

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 13  
REPEAT-UNTIL**



**Disusun oleh:**

**Manggala Patra Raditya**

**109082500179**

**S1IF-13-02**

**Asisten Praktikum**

Adithana dharma putra

Alfin Ilham Berlianto

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

**2025**

## 1. Guided 1

### Source Code

```
package main

import (

    "fmt"

)

func main() {

    var kata string

    var jumlah int

    fmt.Scan(&kata, &jumlah)

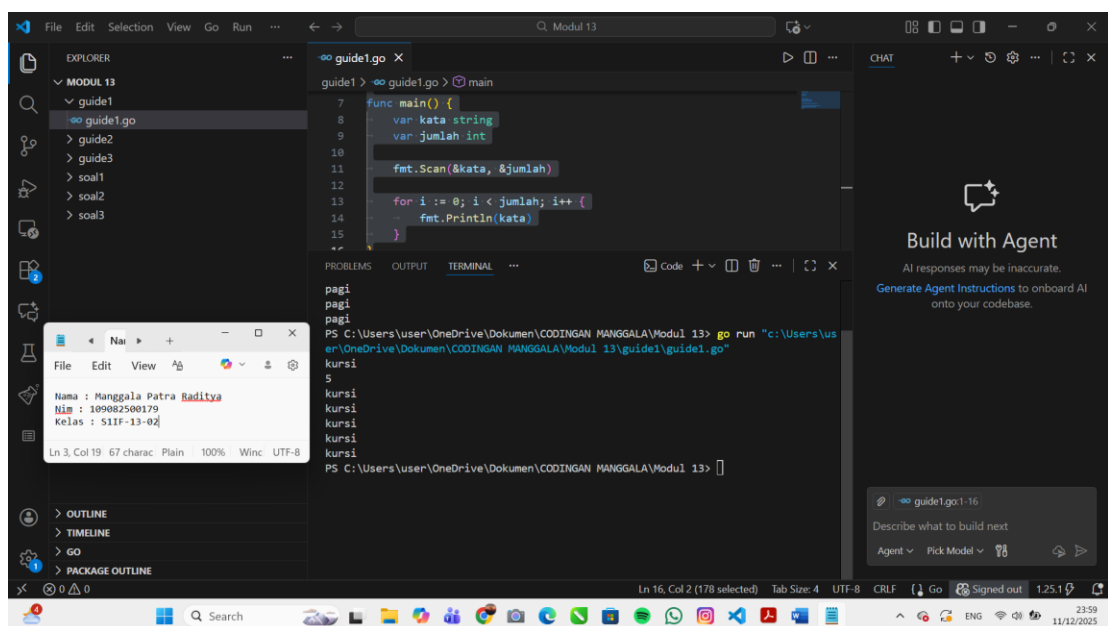
    for i := 0; i < jumlah; i++ {

        fmt.Println(kata)

    }

}
```

### Screenshoot program



## Deskripsi program

Program Go ini meminta input berupa **kata** dan **angka**, misalnya "*pagi 3*" atau "*kursi 5*". Kata pertama akan disimpan sebagai **string**, sedangkan angka kedua disimpan sebagai **integer**. Setelah menerima input tersebut, program menjalankan **perulangan for** sebanyak angka yang diberikan. Pada setiap putaran, program mencetak kembali kata tersebut.

Contohnya:

- Jika input "**pagi 3**", maka program mencetak kata "pagi" sebanyak **3 kali**.
- Jika input "**kursi 5**", maka program mencetak "kursi" sebanyak **5 kali**.

Dengan kata lain, program ini berfungsi untuk **mengulang penampilan suatu kata sesuai jumlah yang diminta pengguna**.

