

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)
    }
}
```

Screenshot program

The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows a folder structure under 'MODUL 13' containing 'guide1' and 'guide2'.
- guide1.go:** The active file in the editor. It contains the Go code provided in the previous section.
- TERMINAL:** Shows the command `go run "c:\Users\user\OneDrive\...` being run, followed by the output:

```
pagi
pagi
pagi
kursi
5
kursi
kursi
kursi
kursi
kursi
kursi
```
- Build with Agent:** A feature panel on the right with the message "Build with Agent" and "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase."
- Status Bar:** Shows the current line (Ln 16, Col 2), tab size (Tab Size: 4), and date/time (11/12/2025).

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)
}
```

Screenshot program

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a project structure under "MODUL 13" with files: guide1.go, guide2.go, guide3, seal1, seal2, seal3.
- Code Editor:** The active file is "guide2.go". The code is as follows:

```
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)
}
```
- Terminal:** Shows the command "go run" being executed in the directory "C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 13". The output is:

```
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 13> go run "c:/Users/user/OneDrive/Dokumen/CODINGAN MANGGALA/Modul 13/guide2.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
```
- Bottom Status Bar:** Shows the current line (Ln 18, Col 2), tab size (Tab Size: 4), and file encoding (UTF-8).

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")

    }

}
```

Screenshot program

The screenshot shows a Go development environment with the following details:

- EXPLORER:** Shows a project structure under "MODUL 13" with files: guide1.go, guide2.go, and guide3.go, along with sub-directories guide1, guide2, and guide3, and sub-files soal1, soal2, and soal3.
- CODE EDITOR:** The active file is "guide3.go" containing the following Go code:

```
func main() {
    for {
        if x == 0 {
            fmt.Println("true")
        } else {
            fmt.Println("false")
        }
    }
}
```
- TERMINAL:** Shows the command "go run" being run from the directory "C:\Users\user\OneDrive\...". The output is:

```
Keluaran:
12
9
6
3
0
true
PS C:\Users\user\OneDrive\...> go run "c:/Users/user/OneDrive/.../guide3.go"
Keluaran:
20
15
10
5
0
true
```
- STATUS BAR:** Displays the current file path ("C:\Users\user\OneDrive\..."), tab size (4), encoding (UTF-8), and date/time (12/12/2025).

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))

}
```

Screenshot program

The screenshot shows a Go development environment with the following details:

- EXPLORER:** Shows the project structure under `MODUL 13`, including `guide1`, `guide2`, `guide3`, and `soal1` subfolders. `soal1.go` is selected.
- Code Editor:** Displays the content of `soal1.go` with syntax highlighting for Go keywords and comments.
- TERMINAL:** Shows the command `go run "c:\Users\user\OneDrive\Kodem\GO\Modul 13\soal1\soal1.go"` being run twice. The first run outputs "Masukkan Bilangan:" and "234". The second run outputs "Masukkan Bilangan:" and "1894256".
- Output Window:** Shows the user input "234" and the resulting output "Keluaran: 3". It also shows the user input "1894256" and the resulting output "Keluaran: 7".
- Build with Agent:** A sidebar feature from GitHub that suggests AI integration with the codebase.

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)
    }
}
```

Screenshot program

The screenshot shows a Windows desktop environment with a code editor window titled "Modul 13". The editor displays a Go file named "soal2.go". The code defines a main function that reads a float64 value from the user, rounds it up to the nearest integer using the math.Ceil function, and then prints the result. The terminal below the editor shows the execution of the program and its output for different input values.

```
func main() {
    var n float64
    fmt.Println("Masukkan Bilangan:")
    fmt.Scan(&n)
    fmt.Println("Keluaran:")
    batas := math.Ceil(n)
    x := n
}

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

PS C:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 13> go run "c:/Users/user/OneDrive/Documents/CODINGAN MANGGALA/Modul 13/soal2/soal2.go"
Masukkan Bilangan:
2.7
Keluaran:
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)
}
```

Screenshot program

The screenshot shows a Go code editor interface. The left sidebar displays a file tree for 'MODUL 13' containing 'guide1', 'guide2', 'guide3', 'soal1', 'soal2', and 'soal3'. The main editor window shows a Go file named 'soal3.go' with the following code:

```
func main() {
    for total < target {
        var donasi int
        donatur++
        fmt.Println("Donasi dari donatur ", donatur, ":")
        fmt.Scan(&donasi)
    }
}
```

The 'OUTPUT' tab shows the execution of the program:

```
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\Documents\CODINGAN MANGGALA\Modul 13> go run "c:/Users/user/OneDrive\Documents\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\Documents\CODINGAN MANGGALA\Modul 13> go run "c:/Users/user/OneDrive\Documents\CODINGAN MANGGALA\Modul 13\soal3\soal3.go"
Masukkan target donasi: 300
Donasi dari donatur 1: 300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
```

The status bar at the bottom indicates the current line (Ln 27, Col 2), number of selected lines (495), tab size (4), and date/time (12/12/2025).

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.