

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 13**

**REPEAT-UNTIL**



**Disusun oleh:**

**RAFFI YULIANSYAH**

**109082500101**

**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 num int
    var word string
    var kondisi bool
    fmt.Scan(&word)
    fmt.Scan(&num)
    for kondisi = false; !kondisi; {
        fmt.Println(word)
        num--
        kondisi = num == 0
    }
}
```

#### Screenshot program

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

- Code Editor:** Displays the source code for `guided1.go`. The code defines a package `main` with a `func main()` block. It reads a word from input and a number from input, then prints the word `num` times.
- Terminal:** Shows the command `go run .\guided1.go` being run twice. The first run prints `pagi` three times, and the second run prints `kursi` five times.
- Note Pad:** A separate window titled `*Untitled - Notepad` contains personal information: `109082500101`, `Raffi Yuliansyah`, and `IF-13-07`.

## Deskripsi program

Program ini adalah program untuk menampilkan sebuah kata secara berulang kali sesuai dengan jumlah yang diinputkan. Program dimulai dengan mendeklarasikan tiga variabel, yakni num yang bertipe data bilangan bulat (int), word yang bertipe data teks (string), dan kondisi yang bertipe data boolean (bool). Selanjutnya, program akan diisikan input, di mana input kata disimpan ke variabel word dan input angka disimpan ke variabel num, (`fmt.Scan(&word)` dan `fmt.Scan(&num)`). Setelah itu, program menjalankan struktur perulangan (for) yang diinisialisasi dengan kondisi bernilai false dan akan terus berjalan selama kondisi bernilai salah karena (`!kondisi`). Di dalam perulangan ini, program menampilkan isi variabel word (`fmt.Println(word)`), lalu mengurangi nilai num sebesar satu (`num--`), dan memperbarui nilai variabel kondisi menjadi hasil perbandingan apakah num sudah sama dengan 0 (`kondisi = num == 0`). Dan ketika num mencapai 0, kondisi menjadi true, (`!kondisi`) yang menyebabkan perulangan berhenti.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    var num int
```

```

for kondisi := false; !kondisi; {

    fmt.Scan(&num)

    if num > 0 {

        fmt.Println(num, " Adalah bilangan bulat
positif")

        kondisi = true

    }

}

}

```

### Screenshot program

The screenshot shows a code editor with several tabs open, including 'guided1.go', 'guided2.go', 'guided3.go', 'soal1.go', 'soal2.go', and 'soal3.go'. The 'guided2.go' tab is active, displaying the following Go code:

```

package main

import "fmt"

func main() {
    var num int
    for kondisi := false; !kondisi; {
        fmt.Scan(&num)
        if num > 0 {
            fmt.Println(num, " Adalah bilangan bulat positif")
            kondisi = true
        }
    }
}

```

To the right of the code editor, a small window titled 'Untitled - Notepad' is open, containing the following text:

```

File Edit Format View Help
109082500101
Raffi Yuliansyah
IF-13-07

```

Below the code editor, a terminal window shows the execution of the program:

```

PS D:\#Coding\Go\modul-13> go run .\guided2.go
-2
-5
0
-17
4
4 Adalah bilangan bulat positif
PS D:\#Coding\Go\modul-13> go run .\guided2.go
19
19 Adalah bilangan bulat positif
PS D:\#Coding\Go\modul-13> []

```

### Deskripsi program

Program ini adalah program untuk memvalidasi input agar pengguna memasukkan bilangan bulat positif. Program dimulai dengan mendeklarasikan satu variabel, yakni num, yang bertipe data bilangan bulat (int). Selanjutnya, program menjalankan struktur perulangan (for) yang menginisialisasi variabel bantu kondisi dengan nilai false dan akan terus berproses selama kondisi tersebut bernilai salah (!kondisi). Di dalam perulangan ini, program akan meminta input angka yang disimpan ke dalam variabel num (fmt.Scan(&num)). Setelah itu, program memeriksa input tersebut menggunakan percabangan (if); jika nilai num lebih besar dari 0, program akan menampilkan pesan bahwa angka tersebut adalah bilangan bulat positif menggunakan fmt.Println() dan mengubah nilai kondisi menjadi true. Dan tindakan pengubahan nilai kondisi menjadi true inilah yang akan menghentikan perulangan,

sedangkan jika input bukan positif, program akan terus mengulang proses permintaan input.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x)
    fmt.Scan(&y)
    for kondisi := false; !kondisi; {
        x = x - y
        fmt.Println(x)
        if x <= 0 {
            kondisi = x == 0
            fmt.Print(kondisi)
            kondisi = true
        }
    }
}
```

#### Screenshoot program

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

- Code Editor:** Displays the file `guided3.go` containing Go code. The code defines a package `main` with a `main()` function that reads two integers `x` and `y`, repeatedly subtracts `y` from `x` until `x` is zero or less, and then prints the result based on the value of `x`.
- Terminal:** Shows the command `go run .\guided3.go` being run twice in a PowerShell window. The first run with input `15 3` outputs `12  
9  
6  
3  
0`. The second run with input `25 5` outputs `20  
15  
10  
5  
0`. Both runs end with `true` on a new line.
- Notepad:** A separate window titled "Untitled - Notepad" contains the text:

