7  
2.8  
2.9  
3
```

### Deskripsi program

Program Go ini berfungsi untuk membuktikan proses kenaikan bilangan desimal secara bertahap (0.1) hingga mencapai nilai pembulatan ke atasnya. Program diawali dengan menerima input float64 (x) dan menetapkan target (limit) menggunakan fungsi `math.Ceil`. Inti program adalah perulangan yang akan terus berjalan selama x kurang dari limit. Di setiap iterasi, nilai x dinaikkan sebesar 0.1, diikuti dengan langkah krusial `math.Round(x*10)/10` yang secara efektif mengoreksi kesalahan presisi floating-point dan menjaga x tetap pada presisi satu desimal. Nilai x kemudian dicetak di setiap langkah, menggunakan format satu desimal (`%.1f`), kecuali pada langkah terakhir di mana x telah sama dengan limit, yang dicetak sebagai bilangan bulat (`%.0f`). Dengan demikian, program ini menyajikan urutan bilangan desimal yang meningkat secara terkontrol dari input awal hingga bilangan bulat optimal.

### 3. Tugas 3

#### Source code

```
package main  
  
import "fmt"  
  
func main() {  
    var target int  
    fmt.Scan(&target)  
  
    total := 0
```

```

donatur := 0

for total < target {
    var donasi int
    fmt.Scan(&donasi)

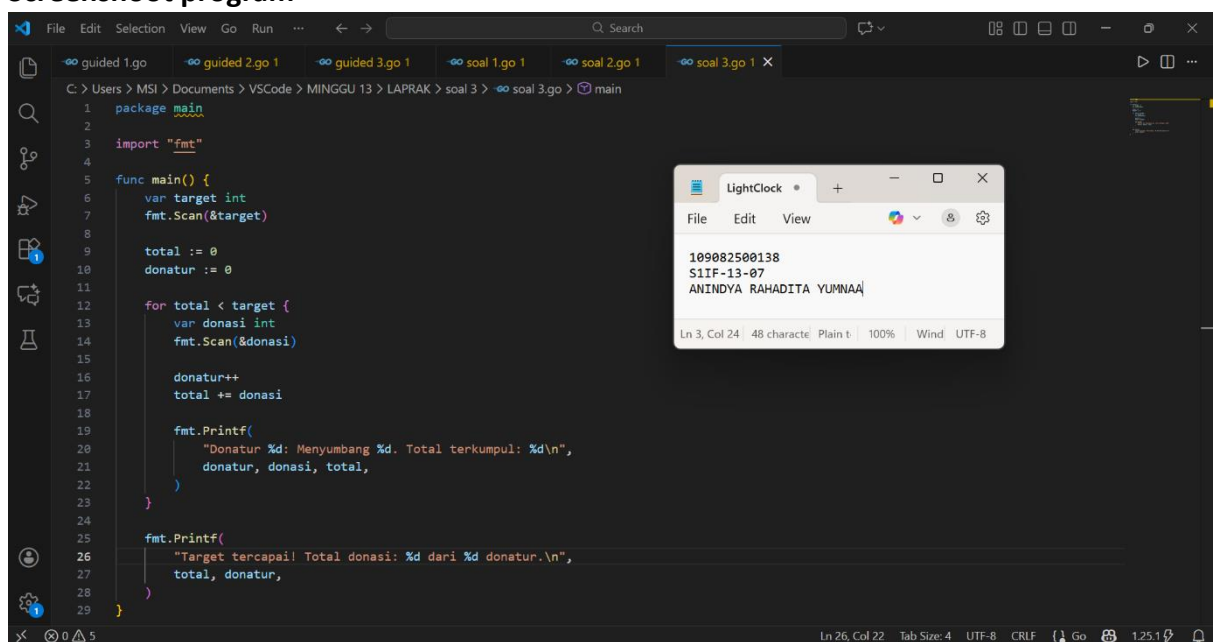
    donatur++
    total += donasi

    fmt.Printf(
        "Donatur %d: Menyumbang %d. Total
terkumpul: %d\n",
        donatur, donasi, total,
    )
}

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

```

## Screenshoot program



The screenshot shows a Go program in VS Code. The code defines a target and a loop where users input donations until the target is reached. The output shows the total donation and the number of donors.

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var target int
7     fmt.Scan(&target)
8
9     total := 0
10    donatur := 0
11
12    for total < target {
13        var donasi int
14        fmt.Scan(&donasi)
15
16        donatur++
17        total += donasi
18
19        fmt.Printf(
20            "Donatur %d: Menyumbang %d. Total terkumpul: %d\n",
21            donatur, donasi, total,
22        )
23    }
24
25    fmt.Printf(
26        "Target tercapai! Total donasi: %d dari %d donatur.\n",
27        total, donatur,
28    )
29 }

```

The output window shows the following text:

```

109082500138
$!IF-13-07
ANINDYA RAHADITA YUMNAA

```

The status bar at the bottom indicates the current line and column: Ln 26, Col 22. The file encoding is UTF-8 and the line ending is CRLF.

```
5 func main() {
20     donatur, donasi, total, terkumpul := 0, 0, 0, 0
21     for {
22         fmt.Println("Masukkan donatur ke-1: ")
23         donatur, _ = fmt.ScanInt(&donatur)
24     }
25     fmt.Println("Target tercapai! Total donasi: %d dari %d donatur.\n",
26         total, donatur,
27     )
28 }
29 }
```

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 3\soal 3.go"
300 100 50 200
Donatur 1: Menyumbang 100. Total terkumpul: 100
Donatur 2: Menyumbang 50. Total terkumpul: 150
Donatur 3: Menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 3\soal 3.go"
500 150 100 50 300
Donatur 1: Menyumbang 150. Total terkumpul: 150
Donatur 2: Menyumbang 100. Total terkumpul: 250
Donatur 3: Menyumbang 50. Total terkumpul: 300
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 3\soal 3.go"
Donatur 3: Menyumbang 50. Total terkumpul: 300
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
```

```
15 func main() {
16     donatur++
17     total += donasi
18     fmt.Println("Donatur %d: Menyumbang %d. Total terkumpul: %d\n",
19         donatur, donasi, total,
20     )
21 }
22 }
23 }
24 }
25 }
26 }
27 }
28 }
29 }
```

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 3\soal 3.go"
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 3\soal 3.go"
Target tercapai! Total donasi: 600 dari 4 donatur.
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 3\soal 3.go"
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 13\LAPRAK\soal 3\soal 3.go"
200 300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
```

## Deskripsi program

Program Go ini berfungsi sebagai simulasi penggalangan dana yang beroperasi berdasarkan target keuangan yang ditetapkan pengguna sebagai input awal. Logika program diatur oleh perulangan yang akan terus dijalankan selama total donasi yang terkumpul masih kurang dari target yang ditentukan. Di setiap iterasi loop, program menerima input sumbangan dari satu donatur, mencatat jumlah donatur yang berpartisipasi, dan menambahkan nilai sumbangan tersebut ke total akumulasi. Untuk baris pertama berupa bilangan bulat yang merupakan target donasi yang harus dicapai. Masukan pada baris berikutnya merupakan bilangan bulat yang menyatakan donasi oleh setiap donatur, masukan terus diterima hingga target tercapai. Program memberikan **feedback** instan dengan mencetak rincian sumbangan dan total terkini. Perulangan akan dihentikan secara otomatis setelah

total donasi mencapai atau melebihi target. Akhirnya, program menyajikan laporan akhir yang mengkonfirmasi pencapaian target serta mencantumkan jumlah total donasi yang berhasil dihimpun dan jumlah total donatur yang terlibat dalam proses tersebut.