## 2. Guided 2

### Source Code

```
package main

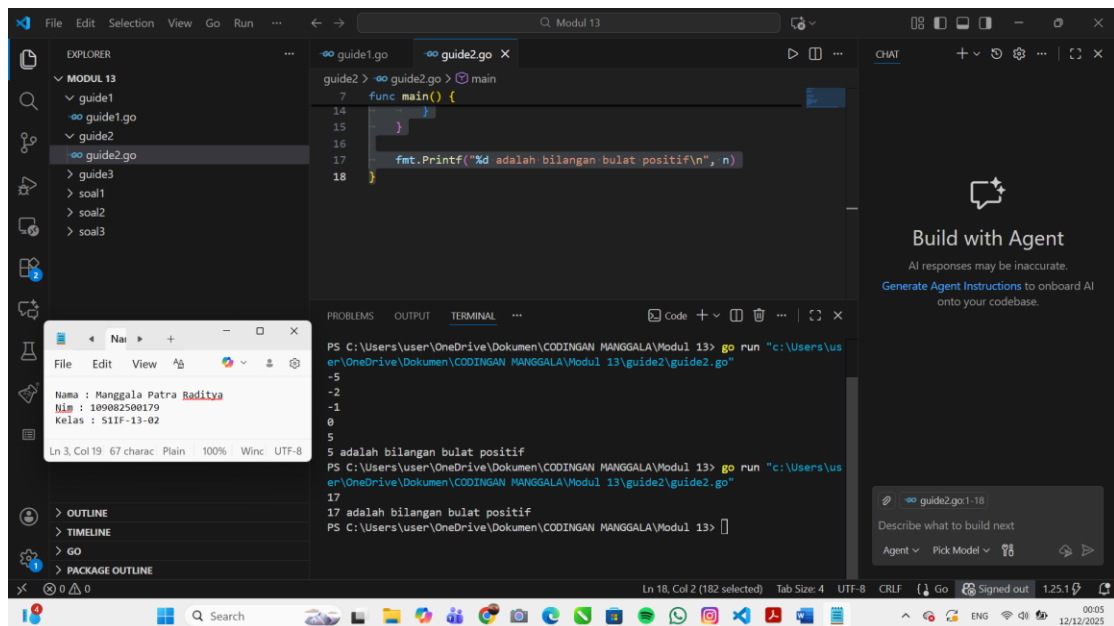
import (
    "fmt"
)

func main() {
    var n int

    for {
        fmt.Scan(&n)
        if n > 0 {
            break
        }
    }

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

### Screenshoot program



### Deskripsi program

Program ini terus meminta input angka dari pengguna hingga mendapatkan bilangan yang benar-benar positif. Jika dimasukkan angka berturut-turut seperti **-5, -2, -1, 0**, program tetap mengulang karena semuanya bukan bilangan positif, dan baru berhenti saat pengguna memasukkan **5**, sehingga muncul output **“5 adalah bilangan bulat positif”**. Jika langsung dimasukkan **17**, program langsung menerima angka tersebut dan menampilkan **“17 adalah bilangan bulat positif”** tanpa pengulangan.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    fmt.Scan(&x, &y)

    fmt.Println("Keluaran:")

    for x > 0 {

        x -= y

        fmt.Println(x)

    }

    if x == 0 {

        fmt.Println("true")

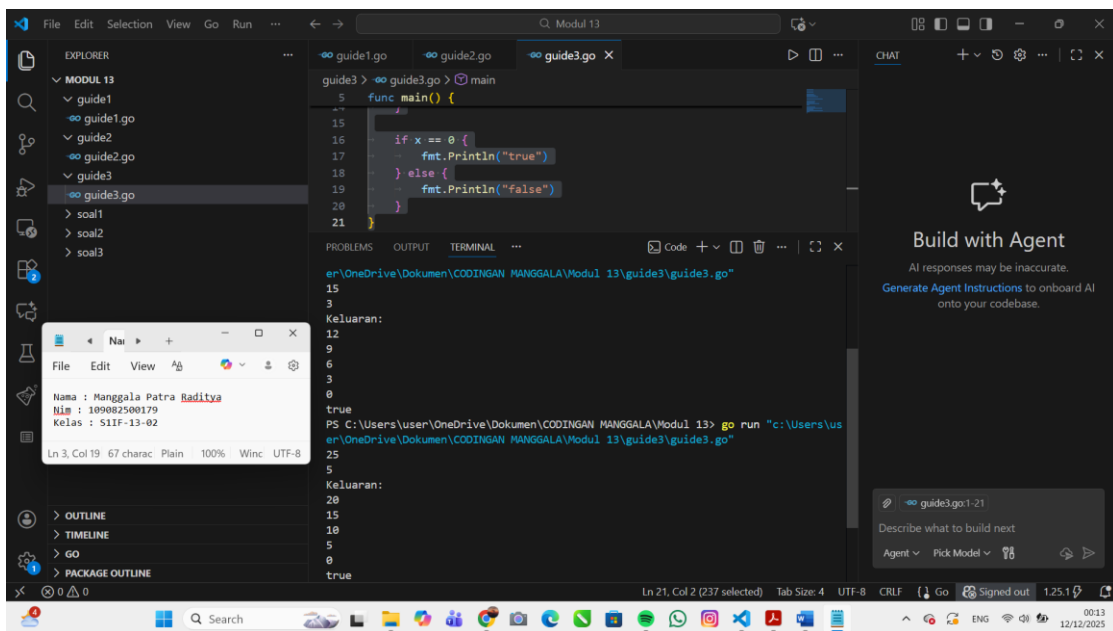
    } else {

        fmt.Println("false")

    }

}
```

