

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE-IF



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 umur int
    var kk bool
    fmt.Scan(&umur, &kk)
    if umur >= 17 && kk == true {
        fmt.Println("Bisa membuat ktp")
    } else {
        fmt.Println("Belum bisa membuat ktp")
    }
}
```

Screenshot program

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

- Editor:** Displays a file named `guided1.go` containing Go code. The code defines a `main` package with a `main` function that reads user input for age and ID card ownership, then prints whether they can make a KTP or not.
- Terminal:** Shows command-line history for running the program. It includes commands like `cd .\modul-10`, `go run .\guided1.go` with inputs 17 and 20, and outputs "Bisa membuat ktp" and "Belum bisa membuat ktp" respectively.
- Note Pad:** A window titled "Untitled - Notepad" containing personal information: 109082500101, Raffi Yuliansyah, and IF-13-07.
- Status Bar:** Shows file details like "Ln 15, Col 1", "Tab Size: 4", and encoding "UTF-8".

Deskripsi program

Program tersebut memiliki tujuan menentukan apakah seseorang bisa membuat KTP atau tidak berdasarkan umur dan kepemilikan KK.

Pada awal program akan diinisialisasikan variabel umur bertipe integer untuk menyimpan input umur (tahun) dan variabel kk bertipe data boolean untuk mewakili kepemilikan KK (true = memiliki dan false = tidak memiliki). Kemudian akan diinputkan menggunakan `fmt.Scan(&umur, &kk)` suatu bilangan untuk umur dan value true/false untuk kk, selanjutnya disimpan di masing masing variabel. Kemudian untuk menentukan dapatkah seseorang membuat KTP menggunakan if-else. Kode di dalam if dengan kondisi, apabila umur lebih besar sama dengan dari 17 DAN kk memiliki value true, maka akan ditampilkan "Bisa membuat ktp". Jika kondisi tersebut salah maka bagian else akan dijalankan yaitu menampilkan "Tidak bisa membuat ktp".

2. Guided 2

Source Code

```
package main

import (
    "fmt"
)
```

```

func main() {
    var char rune

    fmt.Scanf("%c", &char)

    if char == 'a' || char == 'A' || char == 'i' || char
    == 'I' || char == 'u' || char == 'U' || char == 'e' ||
    char == 'E' || char == 'o' || char == 'O' {

        fmt.Println("vokal")

    } else if (char >= 'A' && char <= 'Z') || (char >=
    'a' && char <= 'z') {

        fmt.Println("konsonan")

    } else {

        fmt.Println("Bukan Huruf")

    }
}

```

Screenshot program

The screenshot shows a code editor with multiple tabs open, including guided1.go, guided2.go (the current tab), guided3.go, soal1.go, soal2.go, soal3before.go, and soal3after.go. The guided2.go tab contains the provided Go code. Below the editor is a terminal window showing the execution of the program with different inputs:

```

PS D:\Coding\Go\modul-10> go run .\guided2.go
A
vokal
PS D:\Coding\Go\modul-10> go run .\guided2.go
f
konsonan
PS D:\Coding\Go\modul-10> go run .\guided2.go
1
Bukan Huruf
PS D:\Coding\Go\modul-10> go run .\guided2.go
$
Bukan Huruf
PS D:\Coding\Go\modul-10>

```

On the right side of the interface, there is a small Notepad window titled "Untitled - Notepad" displaying the following text:

```

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

```

Deskripsi program

Program tersebut memiliki tujuan menentukan input yang diberikan termasuk ke dalam vokal, konsonan, atau bukan huruf.

Pada awal program akan diinisialisasikan variabel char bertipe rune untuk menyimpan input. Kemudian akan diinputkan menggunakan fmt.Scan(&x) suatu karakter dan disimpan di variabel char. Kemudian untuk memeriksa karakter yang diinputkan termasuk vokal, konsonan, atau huruf digunakan else-if. Terdapat 3 percabangan,

1. If memiliki kondisi, apabila char berupa salah satu dari a, i, u, e, o, A, I, U, E, ATAU O, maka kondisi ini true dan menampilkan "vokal" menandakan isi dari char termasuk vokal.
2. Else if memiliki kondisi, jika char berada diantara rentang a sampai z ATAU A sampai Z, maka kondisi ini true dan akan menampilkan "konsonan", menandakan char termasuk konsonan
3. Else, apabila kedua kondisi diatas tidak terpenuhi (false), yang berarti bukan huruf, maka akan menampilkan "bukan huruf".

Setelah percabangan dilakukan, program selesai.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3, d4 int
    fmt.Print("Bilangan: ")
    fmt.Scan(&bilangan)
    d4 = bilangan % 10
    d3 = (bilangan % 100) / 10
    d2 = (bilangan % 1000) / 100
    d1 = bilangan / 1000
    if d1 < d2 && d2 < d3 && d3 < d4 {
        fmt.Print("Terurut Membesar")
    } else if d1 > d2 && d2 > d3 && d3 > d4 {
```

```

        fmt.Println("Terurut Mengecil")

    } else {

        fmt.Println("Tidak Terurut")

    }

}