```
109082500101
Raffi Yuliansyah
IF-13-07
```

## Deskripsi program

Program ini adalah program untuk melakukan pengurangan berulang nilai `x` dengan `y` serta menentukan apakah hasil akhirnya mencapai tepat angka nol atau bernilai negatif. Program dimulai dengan mendeklarasikan dua variabel, yakni `x` dan `y`, yang bertipe data bilangan bulat (`int`). Selanjutnya, program akan diisikan input, di mana angka pertama disimpan ke dalam variabel `x` dan angka kedua ke dalam variabel `y` (`fmt.Scan(&x)` dan `fmt.Scan(&y)`). Bunda itu, program menjalankan struktur perulangan (`for`) yang diinisialisasi dengan variabel `kondisi` bernilai `false` dan terus berproses selama kondisi bernilai salah. Di dalam perulangan ini, nilai `x` akan terus dikurangi dengan `y` (`x = x - y`) dan hasilnya ditampilkan baris demi baris. Jika nilai `x` sudah kurang dari atau sama dengan 0, program akan mengevaluasi apakah `x` tepat bernilai 0 (`kondisi = x == 0`) dan menampilkan hasil evaluasi boolean tersebut (mencetak `true` jika habis dibagi, atau `false` jika hasilnya negatif). Dan terakhir, variabel `kondisi` dipaksa menjadi `true` agar perulangan berhenti setelah hasil akhir tercapai.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var n, digit int
    digit = 0
    fmt.Scan(&n)
    for n > 0 {
        n = n / 10
        digit++
    }
    fmt.Println(digit)
}
```

#### Screenshot program

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

- Code Editor:** Displays a file named `soal1.go` containing the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n, digit int
7     digit = 0
8     fmt.Scan(&n)
9     for n > 0 {
10         n = n / 10
11         digit++
12     }
13     fmt.Println(digit)
14 }
```
- Terminal:** Shows the command `go run .\soal1.go` being run three times, resulting in outputs 1, 234, and 71248725473 respectively.
- Notepad:** A separate window titled "Untitled - Notepad" contains the following text:

```
File Edit Format View Help
109082500101
Raffi Yuliansyah
IF-13-07
```

## Deskripsi program

Program ini adalah program untuk menghitung jumlah digit yang terdapat dalam sebuah bilangan bulat positif. Program dimulai dengan mendeklarasikan dua variabel, yakni `n` dan `digit`, yang bertipe data bilangan bulat (`int`), di mana variabel `digit` diinisialisasi dengan nilai 0. Selanjutnya, program akan diisikan input dengan satu angka, yang mana angka tersebut akan disimpan ke dalam variabel `n` (`fmt.Scan(&n)`). Setelah itu, program akan memproses angka tersebut menggunakan struktur perulangan (`for`) yang terus berjalan selama nilai `n` lebih besar dari 0. Di dalam setiap perulangan, nilai `n` akan dibagi dengan 10 (`n = n / 10`) untuk menghilangkan digit terakhir dari bilangan tersebut, lalu variabel `digit` ditambahkan satu (`digit++`) sebagai penghitung. Dan setelah perulangan selesai (saat nilai `n` sudah habis atau menjadi 0), program akan menampilkan hasil akhir berupa jumlah digit yang tersimpan dalam variabel `digit` dengan `fmt.Println()`.

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)
```

```
func main() {  
  
    var n float64  
  
    fmt.Scan(&n)  
  
  
    batas := math.Ceil(n)  
  
  
    saatIni := int(n * 10)  
  
    nLanjutan := int(batas * 10)  
  
  
    for i := saatIni + 1; i <= nLanjutan; i++ {  
  
        if i%10 == 0 {  
  
            fmt.Println(i / 10)  
  
        } else {  
  
            fmt.Printf("%.1f\n", float64(i)/10.0)  
  
        }  
  
    }  
  
}
```