## Screenshoot program



```
guide3.go:5: func main() {
15:     if x == 0 {
16:         fmt.Println("true")
17:     } else {
18:         fmt.Println("false")
19:     }
20: }
21: }
```

```
er\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 13\guide3\guide3.go
15
Keluaran:
12
9
6
3
0
true
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 13> go run "c:\Users\us
er\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 13\guide3\guide3.go"
25
5
Keluaran:
20
15
10
5
0
true
```

## Deskripsi program

Program ini membaca dua bilangan, yaitu **x** sebagai nilai awal dan **y** sebagai pengurang. Setelah menampilkan teks “**Keluaran:**”, program memasuki perulangan yang terus berjalan selama nilai **x masih lebih besar dari 0**. Pada setiap putaran, nilai **x** dikurangi dengan **y**, kemudian nilai **x** yang baru langsung dicetak. Setelah perulangan selesai—baik karena **x** menjadi 0 maupun negatif—program melakukan pengecekan: bila nilai akhir **x tepat sama dengan 0**, maka program mencetak “**true**”, tetapi bila nilai akhirnya **negatif**, program mencetak “**false**”, menandakan bahwa **x** tidak habis dibagi secara tepat oleh **y**.

Dengan masukan **5 2**, pengurangan terjadi menjadi **3, 1, -1**, sehingga loop berhenti saat **x** sudah negatif, dan hasil akhirnya **false**. Pada masukan **15 3**, **x** berkurang bertahap menjadi **12, 9, 6, 3, lalu 0**, sehingga berakhir tepat 0 dan program mencetak **true**. Sementara itu, untuk masukan **25 5**, **x** turun menjadi **20, 15, 10, 5, dan 0**, juga berakhir tepat di 0 sehingga hasilnya **true**. Program ini pada dasarnya memeriksa apakah **x** dapat dikurangi dengan **y** secara berulang hingga tepat menjadi nol.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {

    var n string

    fmt.Println("Masukkan Bilangan:")

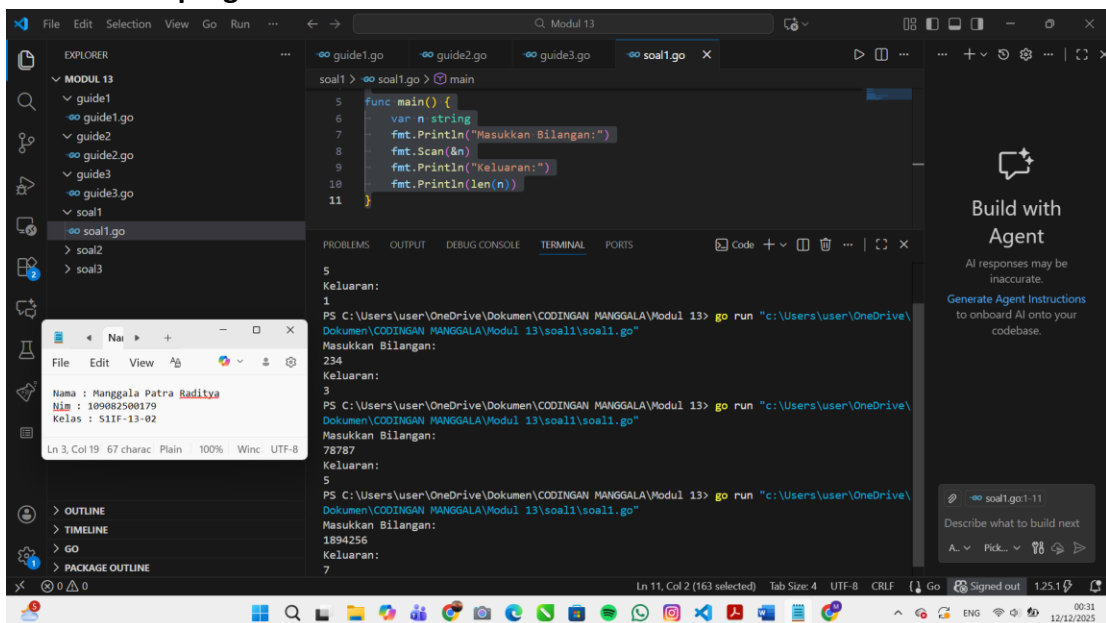
    fmt.Scan(&n)