```

Screenshot program

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

- Code Editor:** Displays the source code for `guided3.go`. The code reads a four-digit number, extracts its digits, and prints whether they are in ascending, descending, or random order.
- Terminal:** Shows command-line interactions with the Go runtime. It includes examples for numbers 2489, 1098, 3861, 9651, and 1234.
- Note Pad:** A separate window titled "Untitled - Notepad" contains personal information: "109082500101", "Raffi Yuliansyah", and "IF-13-07".

Deskripsi program

Program tersebut memiliki tujuan memeriksa apakah digit yang berurutan dari suatu bilangan ribuan (1000 – 9999) terurut secara menaik, menurun, atau tidak terurut (acak)

Program memiliki 5 variabel integer, diantaranya bilangan, d1, d2, d3, d4. Variabel bilangan berfungsi menampung input angka ribuan, kemudian d1, d2, d3, d4 untuk menyimpan masing-masing digit dari bilangan. Selanjutnya program akan menerima input lewat `fmt.Scan(&bilangan)`. Untuk menentukan jenis urutan perlu memisahkan tiap tiap digit dengan cara sebagai berikut.

1. Digit pertama (ribuan) akan disimpan di d1 yang didapat dari membagi bilangan dengan 1000 ($\text{bilangan}/1000$).
2. Digit kedua (ratusan) akan disimpan di d2 yang didapat dari me-modulo bilangan dengan 1000, kemudian dibagi 100 ($((\text{bilangan}\%1000)/100)$).
3. Digit ketiga (puluhan) akan disimpan di d3 yang didapat dari me-modulo bilangan dengan 1000, kemudian dibagi 10 ($((\text{bilangan}\%1000)/10)$).

4. Digit keempat (satuan) akan disimpan di d4 yang didapat dari me-modulo bilangan dengan 10 (bilangan%10)

Setelah keempat digit didapatkan, bilangan ditentukan terurut menaik atau menurun dengan membandingkan masing masing digit dalam 3 kondisi.

1. Ketika (if) $d1 < d2$ DAN $d2 < d3$ DAN $d3 < d4$, maka akan ditampilkan bahwa bilangan tersebut “Terurut menaik”.
2. Ketika (else if) $d1 > d2$ DAN $d2 > d3$ DAN $d3 > d4$, maka akan ditampilkan bilangan tersebut “Terurut menurun”.
3. Ketika kedua kondisi tidak terpenuhi (else), maka akan ditampilkan bilangan tersebut “Tidak terurut”

Kemudian program selesai.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var berat int
    fmt.Print("Berat parsel (gram) : ")
    fmt.Scan(&berat)

    fmt.Println("Detail berat : ", berat/1000, " kg", " + ",
    berat%1000, " gram")

    if berat >= 10000 {
        if berat%1000 >= 500 {
            fmt.Println("Detail Biaya : ", "Rp. ",
(berat/1000)*10000, " + ", "Rp. ", (berat%1000)*5)
    }
}
```

```
        fmt.Println("Total Biaya : ", "Rp. ", ((berat /  
1000) * 10000))  
  
    } else {  
  
        fmt.Println("Detail Biaya : ", "Rp. ",  
(berat/1000)*10000, " + ", "Rp. ", (berat%1000)*15)  
  
        fmt.Println("Total Biaya : ", "Rp. ", ((berat /  
1000) * 10000))  
  
    }  
  
} else if berat%1000 >= 500 {  
  
    fmt.Println("Detail Biaya : ", "Rp. ",  
(berat/1000)*10000, " + ", "Rp. ", (berat%1000)*5)  
  
    fmt.Println("Total Biaya : ", "Rp. ",  
((berat/1000)*10000)+((berat%1000)*5))  
  
