

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 10**

**ELSE-IF**



**Telkom  
University  
PURWOKERTO**

**Disusun oleh:**

**RAFI IMAM NASRULLAH**

**109082530010**

**S1IF-13-02**

**Asisten Praktikum**

Adithana dharma putra

Alfin Ilham Berlianto

**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 umur int
    var kk bool

    fmt.Print("masukan umur : ")
    fmt.Scan(&umur)

    fmt.Print("true/false : ")
    fmt.Scan(&kk)

    if umur >= 17 && kk == true {
        fmt.Print("bisa membuat ktp")
    }else {
        fmt.Print("belum bisa membuat ktp")
    }
}
```

## Screenshot program

The screenshot shows a Windows desktop environment. In the foreground, the Visual Studio Code (VS Code) interface is open. The Explorer sidebar on the left lists several Go source files: scal3.go, latihan1.go, modul 10, kanjut3.go, latihan1.go, main2.go, soal1.go, soal2.go, soal3.go, nyoba, UJIAN, uts, and zz. The terminal tab at the bottom of VS Code displays the command 'go run "c:\Users\rafi\_imam\Downloads\PTW\modul 10\latihan1.go"' followed by user input 'masukan umur : 17' and the output 'true/false : true bisa membuat ktp'. A separate terminal window titled 'NAMA' is also visible, showing the user's details: NAMA : Rafi Imam Nasrullah, NIM : 109082530010, and KELAS : S1IF-13-02. The taskbar at the bottom of the screen shows various pinned icons, including Microsoft Edge, File Explorer, and File History.

### Deskripsi program

Program ini meminta pengguna memasukkan umur dan menjawab apakah sudah memiliki KK (true/false). Setelah kedua data diterima, program mengecek apakah umur pengguna minimal 17 tahun dan status KK bernilai true. Jika kedua syarat terpenuhi, program menampilkan pesan bahwa pengguna bisa membuat KTP, sedangkan jika salah satu syarat tidak terpenuhi, program menampilkan bahwa pengguna belum bisa membuat KTP.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    var huruf string
```

```

fmt.Println("Masukkan huruf: ")

fmt.Scan(&huruf)

if huruf == "a" || huruf == "i" || huruf == "u" ||
huruf == "e" || huruf == "o" || huruf == "A" || huruf ==
"I" || huruf == "U" || huruf == "E" || huruf == "O"{

    fmt.Println("vokal")

} else if huruf >= "a" && huruf <= "z" || huruf >=
"A" && huruf <= "Z" {

    fmt.Println("konsonan")

} else {

    fmt.Println("bukan huruf")

}

}

```

## Screenshot program

The screenshot shows a Windows desktop environment with a Visual Studio Code (VS Code) window open. The VS Code interface includes:

- EXPLORER**: Shows a file tree with various Go files like `soal3.go`, `kanjut3.go`, `latihan1.go`, `main2.go` (which is the active file), `soal1.go`, `soal2.go`, and `soal3.go`.
- TERMINAL**: Displays the command-line output of running the program. It shows the program prompting for input ("Masukkan huruf: "), accepting a character like 'a' (labeled "vokal"), and accepting a character like 'k' (labeled "konsonan"). It also handles non-alphabetic input like '1' (labeled "bukan huruf").
- OUTPUT**: Shows standard Go build logs.
- PROBLEMS**: Shows no problems.
- STATUS BAR**: Shows the current file path as `C:\Users\rafi\_imam\Downloads\PIM\modul 10\main2.go` and other system information.

### **Deskripsi program**

Program ini meminta pengguna memasukkan satu karakter berupa huruf kemudian program mengecek apakah huruf tersebut merupakan huruf vokal, huruf konsonan, atau bukan huruf sama sekali. Jika input yang dimasukkan adalah salah satu huruf vokal (a, i, u, e, o) baik huruf kecil maupun huruf besar, program menampilkan “vokal”. Jika huruf yang dimasukkan masih berada dalam rentang alfabet (a–z atau A–Z) tetapi bukan vokal, program menampilkan “konsonan”. Namun jika karakter yang diketik bukan huruf alfabet, program akan menampilkan “bukan huruf”.

### **3. Guided 3**

#### **Source Code**

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)

    d1 := n / 1000
    d2 := (n / 100) % 10
    d3 := (n / 10) % 10
    d4 := n % 10

    membesar := (d1 < d2) && (d2 < d3) && (d3 < d4)
    mengecil := (d1 > d2) && (d2 > d3) && (d3 > d4)
```

```

        if membesar {

            fmt.Println("Digit pada bilangan", n , "terurut
membesar")

        } else if mengecil {

            fmt.Println("Digit pada bilangan", n , " terurut
mengecil")

        } else {

            fmt.Println("Digit pada bilangan", n , "tidak
terurut")

        }

    }

```

## Screenshot program

The screenshot shows a Windows desktop environment with a Visual Studio Code (VS Code) window open. The VS Code interface includes a top menu bar with File, Edit, Selection, View, Go, Run, Terminal, Help, and a status bar at the bottom. On the left is the Explorer sidebar showing a file tree under 'PIW'. The main editor area has two tabs: 'soal3.go 1' and 'kanjut3.go 5'. The code in 'kanjut3.go' is as follows:

```

package main
import "fmt"
func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)
    d1 := n / 1000
    d2 := (n / 100) % 10
    d3 := (n / 10) % 10
    d4 := n % 10
    membesar := (d1 < d2) && (d2 < d3) && (d3 < d4)
    mengecil := (d1 > d2) && (d2 > d3) && (d3 > d4)
    if membesar {
        fmt.Println("Digit pada bilangan", n , "terurut membesar")
    } else if mengecil {
        fmt.Println("Digit pada bilangan", n , " terurut mengecil")
    } else {
        fmt.Println("Digit pada bilangan", n , "tidak terurut")
    }
}

```

Below the editor is a terminal window titled 'TERMINAL' showing command-line interaction:

```

Masukkan bilangan: 2489
Digit pada bilangan 2489 terurut membesar
PS C:\Users\rafi_imam\Downloads\PIW> go run "c:\Users\rafi_imam\Downloads\PIW\modul 10\kanjut3.go"
Masukkan bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\Users\rafi_imam\Downloads\PIW> go run "c:\Users\rafi_imam\Downloads\PIW\modul 10\kanjut3.go"
Masukkan bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
PS C:\Users\rafi_imam\Downloads\PIW>

```

## Deskripsi program

Program ini meminta pengguna memasukkan sebuah bilangan empat digit, lalu program memecah bilangan tersebut menjadi digit ribuan, ratusan, puluhan, dan satuan. Setelah digit-digit itu dipisahkan, program mengecek apakah urutan angkanya tersusun menaik (dari kecil ke besar), menurun (dari besar ke kecil), atau tidak terurut. Jika setiap digit berikutnya lebih

besar dari digit sebelumnya, program menampilkan bahwa digit pada bilangan tersebut “terurut membesar”. Jika setiap digit berikutnya lebih kecil dari digit sebelumnya, program menampilkan “terurut mengecil”. Namun jika pola angka tidak konsisten naik atau turun, program menampilkan bahwa angka tersebut “tidak terurut”.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import (
    "fmt"
)

func main() {
    var totalGram int

    fmt.Print("Berat parsel (gram) : ")
    fmt.Scan(&totalGram)

    tarifPerKg := 10000

    kg := totalGram / 1000
    sisaGram := totalGram % 1000

    biayaKg := kg * tarifPerKg
    biayaSisa := 0
```

```
if totalGram > 10000 {  
  
    biayaSisa = 0  
  
} else {  
  
    if sisaGram >= 500 {  
  
        biayaSisa = sisaGram * 5  
  
    } else {  
  
        biayaSisa = sisaGram * 15  
  
    }  
  
}  
  
  
totalBiaya := biayaKg + biayaSisa  
  
  
fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisaGram)  
fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg,  
biayaSisa)  
  
fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)  
}
```

## Screenshot program

The screenshot shows a Windows desktop environment. In the foreground, a Microsoft Visual Studio Code (VS Code) window is open, displaying a Go code editor. The file being edited is 'soal1.go' located in a folder named 'modul 10'. The code calculates shipping costs based on weight in grams. A terminal window is also visible, showing the execution of the program and its output. The terminal output includes the calculation of kilograms and grams from total weight, and the resulting shipping costs.

```
func main() {
    var totalGram int
    fmt.Println("Berat parsel (gram): ")
    fmt.Scan(&totalGram)

    tarifPerKg := 10000

    kg := totalGram / 1000
    sisaGram := totalGram % 1000

    biayaKg := kg * tarifPerKg
    biayaSisa := 0

    if totalGram > 10000 {
        biayaSisa = 0
    } else {
        if sisaGram >= 500 {
            biayaSisa = sisaGram * 5
        } else {
            biayaSisa = sisaGram * 15
        }
    }
}
```

```
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\rafi_imam\Downloads\PIW> go run "c:\Users\rafi_imam\Downloads\PIW\modul 10\soal1.go"
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
```

### Deskripsi program

Program ini digunakan untuk menghitung biaya pengiriman parsel berdasarkan beratnya. Pengguna memasukkan berat parsel dalam gram, lalu program otomatis mengubahnya menjadi kilogram dan sisa gram. Setiap 1 kg dikenakan biaya tetap sebesar Rp10.000. Untuk sisa gramnya, jika berat parsel lebih dari 10.000 gram (10 kg), maka sisa gram tidak dihitung biaya tambahan. Namun jika masih di bawah 10 kg, sisa gram dikenakan tarif berbeda: bila sisa gram  $\geq 500$  maka dihitung Rp5 per gram, sedangkan jika kurang dari 500 gram dihitung Rp15 per gram. Setelah itu program menampilkan rincian berat, rincian biaya per bagian, dan total biaya pengiriman.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var nam float64
```

```
var nmk string

fmt.Println("Nilai akhir mata kuliah: ")

fmt.Scan(&nam)

if nam >= 80 {

    nmk = "A"