### Screenshot program

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

- Code Editor:** Displays a file named `soal2.go` with the following content:

```

7
8 func main() {
9     var n float64
10    fmt.Scan(&n)
11
12    batas := math.Ceil(n)
13
14    saatIni := int(n * 10)
15    nLanjutan := int(batas * 10)
16
17    for i := saatIni + 1; i <= nLanjutan; i++ {
18        if i%10 == 0 {
19            fmt.Println(i / 10)
20        } else {
21            fmt.Printf("%.1f\n", float64(i)/10.0)
22        }
23    }
24 }
```
- Terminal:** Shows command-line interactions:
  - `PS D:\#Coding\Go\modul-13> go run .\soal2.go` followed by a series of floating-point numbers from 0.2 to 0.9.
  - `PS D:\#Coding\Go\modul-13> go run .\soal2.go` followed by integers 2, 2.8, 3.0, and 3.
  - `PS D:\#Coding\Go\modul-13> []`
- Notepad:** A separate window titled "Untitled - Notepad" containing the text:

```

File Edit Format View Help
109082500101
Raffi Yuliansyah
IF-13-07
```

### Deskripsi program

Program ini adalah program untuk menampilkan urutan bilangan desimal dengan kenaikan 0.1, dimulai dari setelah angka input hingga mencapai bilangan bulat terdekat di atasnya (*ceiling*). Program dimulai dengan mendeklarasikan satu variabel, yakni `n`, yang bertipe data bilangan desimal (`float64`). Selanjutnya, program akan diisikan input, yang mana angka tersebut akan disimpan ke dalam variabel `n` (`fmt.Scan(&n)`). Setelah itu, program menghitung batas atas bilangan bulat menggunakan fungsi `math.Ceil(n)` dan mengonversi nilai input serta batas atas tersebut menjadi bilangan bulat dengan cara dikalikan 10 (disimpan dalam variabel `saatIni` dan `nLanjutan`) untuk mempermudah iterasi. Program kemudian menjalankan struktur perulangan (`for`) yang dimulai dari `saatIni + 1` hingga mencapai `nLanjutan`. Di dalam perulangan ini, program memeriksa apakah nilai iterasi saat ini habis dibagi 10 (`i%10 == 0`); jika ya, program menampilkan hasil pembagian sebagai bilangan bulat, namun jika tidak, program mengonversinya kembali menjadi desimal dan menampilkannya dengan format satu angka di belakang koma (`%.1f`).

### 3. Tugas 3

#### Source code

```

package main

import "fmt"

func main() {
    var target, total, donatur int
```

```

        fmt.Println("Masukkan target donasi: ")
        fmt.Scan(&target)

        total = 0
        donatur = 0

        fmt.Println("Masukkan jumlah uang untuk di
donasikan: ")

        for total <= target {
            var donasi int
            fmt.Scan(&donasi)
            donatur += 1

            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 development environment with the following components:

- Code Editor:** A main window displaying the source code of a Go program named `soal3.go`. The code implements a donation collection loop.
- Terminal:** An integrated terminal window showing the command `go run .soal3.go` and its output. The output shows the program prompting for a target donation amount (1000), then asking for individual donations from four donors (500, 255, 123, 124), and finally announcing that the target has been reached with a total of 1002 from 4 donors.
- Note Pad:** A separate window titled "Untitled - Notepad" containing the text "109082500101 Raffi Yuliansyah IF-13-07".

### **Deskripsi program**

Program ini adalah program untuk menghitung proses penggalangan dana hingga jumlah total donasi mencapai atau melebihi target yang ditentukan. Program dimulai dengan mendeklarasikan tiga variabel utama, yakni target, total, dan donatur, yang bertipe data bilangan bulat (int). Selanjutnya, program akan meminta input awal untuk menetapkan sasaran dana yang akan disimpan ke dalam variabel target (fmt.Scan(&target)), serta menginisialisasi variabel total dan donatur dengan nilai 0. Setelah itu, program akan menjalankan proses penerimaan donasi menggunakan struktur perulangan (for) yang terus berjalan selama nilai total masih kurang dari atau sama dengan nilai target. Di dalam setiap perulangan ini, program memproses input nominal donasi baru (disimpan dalam variabel lokal donasi), lalu menambahkan jumlah donatur (donatur += 1), mengakumulasikan donasi tersebut ke dalam total terkumpul (total += donasi), serta menampilkan status donasi saat ini ke layar. Dan ketika total donasi akhirnya sudah melampaui target (perulangan berhenti), program akan menampilkan pesan keberhasilan beserta ringkasan akhir total dana dan jumlah donatur yang berpartisipasi.