} else {  
  
    fmt.Println("Detail Biaya : ", "Rp. ",  
(berat/1000)*10000, " + ", "Rp. ", (berat%1000)*15)  
  
    fmt.Println("Total Biaya : ", "Rp. ",  
((berat/1000)*10000)+((berat%1000)*15))  
  
}  
}
```

Screenshoot program

The screenshot shows two separate instances of a Go development environment, likely VS Code, running side-by-side.

Top Window:

- Code Editor:** The file `guided1.go` is open, containing a Go program that calculates the total price of a package based on its weight. The code uses `fmt.Println` statements to output intermediate details and a final total.
- Terminal:** The terminal window shows the command `PS D:\Coding\Go\modul-10> []` and the output of the program's execution, which includes the calculation of `Detail Biaya` and the final `Total Biaya`.
- Output:** A Notepad window titled "Untitled - Notepad" displays the generated output from the program, including the ID `109082500101`, the name `Raffi Yuliansyah`, and the reference `IF-13-07`.

Bottom Window:

- Code Editor:** The file `guided1.go` is open, showing the same Go program as the top window.
- Terminal:** The terminal window shows the command `PS D:\Coding\Go\modul-10> []` and the output of the program's execution, which includes the calculation of `Detail Biaya` and the final `Total Biaya`.
- Output:** A Notepad window titled "Untitled - Notepad" displays the generated output from the program, including the ID `109082500101`, the name `Raffi Yuliansyah`, and the reference `IF-13-07`.

Deskripsi program

Program tersebut memiliki tujuan menghitung berat dan biaya pengiriman parsel.

Pada awal program akan diinisialisasikan variabel berat bertipe data integer untuk menyimpan berat parsel yang diinput. Kemudian akan diinputkan berat parsel dalam satuan gram dengan `fmt.Scan(&berat)` dan disimpan di variabel berat. Untuk menghitung berat parsel secara spesifik maka akan ditampilkan berat dalam kg didapatkan dari `berat/1000` dan ditampilkan sisa berat dalam gram dari `berat%1000`.

Untuk menhitung biaya pengiriman dan rinciannya menggunakan else if dengan 3 kondisi sesuai dengan ketentuan.

- a. If memiliki kondisi apabila berat lebih dari sama dengan 10000 (10kg), maka biaya total untuk pengiriman hanya dikenakan biaya per 1000 gram saja ($\text{berat}/1000 * 10000$).

- b. Else if memiliki kondisi apabila sisa dari berat persatuan kg lebih dari sama dengan 500 pada kode program ($\text{berat} \% 1000 \geq 500$, maka biaya pengiriman adalah $((\text{berat}/1000)*10000) + (\text{berat}\%1000)*5$)
- c. Else, ketika semua kondisi diatas bernilai salah, yang memiliki arti sisa dari berat persatuan kg kurang dari 500 ($\text{berat} \% 1000 < 500$, maka biaya pengiriman adalah $((\text{berat}/1000)*10000) + (\text{berat}\%1000)*15$)

Program Selesai.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Print("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 = "D"
    } else {
        nmk = "E"
    }
    fmt.Println("Nilai akhir anda adalah", nmk)
}
```

```

        nmk = "C"

    } else if nam > 40 {

        nmk = "D"

    } else if nam <= 40 {

        nmk = "E"

    }

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

}

```

Screenshot program

The screenshot shows a code editor with several tabs open, including 'soal2.go'. The code in 'soal2.go' is as follows:

```

func main() {
    var nam float64
    var nmk string
    fmt.Print("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 if nam <= 40 {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

Below the code editor is a terminal window showing command-line output:

- PS D:\#Coding\Go\modul-10> go run .\soal2.go
- Nilai akhir mata kuliah: 93.5
- Nilai mata kuliah: A
- PS D:\#Coding\Go\modul-10> go run .\soal2.go
- Nilai akhir mata kuliah: 76.6
- Nilai mata kuliah: B
- PS D:\#Coding\Go\modul-10> go run .\soal2.go
- Nilai akhir mata kuliah: 49.5
- Nilai mata kuliah: D
- PS D:\#Coding\Go\modul-10>

On the right side of the screenshot, there is a powershell window titled 'Untitled - Notepad' containing the following text:

