

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11

SWITCH-CASE



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 jam int
    fmt.Scan(&jam)
    switch {
        case jam == 0:
            fmt.Println("12 AM")
        case jam == 12:
            fmt.Println("12 PM")
        case jam < 12:
            fmt.Println(jam, " AM")
        case jam >= 12:
            jam -= 12
            fmt.Println(jam, " PM")
        default:
            fmt.Println("invalid")
```

```
}
```

```
}
```

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 `main` package with a `main()` function that reads an integer from the user and prints it in 12-hour format (AM/PM). The code includes a switch statement for values 0, 12, and ranges 1-11 and 13-23.
- Terminal:** Shows the command `go run .\guided1.go` being run twice in a PowerShell window. The first run outputs "1 PM" and the second run outputs "12 AM".
- Note Window:** A separate window titled "Untitled - Notepad" contains personal information: "109082500101", "Raffi Yuliansyah", and "IF-13-07".

Deskripsi program

Program ini adalah program konversi waktu yang mengubah format 24 jam menjadi format 12 jam (AM/PM). Pertama, dibuat variabel jam bertipe int untuk menyimpan input waktu. Program akan meminta input sebuah angka jam format 24.

Untuk mengubah format 24 menjadi format 12 digunakan switch case. Jika jam berisi angka 0, program akan menerjemahkannya sebagai "12 AM" (tengah malam), dan jika tepat angka 12, ia akan mencetak "12 PM". Untuk 1 - 12, program langsung menambahkan label "AM". Sebaliknya, jika angkanya lebih dari 12, program akan menguranginya dengan 12 terlebih dahulu lalu melabelinya dengan "PM". Terakhir, program juga disiapkan untuk mendeteksi angka yang tidak masuk akal (invalid) apabila input kurang dari 0.

2. Guided 2

Source Code

```
package main
```

```
import "fmt"
```

```

func main() {

    var tanaman string

    fmt.Scanf("%s", &tanaman)

    switch tanaman {

        case "nepenthes":

            fmt.Println("Termasuk Tanaman Karnivora \nAsli
Indonesia")

        case "venus":

            fmt.Println("Termasuk Tanaman Karnivora \nBukan
Asli Indonesia")

        default:

            fmt.Println("Tidak termasuk tanaman")

    }

}

```

Screenshot program

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

- Code Editor:** The main editor window displays the Go code for identifying carnivorous plants. It includes imports for `fmt`, a `main` function with a `switch` statement for `tanaman`, and three cases: `nepenthes` and `venus` which print to the terminal, and a default case which prints "Tidak termasuk tanaman".
- Terminal:** The terminal window shows the command `go run .\guided2.go` being run twice. The first run prints "nepenthes" followed by the expected output "Termasuk Tanaman Karnivora Asli Indonesia". The second run prints "venus" followed by "Tidak termasuk tanaman".
- Notepad:** A separate window titled "Untitled - Notepad" contains personal information: "109082500101", "Raffi Yuliansyah", and "IF-13-07".
- Status Bar:** The bottom status bar shows file details like "Ln 17, Col 2", "Tab Size: 4", "UTF-8", and "CRLF". It also shows the Go version "1.25.1" and the "Go Live" button.

Deskripsi program

Program ini akan mengklasifikasikan apakah tanaman yang diinputkan termasuk karnivora dan atau asli Indonesia. Pertama dibuat variabel tanaman bertipe string yang akan menyimpan nama tanaman. Kemudian program akan meminta input nama tanaman. Setelah itu cara mengklasifikasikannya menggunakan switch case. Jika variabel tanaman berisi "nepenthes", program akan menampilkan tanaman karnivora asli Indonesia. Kalau variabel berisi "venus", akan ditampilkan tanaman karnivora, tapi bukan berasal dari Indonesia. Apabila isi dari variabel tanaman berisi selain dua tanaman tersebut, program akan menampilkan "Tidak termasuk tanaman".

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi, tarif int
    fmt.Scan(&kendaraan)
    fmt.Scan(&durasi)

    switch kendaraan {
        case "mobil", "Mobil":
            if durasi > 2 {
                tarif = 20000
            } else {
                tarif = 15000
            }
        case "motor", "Motor":
```

```
    if durasi > 2 {  
  
        tarif = 9000  
  
    } else {  
  
        tarif = 7000  
  
    }  
  
    case "truk", "Truk":  
  
        if durasi > 2 {  
  
            tarif = 35000  
  
        } else {  
  
            tarif = 25000  
  
        }  
  
    default:  
  
        fmt.Println("Invalid")  
  
        tarif = 0  
  
    }  
  
    fmt.Println("Tarif = ", tarif)  
}
```