    fmt.Println("Keluaran:")

    fmt.Println(len(n))

}
```

#### Screenshoot program





**Deskripsi program**

Program ini meminta pengguna memasukkan sebuah bilangan lalu membaca input tersebut sebagai string agar setiap digit dapat dihitung tanpa perlu konversi angka. Setelah input diterima melalui `fmt.Scan(&n)`, program menampilkan teks "Keluaran:" dan mencetak jumlah digit menggunakan `len(n)`. Dengan cara ini, masukan 5 menghasilkan 1 *digit*, 234 menghasilkan 3 *digit*, 78787 menghasilkan 5 *digit*, dan 1894256 menghasilkan 7 *digit*, karena fungsi `len` langsung menghitung banyaknya karakter yang membentuk bilangan tersebut.

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n float64

    fmt.Println("Masukkan Bilangan:")

    fmt.Scan(&n)

    fmt.Println("Keluaran:")

    batas := math.Ceil(n)

    x := n

    for x < batas {
        x = x + 0.1

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

## Screenshoot program

```
soal2.go: main
7
8 func main() {
9     var n float64
10    fmt.Println("Masukkan Bilangan:")
11    fmt.Scan(&n)
12    fmt.Println("Keluaran:")
13
14    batas := math.Ceil(n)
15    x := n
16
17    for x < batas {
18        fmt.Println(x)
19        x += 0.1
20    }
21}
```

OUTPUT:

```
Dokumen\CODINGAN MANGGALA\Modul 13\soal2\soal2.go
Masukkan Bilangan:
0.2
Keluaran:
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1

PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 13> go run "c:\Users\user\OneDrive\
Dokumen\CODINGAN MANGGALA\Modul 13\soal2\soal2.go"
Masukkan Bilangan:
2.7
Keluaran:
2.7
2.8
2.9
3.0
```

## Deskripsi program

Program Go di atas membaca sebuah bilangan desimal ke dalam variabel `n`, lalu menentukan nilai pembulatan ke atasnya menggunakan `math.Ceil` dan menyimpannya pada variabel `m`. Setelah itu, nilai awal `n` disalin ke variabel `x`, yang kemudian dinaikkan sebesar **0.1 setiap iterasi** di dalam perulangan `for x < m`. Setiap penambahan dicetak menggunakan `fmt.Printf("%.1f\n", x)`.

Ketika pengguna memasukkan **0.2**, variabel `m` menjadi **1**, sehingga perulangan menambah `x` dari 0.2 menjadi 0.3, 0.4, 0.5, dan seterusnya sampai mencapai 1.0. Begitu pula saat memasukkan **2.7**, nilai `m` menjadi **3**, sehingga program mencetak kenaikan 0.1 secara bertahap mulai dari 2.8, 2.9, hingga 3.0. Dengan demikian, keluaran yang muncul mengikuti tepat cara kerja kode: menaikkan nilai input secara bertahap sampai mencapai nilai pembulatan ke atas.

### 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

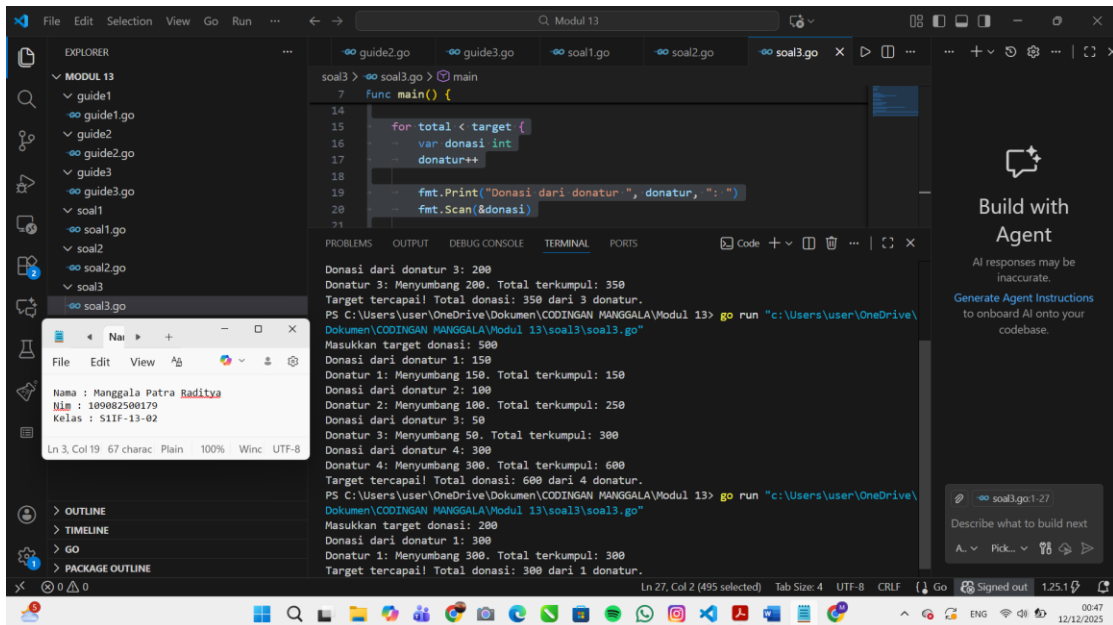
    for total < target {
        var donasi int
        donatur++

        fmt.Print("Donasi dari donatur ", donatur, ": ")
        fmt.Scan(&donasi)

        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 Explorer pane on the left shows a project structure with files `guide1.go`, `guide2.go`, `guide3.go`, `soal1.go`, `soal2.go`, and `soal3.go`. The main editor displays `soal3.go` with the following code:

```
14 func main() {
15     for total < target {
16         var donasi int
17         donatur++
18
19         fmt.Print("Donasi dari donatur ", donatur, ": ")
20         fmt.Scan(&donasi)
21     }
22 }
```

The Output pane at the bottom shows the program's execution:

```
Donasi dari donatur 3: 200
Donatur 3: Menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 13> go run "c:\Users\user\OneDrive\
Dokumen\CODINGAN MANGGALA\Modul 13\soal3\soal3.go"
Masukkan target donasi: 500
Donasi dari donatur 1: 150
Donatur 1: Menyumbang 150. Total terkumpul: 150
Donasi dari donatur 2: 100
Donatur 2: Menyumbang 100. Total terkumpul: 250
Donasi dari donatur 3: 50
Donatur 3: Menyumbang 50. Total terkumpul: 300
Donasi dari donatur 4: 300
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 13> go run "c:\Users\user\OneDrive\
Dokumen\CODINGAN MANGGALA\Modul 13\soal3\soal3.go"
Masukkan target donasi: 200
Donasi dari donatur 1: 300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
```

A small window in the foreground shows a form with the following text:

```
Nama : Manggala Patra Raditya
Nim : 109082500179
Kelas : S1IF-13-02
```

## Deskripsi program

Program ini pada dasarnya berjalan dengan cara meminta saya memasukkan target donasi terlebih dahulu, misalnya **300**. Setelah itu program mulai meminta donasi dari setiap donatur satu per satu. Setiap kali saya memasukkan angka, seperti **100**, lalu **50**, dan **200**, angka-angka tersebut langsung dijumlahkan menjadi **100 → 150 → 350**. Setiap penambahan ditampilkan oleh program lewat keluaran seperti *"Donatur 1 menyumbang 100, total 100"* dan seterusnya. Program akan terus mengulang proses ini selama total donasi masih di bawah target. Begitu total akhirnya mencapai atau melebihi target, program otomatis berhenti dan menampilkan laporan akhir bahwa target sudah tercapai beserta total donasi dan jumlah donaturnya. Dengan kata lain, program ini bekerja dengan cara menerima angka, menambahkannya ke total, menampilkan perkembangan, lalu berhenti ketika angka total sudah memenuhi target.