```

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

```

Jawaban :

- Apabila dijalankan sesuai dengan soal maka akan menampilkan Nilai D.
- Program tersebut memiliki kesalahan, yaitu percabangan menggunakan If-then yang mengakibatkan isi dari variabel nmk pada if then pertama (ketika nam > 80) terganti dengan if yang terakhir (ketika nam > 40), pada contoh kasusnya apabila nam = 93,5 maka yang seharusnya isi variabel nmk adalah A, tetapi yang tertampil nmk adalah D, sehingga terjadi kesalahan dari program.
Hal tersebut terjadi karena percabangan menggunakan if-then yang menyebabkan semua percabangan tereksekusi, sehingga aksi mengganti isi variabel nmk = "A" pada if then pertama tertimpa (terganti) dengan aksi if-then terakhir yang mengganti isi variabel nmk = "D".

- Seharusnya program tersebut memiliki alur sebagai berikut, yaitu program akan membuat variabel nam bertipe float64 dan nmk yang bertipe string. Setelah itu program akan menerima input nam. Kemudian nmk akan ditentukan menggunakan nam dengan percabangan else-if (dipilih karena hanya satu kondisi saja yang akan dieksekusi) sesuai nilai yang ditentukan. If nam > 80 maka nmk berisi "A", else if nam > 72,5 maka nmk berisi "AB", else if nam > 65 maka nmk berisi "B", else if nam > 57,5 maka nmk berisi "BC", else if nam > 50 maka nmk berisi "C", else if nam > 40 maka nmk berisi "D", else if nam <= 40 maka nmk berisi "E". Kemudian isi variabel nmk ditampilkan.
- c. Setelah diperbaiki, apabila diberi masukan 93,5 akan menampilkan nmk yaitu A, dengan masukan 70,6 akan menampilkan nmk yaitu B, dengan masukan 49,5 akan menampilkan D

3. Tugas 3

Source code

Before

```
package main

import "fmt"

func main() {
    var bilangan int
    fmt.Print("Bilangan : ")
    fmt.Scan(&bilangan)
    fmt.Print("Faktor : ")
    for i:=1; i<=bilangan; i++{
        if bilangan%i == 0 {
            fmt.Print(i, " ")
        }
    }
}
```

After

```
package main
```

```
import "fmt"

func main() {

    var bilangan int

    var prima bool

    prima = true

    fmt.Print("Bilangan : ")

    fmt.Scan(&bilangan)

    fmt.Print("Faktor : ")

    for i := 1; i <= bilangan; i++ {

        if bilangan%i == 0 {

            fmt.Print(i, " ")

        }

    }

    if bilangan == 0 || bilangan == 1 {

        prima = false

    }

    for i := 2; i <= bilangan-1; i++ {

        if bilangan%i == 0 {

            prima = false

            break

        }

    }

    if prima {

        fmt.Print("Prima : ", prima)

    } else {
```

```

        fmt.Println("Prima : ", prima)

    }

}

```

Screenshot program

Before

The screenshot shows the Visual Studio Code interface. The left pane displays the code for `soal3before.go`:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7     fmt.Println("Bilangan : ")
8     fmt.Scan(&bilangan)
9     fmt.Println("Faktor : ")
10    for i:=1; i<=bilangan; i++{
11        if bilangan%i == 0 {
12            fmt.Println(i, " ")
13        }
14    }
15 }

```

The right pane shows the terminal output:

```

PS D:\Coding\Go\modul-10> go run .\soal3before.go
Bilangan : 12
Faktor : 1 2 3 4 6 12
PS D:\Coding\Go\modul-10> go run .\soal3before.go
Bilangan : 7
Faktor : 1 7
PS D:\Coding\Go\modul-10>

```

A separate Notepad window titled "Untitled - Notepad" is open, showing the following text:

```

109082500101
Raffi Yuliansyah
IF-13-07

```

After

The screenshot shows the Visual Studio Code interface. The left pane displays the code for `soal3after.go`:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7     var prima bool
8     prima = true
9     fmt.Println("Bilangan : ")
10    fmt.Scan(&bilangan)
11    fmt.Println("Faktor : ")
12    for i := 1; i <= bilangan; i++ {
13        if bilangan%i == 0 {
14            fmt.Println(i, " ")
15        }
16    }
17
18    if bilangan == 0 || bilangan == 1 {
19        prima = false
20    }
21
22    for i := 2; i <= bilangan-1; i++ {
23        if bilangan%i == 0 {
24            prima = false
25            break
26        }
27    }
28
29    if prima {
30        fmt.Println("Prima : ", prima)
}

```

The right pane shows the terminal output:

```

PS D:\Coding\Go\modul-10> go run .\soal3after.go
Faktor : 1 7
PS D:\Coding\Go\modul-10>

```

A separate Notepad window titled "Untitled - Notepad" is open, showing the following text:

```

109082500101
Raffi Yuliansyah
IF-13-07

```

The screenshot shows a code editor with several tabs open, including `guided1.go`, `guided2.go`, `guided3.go`, `soal1.go`, `soal2.go`, `soal3before.go`, and `soal3after.go`. The `soal3after.go` tab is active, displaying the following Go code:

```

17 if bilangan == 0 || bilangan == 1 {
18     prima = false
19 }
20
21 for i := 2; i <= bilangan-1; i++ {
22     if bilangan%i == 0 {
23         prima = false
24         break
25     }
26 }
27
28 if prima {
29     fmt.Println("Prima : ", prima)
30 } else {
31     fmt.Println("Bukan Prima : ", prima)
32 }
33
34 }

```

Below the code editor is a terminal window showing the execution of the program:

```

Faktor : 1 7
PS D:\#Coding\Go\modul-10> go run .\soal3after.go
Bilangan : 12
Faktor : 1 2 3 4 6 12 Prima : false
PS D:\#Coding\Go\modul-10> go run .\soal3after.go
Bilangan : 7
Faktor : 1 7 Prima : true
PS D:\#Coding\Go\modul-10>

```

To the right of the terminal is a small note pad window titled "Untitled - Notepad" containing the text:

```

File Edit Format View Help
109082500101
Raffi Yuliansyah
IF-13-07
Ln 3, Cc 100% Windows (CRLF) UTF-8

```

Deskripsi program

Before

Program tersebut memiliki tujuan menampilkan beberapa angka yang menjadi faktor dari suatu bilangan bulat.

Pada awal program akan diinisialisasikan variabel “bilangan” bertipe data integer untuk menyimpan angka yang diinput. Kemudian akan diinputkan suatu bilangan bulat dengan `fmt.Scan(&bilangan)` dan disimpan di variabel bilangan. Untuk menentukan angka apa saja yang menjadi faktor dari variabel bilangan menggunakan for-loop. Dimana pada forloop tersebut akan dicek dari angka 1 sampai dengan variabel bilangan apakah angka tersebut dapat membagi variabel bilangan.

```

for i:=1; i<=bilangan; i++{
    if bilangan%i == 0 {
        fmt.Println(i, " ")
    }
}

```

Apabila var bilangan dimodulo dengan i (angka yang akan dicek berdasarkan loop) memiliki hasil 0 maka angka tersebut ditampilkan sebagai faktor dari variabel bilangan. Program selesai.

After

Program tersebut memiliki tujuan menampilkan beberapa angka yang menjadi faktor dari suatu bilangan bulat dan menentukan apakah bilangan tersebut prima atau bukan.

Pada awal program akan diinisialisasikan variabel “bilangan” bertipe data integer untuk menyimpan angka yang diinput dan variabel “prima” bertipe boolean. Kemudian akan diinputkan suatu bilangan bulat dengan `fmt.Scan(&bilangan)` dan disimpan di variabel bilangan. Untuk menentukan angka apa saja yang menjadi

faktor dari variabel bilangan menggunakan for-loop. Dimana pada forloop tersebut akan memeriksa dari angka 1 sampai dengan variabel bilangan, apakah angka angka tersebut dapat membagi variabel bilangan.

```
for i:=1; i<=bilangan; i++{  
  
    if bilangan%i == 0 {  
  
        fmt.Println(i, " ")
```

Apabila var bilangan dimodulo dengan i (angka yang akan dicek berdasarkan loop) memiliki hasil 0 maka angka tersebut ditampilkan sebagai faktor dari variabel bilangan. Kemudian untuk menentukan apakah bilangan tersebut prima atau bukan menggunakan cara foor-loop dimana akan diperiksa dari 2 sampai dengan satu angka sebelum variabel bilangan, apakah angka angka tersebut bisa membagi variabel bilangan.

```
if bilangan == 0 || bilangan == 1 {  
  
    prima = false  
  
}  
  
for i := 2; i <= bilangan-1; i++ {  
  
    if bilangan%i == 0 {  
  
        prima = false  
  
    }  
  
}  
  
if prima {  
  
    fmt.Println("Prima : ", prima)
```

Apabila isi dari bilangan adalah 0 dan 1 maka prima bernilai false yang menandakan bilangan bukanlah bilangan prima. Dengan foor-loop akan dicek dari 2 sampai (bilangan-1) apabila terdapat i yang habis membagi bilangan maka prima bernilai true yang menandakan bahwa bilangan termasuk prima. Program selesai.