Screenshot program

```

File Edit Selection View Go Run ...
modul-10 > guided1.go 1 > guided2.go 1 > guided3.go 1 > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan, d1, d2, d3, d4 int
7     fmt.Println("Bilangan: ")
8     fmt.Scan(&bilangan)
9     d4 = bilangan % 10
10    d3 = (bilangan % 100) / 10
11    d2 = (bilangan % 1000) / 100
12    d1 = bilangan / 1000
13    if d1 < d2 && d2 < d3 && d3 < d4 {
14        fmt.Println("Terurut Membesar")
15    } else if d1 > d2 && d2 > d3 && d3 > d4 {
16        fmt.Println("Terurut Mengurut")
17    } else {
18        fmt.Println("Tidak Terurut")
19    }
20 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Coding\Go\modul-10> go run .\guided3.go
Bilangan: 2489
Terurut Membesar
PS D:\Coding\Go\modul-10> go run .\guided3.go
Bilangan: 3861
Tidak Terurut
PS D:\Coding\Go\modul-10> go run .\guided3.go
Bilangan: 9651
Terurut Mengurut
PS D:\Coding\Go\modul-10>

```

The screenshot shows a code editor with several tabs open, including guided1.go, guided2.go, guided3.go (the active tab), and several files named 'soal'. The code in guided3.go reads a four-digit number from the user, extracts its digits, and prints whether they are in ascending, descending, or random order. A terminal window shows the program's execution and output for three different inputs: 2489, 3861, and 9651.

Deskripsi program

Program ini adalah program penentu tarif parkir yang menghitung biaya berdasarkan jenis kendaraan dan lama waktu parkir. Pertama, dibuat variabel kendaraan bertipe *string* untuk menyimpan jenis kendaraan, serta variabel durasi dan tarif bertipe *int* untuk menyimpan lama parkir dan hasil biaya. Program akan meminta input jenis kendaraan dan angka durasi parkir.

Untuk menentukan besaran biaya, digunakan struktur *switch case* pada variabel kendaraan. Jika variabel berisi "mobil" atau "Mobil", program akan memeriksa durasi menggunakan *if-else*; apabila durasi lebih dari 2, variabel tarif diisi 20000, sedangkan jika tidak maka 15000. Logika serupa diterapkan untuk "motor" (tarif 9000 jika durasi lebih dari 2, selain itu 7000) dan "truk" (tarif 35000 jika durasi lebih dari 2, selain itu 25000). Apabila input kendaraan tidak dikenali, program akan mencetak "Invalid" dan mengatur tarif menjadi 0. Terakhir, program akan menampilkan total tarif yang harus dibayar ke layar.

TUGAS

1. Tugas 1

Source code

```
package main
```

```
import "fmt"

func main() {
    var ph float64
    fmt.Scan(&ph)

    switch {
        case ph >= 6.5 && ph <= 8.6:
            fmt.Println("Air Layak Minum")
        case ph <= 14 && ph >= 0 && (ph < 6.5 || ph > 8.6):
            fmt.Println("Air Tidak Layak Minum")
        default:
            fmt.Println("Nilai pH tidak valid. Nilai pH harus
antara 0 dan 14")
    }
}
```

Screenshoot program

The screenshot shows a Go development environment with several tabs open in the top bar: guided1.go, guided2.go, guided3.go, soal1.go (active), soal2.go, and soal3.go. The main editor window contains the following Go code:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float64
7     fmt.Scan(&ph)
8
9     switch {
10    case ph >= 6.5 && ph <= 8.6:
11        fmt.Println("Air Layak Minum")
12    case ph <= 14 && ph >= 0 && (ph < 6.5 || ph > 8.6):
13        fmt.Println("Air Tidak Layak Minum")
14    default:
15        fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14")
16    }
17 }
18

```

A terminal window below shows the execution of the program:

```

PS D:\#Coding\Go\modul-11> go run .soal1.go
8.6
Air Layak Minum
PS D:\#Coding\Go\modul-11> go run .soal1.go
● 9
● Air Tidak Layak Minum
PS D:\#Coding\Go\modul-11> go run .soal1.go
16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14
○ PS D:\#Coding\Go\modul-11> []

```

A small Notepad window titled "Untitled - Notepad" is also visible, containing the text:

```

109082500101
Raffi Yuliansyah
[IF-13-07]

```

Deskripsi program

Program ini adalah program penentu kualitas air yang mengklasifikasikan apakah air tersebut layak dikonsumsi berdasarkan tingkat keasamannya (pH). Pertama, dibuat variabel ph bertipe *float64* untuk menyimpan nilai input yang berupa angka desimal. Program akan meminta input nilai pH air dari pengguna.

Untuk menentukan apakah air layak diminum, digunakan struktur *switch case* dengan kondisi logika. Jika variabel ph bernilai antara 6.5 hingga 8.6, program akan menampilkan "Air Layak Minum". Apabila nilai ph berada di rentang valid skala pH (0 sampai 14) namun di luar batas layak minum tadi (kurang dari 6.5 atau lebih dari 8.6), maka akan ditampilkan "Air Tidak Layak Minum". Terakhir, jika nilai yang dimasukkan berada di luar skala 0 hingga 14, program akan masuk ke kondisi *default* dan menampilkan pesan bahwa nilai pH tidak valid.

2. Tugas 2

Source code

```

package main

import "fmt"

func main() {

```

```
var kendaraan string  
  
var total, waktu int  
  
  
fmt.Scan(&kendaraan)  
  
fmt.Scan(&waktu)  
  
  
switch kendaraan {  
  
case "motor", "Motor":  
  
    total = waktu * 2000  
  
  
case "mobil", "Mobil":  
  
    total = waktu * 5000  
  
  
case "truk", "Truk":  
  
    total = waktu * 8000  
  
  
default:  
  
    fmt.Println("Invalid Input")  
  
}  
  
fmt.Println("Tarif Parkir ", waktu, " jam : Rp. ",  
total)  
}
```

Screenshoot program

The screenshot shows a Go development environment with several tabs open. The main tab contains the following Go code:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var total, waktu int
8
9     fmt.Scan(&kendaraan)
10    fmt.Scan(&waktu)
11
12    switch kendaraan {
13        case "motor", "Motor":
14            total = waktu * 2000
15
16        case "mobil", "Mobil":
17            total = waktu * 5000
18
19        case "truk", "Truk":
20            total = waktu * 8000
21
22    }
23
24    fmt.Println("Total Biaya Parkir: ", total)
25}

```

The terminal window shows the execution of the program and its output:

```

PS D:\#Coding\Go\modul-11> go run .\soal2.go
motor 3
Tarif Parkir 3 jam : Rp. 6000
PS D:\#Coding\Go\modul-11> go run .\soal2.go
mobil 1
Tarif Parkir 1 jam : Rp. 5000
PS D:\#Coding\Go\modul-11> go run .\soal2.go
truk 5
Tarif Parkir 5 jam : Rp. 40000
PS D:\#Coding\Go\modul-11>

```

A separate Notepad window displays the following text:

```

*Untitled - Notepad
File Edit Format View Help
109082500101
Raffi Yuliansyah
[IF-13-07]
Ln 3 Cc 100% Windows (CRLF) UTF-8

```

Deskripsi program

Program ini adalah program penghitung tarif parkir yang mengkalkulasi total biaya berdasarkan jenis kendaraan dan durasi parkir. Pertama, dibuat variabel kendaraan bertipe *string* untuk menyimpan jenis kendaraan, serta variabel total dan waktu bertipe *int* yang masing-masing digunakan untuk menyimpan hasil perhitungan biaya dan lama parkir. Kemudian program akan meminta input jenis kendaraan dan durasi waktu parkir dari pengguna.

