

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11

SWITCH-CASE



Disusun oleh:

FIRDAUS RAMADHANA

109082500151

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.Print("Masukkan waktu dalam bentuk 24 jam: ")
    fmt.Scan(&jam)
    switch {
        case jam == 0:
            fmt.Println("12 AM")
        case jam == 12:
            fmt.Println("12 PM")
        case jam > 0 && jam < 12:
            fmt.Printf("%d AM\n", jam)
        case jam > 12 && jam < 24:
            fmt.Printf("%d PM\n", jam-12)
        default:
            fmt.Println("Tidak ada waktu yang valid")
    }
}
```

Screenshoot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)
CONTOH TANGKAPAN LAYAR:

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a tree view of files under "modull1_switch-case". The "mungo" folder contains "guided1", "guided2", "guided3", "unguided1", "unguided2", "unguided3", and "main.go".
- Code Editor:** Displays a Go program named "main.go" with the following code:

```
func main() {
    var jam int
    fmt.Println("Masukkan waktu dalam bentuk 24 jam: ")
    fmt.Scan(&jam)
    switch {
        case jam == 0:
            fmt.Println("12 AM")
        case jam == 12:
            fmt.Println("12 PM")
        case jam > 0 && jam < 12:
            fmt.Printf("%d AM\n", jam)
        case jam > 12 && jam < 24:
            fmt.Printf("%d PM\n", jam-12)
        default:
            fmt.Println("Tidak ada waktu yang valid")
    }
}
```
- Terminal:** Shows three runs of the program:

```
D:\z PERKULIAHAN ASDY GEBOYY\Praktikum\go\modull1_switch-case\unguided2>cd ..
D:\z PERKULIAHAN ASDY GEBOYY\Praktikum\go\modull1_switch-case\guided2>go run main.go
Masukkan waktu dalam bentuk 24 jam: 13
1 PM

D:\z PERKULIAHAN ASDY GEBOYY\Praktikum\go\modull1_switch-case\guided1>go run main.go
Masukkan waktu dalam bentuk 24 jam: 8
12 AM

D:\z PERKULIAHAN ASDY GEBOYY\Praktikum\go\modull1_switch-case\guided1>
```
- Output Panel:** Shows the output of the third terminal command:

```
FIRDAUS RAMADHANA
189082500151
IF-13-07
```
- Bottom Status Bar:** Shows file path, line number, tab size, encoding, and other system information.

Deskripsi program

Program di atas meminta user untuk memasukkan waktu dalam format 24 jam, kemudian menggunakan struktur switch untuk mengonversinya ke format 12 jam (AM/PM). Jika user memasukkan angka 0, program menampilkan "12 AM", sedangkan jika memasukkan 12, program menampilkan "12 PM". Untuk waktu antara 1 sampai 11, program menampilkan waktu yang sama tetapi dalam format AM. Untuk waktu antara 13 sampai 23, program menampilkan waktu dalam format PM dengan mengurangi angka input sebesar 12. Jika nilai waktu berada di luar rentang 0–23, program menampilkan pesan "Tidak ada waktu yang valid".

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var namaTumbuhan string
    fmt.Print("Masukkan nama tumbuhan: ")
    fmt.Scan(&namaTumbuhan)
    switch namaTumbuhan {
        case "nepenthes":
```

```

        fmt.Println("Termasuk tumbuhan karnivora.")

        fmt.Println("Asli Indonesia.")

    case "venus":

        fmt.Println("Termasuk tumbuhan karnivora.")

        fmt.Println("Bukan Asli Indonesia.")

    default:

        fmt.Println("Tidak termasuk tanaman Karnivora")

    }

}

```

Screenshoot program

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure named "MODUL11_SWITCH-CASE" containing several files: guided1, guided2, guided3, unguided1, unguided2, and unguided3, each with a main.go file.
- Code Editor:** Displays the content of main.go:

```

package main

import "fmt"

func main() {
    var namaTumbuhan string
    fmt.Print("Masukkan nama tumbuhan: ")
    fmt.Scan(&namaTumbuhan)
    switch namaTumbuhan {
        case "nepenthes":
            fmt.Println("Termasuk tumbuhan karnivora.")
            fmt.Println("Asli Indonesia.")
        case "venus":
            fmt.Println("Termasuk tumbuhan karnivora.")
            fmt.Println("Bukan Asli Indonesia.")
        default:
            fmt.Println("Tidak termasuk tanaman Karnivora")
    }
}

```
- Terminal:** Shows command-line output for three runs of the program:
 - Run 1: Input "nepenthes", Output: "Termasuk tumbuhan karnivora." and "Asli Indonesia."
 - Run 2: Input "venus", Output: "Termasuk tumbuhan karnivora." and "Bukan Asli Indonesia."
 - Run 3: Input "karedok", Output: "Tidak termasuk tanaman Karnivora"
- Output:** Shows a terminal window titled "FIRDAUS RAMADHANA" with the user's name and session ID.
- Status Bar:** Shows the current file path as "D:\2. PERKULIAHAN ASOY GEBOYY\Praktikum\go\modul11_switch-case\guided2\main.go", line 3, column 9, and other standard status bar information.