} else if nam >= 72.5 {

    nmk = "AB"

}

} else if nam >= 65 {

    nmk = "B"

}

} else if nam >= 57.5 {

    nmk = "BC"

}

} else if nam >= 50 {

    nmk = "C"

}

} else if nam >= 40 {

    nmk = "D"

}

} else {

    nmk = "E"

}

fmt.Println("Nilai mata kuliah: ", nmk)

}
```

## Screenshoot program

```
func main() {
    var nam string
    fmt.Println("Nilai akhir mata kuliah: ")
    fmt.Scan(&nam)
    if nam >= 80 {
        rnk = "A"
    } else if nam >= 72.5 {
        rnk = "AB"
    } else if nam >= 65 {
        rnk = "B"
    } else if nam >= 57.5 {
        rnk = "BC"
    } else if nam >= 50 {
        rnk = "C"
    } else if nam >= 40 {
        rnk = "D"
    } else {
        rnk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", rnk)
}
```

Nilai akhir mata kuliah: 93.5  
Nilai akhir mata kuliah: A  
PS C:\Users\rafi\_imam\Downloads\PTW> go run "c:/Users/rafi\_imam/Downloads/PTW/modul 10/soal2.go"  
Nilai akhir mata kuliah: 70.6  
Nilai akhir mata kuliah: B  
PS C:\Users\rafi\_imam\Downloads\PTW> go run "c:/Users/rafi\_imam/Downloads/PTW/modul 10/soal2.go"  
Nilai akhir mata kuliah: 49.5  
Nilai akhir mata kuliah: D

NAMA : Rafi Imam Nasrullah  
NIM : 109082530010  
KELAS : S1IF-13-02

## Deskripsi program

- Jika nilai diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?
- Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!
- Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

## JAWABAN

- Sebelum Program di perbaiki nilai 80.1 adalah D, Program sebelumnya tidak sesuai spesifikasi
- arena pada kondisi ini terlalu banyak if yang digunakan, mengapa demikian? karena urutan yang benar harusnya if > else if > else karena jika if lebih dari 1 maka akan tereksekusi satu blok, Cara kerja yang benar untuk program ini seharusnya ketika pengguna memasukkan nilai akhir mata kuliah, program langsung mengecek nilai tersebut dari kategori tertinggi. Jika nilai itu memenuhi syarat untuk huruf A, maka program langsung memberikan nilai A dan tidak mengecek kondisi lain. Kalau ternyata tidak memenuhi syarat A, barulah program mengecek apakah termasuk AB, lalu B, BC, C, dan seterusnya, hingga akhirnya jatuh ke E jika nilainya paling rendah. Dengan alur seperti ini, hanya satu kategori yang akan dipilih, sehingga hasilnya pasti tepat.
- Ada pada bagian Screenshoot program

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {
    var bulat, i int
    var prima bool
    var faktor int = 0

    fmt.Println("Bilangan: ")
    fmt.Scan(&bulat)

    for i = 1 ; i <= bulat ; i++{
        if bulat % i == 0 {
            fmt.Print(i, " ")
            faktor = faktor + 1
        }
    }
    fmt.Println()
    prima = faktor != 2
    fmt.Println(prima)
}
```

#### Screenshoot program

The screenshot shows a Go development environment with two windows. The top window is the code editor (VS Code) displaying the source code for 'soal3.go'. The bottom window is a terminal window showing the execution of the program.

**Code Editor (VS Code):**

```
package main

import "fmt"

func main() {
    var bulat, i int
    var prima bool
    var faktor int = 0

    fmt.Println("Bilangan: ")
    fmt.Scan(&bulat)

    for i = 1 ; i <= bulat ; i++{
        if bulat % i == 0 {
            fmt.Print(i, " ")
            faktor++
        }
    }
    fmt.Println()
    prima = faktor != 2
    fmt.Println(prima)
}
```

**Terminal:**

```
PS C:\Users\ruffi\Downloads\lptho> go run "c:\Users\ruffi\Imam\Downloads\lptho\modul_10\soal3.go"
Bilangan: 7
1 7
false
PS C:\Users\ruffi\Downloads\lptho>
```

### **Deskripsi program**

Program ini meminta pengguna memasukkan sebuah bilangan bulat, lalu program mencari semua angka yang bisa membagi habis bilangan tersebut. Caranya, program melakukan pengecekan dari angka 1 sampai angka itu sendiri, dan setiap angka yang bisa membagi bilangan tanpa sisa akan ditampilkan sebagai faktor. Setelah semua faktor ditemukan, program menghitung berapa banyak faktornya. Jika jumlah faktor bukan sama dengan 2, maka variabel prima akan bernilai true, dan jika jumlah faktornya tepat 2, maka prima bernilai false. Misalnya, jika inputnya 12, program menampilkan faktor-faktornya yaitu 1 2 3 4 6 12 dan karena jumlah faktornya lebih dari 2, hasil prima adalah true. Sedangkan untuk input 7, faktornya hanya 1 7, sehingga jumlahnya tepat 2, dan hasil prima menjadi false.