Setelah itu, perhitungan biaya dilakukan menggunakan struktur *switch case* pada variabel kendaraan. Jika variabel berisi "motor" atau "Motor", program akan menghitung total dengan mengalikan waktu dengan 2000. Jika berisi "mobil" atau "Mobil", durasi waktu akan dikalikan dengan 5000. Sedangkan untuk "truk" atau "Truk", total diperoleh dari perkalian waktu dengan 8000. Apabila input kendaraan tidak sesuai dengan ketiga jenis tersebut, program akan mencetak "Invalid Input". Terakhir, program akan menampilkan teks rincian tarif parkir sesuai durasi beserta total biaya ke layar.

3. Tugas 3

Source code

```

package main

import "fmt"

func main() {
    var num int
}
```

```
fmt.Scan(&num)

switch {
    case num%2 == 0 && num%10 != 0 && num%5 != 0:
        fmt.Println("Kategori : Bilangan Genap")
        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d", num, num+1, num*(num+1))

    case num%2 == 1 && num%10 != 0 && num%5 != 0:
        fmt.Println("Kategori : Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d", num, num+1, num+(num+1))

    case num%10 == 0:
        fmt.Println("Kategori : Bilangan Kelipatan 10")
        fmt.Printf("Hasil Pembagian antara %d / %d = %d", num, 10, num/10)

    case num%5 == 0:
        fmt.Println("Kategori : Bilangan Kelipatan 5")
        fmt.Printf("Hasil Kuadrat dari %d ^ %d = %d", num, 2, num*num)

    default:
        fmt.Println("Invalid Input")
}
}
```

Screenshoot program

The screenshot shows a Go development environment with several tabs open. The main tab contains the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var num int
7
8     fmt.Scan(&num)
9
10    switch {
11        case num%2 == 0 && num%10 != 0 && num%5 != 0:
12            fmt.Println("Kategori : Bilangan Genap")
13            fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d", num, num+1, num*(num+1))
14
15        case num%2 == 1 && num%10 != 0 && num%5 != 0:
16            fmt.Println("Kategori : Bilangan Ganjil")
17            fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d", num, num+1, num+num+1)
18
19        case num%10 == 0:
20            fmt.Println("Kategori : Bilangan Kelipatan 10")
21            fmt.Printf("Hasil pembagian dengan %d = %d", 10, num/10)
22
23    }
24 }
```

The terminal window shows the execution of the program and its output:

```
Hasil Kuadrat dari 5 ^ 2 = 25
PS D:\#Coding\Go\modul-11> go run .\soal3.go
● 8
Kategori : Bilangan GenapHasil perkalian dengan bilangan berikutnya 8 * 9 = 72
● PS D:\#Coding\Go\modul-11> go run .\soal3.go
● 25
Kategori : Bilangan Kelipatan 5
Hasil Kuadrat dari 25 ^ 2 = 625
● PS D:\#Coding\Go\modul-11> go run .\soal3.go
● 20
Kategori : Bilangan Kelipatan 10
```

A separate Notepad window titled "Untitled - Notepad" shows the following text:

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

Deskripsi program

Program ini adalah program pengolah bilangan yang mengategorikan angka inputan dan melakukan operasi matematika spesifik berdasarkan sifat bilangan tersebut. Pertama, dibuat variabel num bertipe *int* untuk menyimpan angka yang dimasukkan. Kemudian, program akan meminta input angka dari pengguna. Proses klasifikasi dilakukan menggunakan struktur *switch case* dengan berbagai kondisi logika. Jika num adalah bilangan genap (habis dibagi 2) namun bukan kelipatan 10 maupun 5, program akan menampilkan kategori "Bilangan Genap" lalu menghitung hasil perkalian angka tersebut dengan bilangan urutan setelahnya. Jika num adalah bilangan ganjil (sisa bagi 1) dan juga bukan kelipatan 10 atau 5, program menampilkan kategori "Bilangan Ganjil" lalu menjumlahkan angka tersebut dengan bilangan urutan setelahnya.

Selanjutnya, jika angka tersebut habis dibagi 10, program mengategorikannya sebagai "Bilangan Kelipatan 10" dan menampilkan hasil pembagian angka tersebut dengan 10. Jika angka tersebut habis dibagi 5 (namun tidak memenuhi kondisi sebelumnya), program mengategorikannya sebagai "Bilangan Kelipatan 5" dan menampilkan hasil kuadrat dari angka tersebut. Apabila input tidak memenuhi kriteria-kriteria di atas, program akan masuk ke kondisi *default* dan mencetak "Invalid Input".