Deskripsi program

Program ini meminta user untuk memasukkan nama sebuah tumbuhan, kemudian menggunakan struktur switch untuk mengidentifikasi apakah tumbuhan tersebut termasuk tumbuhan karnivora dan apakah berasal dari Indonesia. Jika user memasukkan nama "nepenthes", program menampilkan bahwa tumbuhan tersebut merupakan tumbuhan karnivora dan berasal dari Indonesia. Jika user memasukkan "venus", program menampilkan bahwa tumbuhan tersebut adalah tumbuhan karnivora namun bukan asli Indonesia. Jika input selain kedua nama tersebut dimasukkan, program akan menampilkan pesan bahwa tumbuhan tersebut tidak termasuk tanaman karnivora.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var lamaParkir, biaya int
    fmt.Println("Masukkan jenis kendaraan (mobil/motor/truk): ")
    fmt.Scan(&kendaraan)
    fmt.Println("Masukkan lama parkir (dalam jam): ")
    fmt.Scan(&lamaParkir)
    switch kendaraan {
        case "motor":
            if lamaParkir <=2 {
                biaya = 7000
                fmt.Println("Biaya parkir: Rp", biaya)
            } else {
                biaya = 9000
                fmt.Println("Biaya parkir: Rp", biaya)
            }
        case "mobil":
            if lamaParkir <=2 {
                biaya = 15000
                fmt.Println("Biaya parkir: Rp", biaya)
            } else {
                biaya = 20000
                fmt.Println("Biaya parkir: Rp", biaya)
            }
        case "truk":
            if lamaParkir <=2 {
                biaya = 25000
            }
    }
}
```

```

        fmt.Println("Biaya parkir: Rp", biaya)

    } else {

        biaya = 35000

        fmt.Println("Biaya parkir: Rp", biaya)

    }

default:

    fmt.Println("Jenis kendaraan atau durasi tidak
valid.")

    fmt.Println("Biaya parkir: Rp 0")

}

}

```

Screenshot program

The screenshot shows a code editor with a Go file named `main.go`. The code implements a parking fee calculation logic using a switch statement. It handles three types of vehicles: mobil, motor, and truk. For each vehicle type, it asks for the duration of parking in hours and calculates the fee based on the following rules:

- mobil:** If duration is less than or equal to 2 hours, the fee is 15000. Otherwise, it is 20000.
- motor:** If duration is less than or equal to 2 hours, the fee is 7000. Otherwise, it is 9000.
- truk:** If duration is less than or equal to 2 hours, the fee is 15000. Otherwise, it is 25000.

If the vehicle type or duration is invalid, the program prints an error message and a fee of 0.

The terminal window below shows the execution of the program and its output for different inputs:

```

D:\_PENULISAN ASN GEDORY\Praktikum\modul11_switch-case\guided>go run main.go
Masukan jenis kendaraan (mobil/motor/truk): mobil
Masukan lama parkir (dalam jam): 4
Biaya parkir: Rp 20000

D:\_PENULISAN ASN GEDORY\Praktikum\modul11_switch-case\guided>go run main.go
Masukan jenis kendaraan (mobil/motor/truk): motor
Masukan lama parkir (dalam jam): 3
Biaya parkir: Rp 9000

D:\_PENULISAN ASN GEDORY\Praktikum\modul11_switch-case\guided>go run main.go
Masukan jenis kendaraan (mobil/motor/truk): sepeda
Masukan lama parkir (dalam jam): 1
Jenis kendaraan atau durasi tidak valid.

Biaya parkir: Rp 0

```

Deskripsi program

Program ini meminta user untuk memasukkan jenis kendaraan yaitu mobil, motor, atau truk serta lama parkir dalam satuan jam. Setelah menerima input tersebut, program menggunakan struktur switch untuk menentukan biaya parkir berdasarkan jenis kendaraan dan durasi parkir. Untuk kendaraan motor, biaya parkir adalah Rp 7.000 jika lama parkir kurang dari atau sama dengan dua jam, dan Rp 9.000 jika lebih lama. Untuk mobil, tarifnya adalah Rp 15.000 untuk parkir hingga dua jam dan Rp 20.000 jika melebihi dua jam. Untuk truk, biayanya Rp 25.000 untuk dua jam pertama

dan Rp 35.000 jika durasinya lebih lama. Jika user memasukkan jenis kendaraan yang tidak dikenali atau input tidak sesuai, program menampilkan pesan bahwa data tidak valid dan menetapkan biaya parkir sebesar Rp 0.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var pH float64

    fmt.Print("Masukkan pH air: ")

    fmt.Scan(&pH)

    switch {
    case pH < 0 || pH > 14:
        fmt.Println("Input tidak valid, rentang pH 0 - 14")
    case pH >= 6.5 && pH <= 8.6:
        fmt.Println("Air Layak Minum")
    default:
        fmt.Println("Air Tidak Layak Minum")
    }
}
```

```
}
```

Screenshoot program

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

- Explorer:** Shows a tree view of files and packages. The current file is `main.go` under the `modul11_switch-case` package. Other files include `guided1.go`, `guided2.go`, `guided3.go`, `unguided1.go`, `unguided2.go`, `unguided3.go`, and `main.go`.
- Editor:** Displays the `main.go` file content:import "fmt"

func main() {
 var pH float64

 fmt.Print("Masukkan pH air: ")
 fmt.Scan(&pH)

 switch {
 case pH < 0 || pH > 14:
 fmt.Println("Input tidak valid, rentang pH 0 - 14")
 case pH >= 6.5 && pH <= 8.6:
 fmt.Println("Air Layak Minum")
 default:
 fmt.Println("Air Tidak Layak Minum")
 }
}
- Terminal:** Shows command-line interactions:D:\2. PERKULIAHAN ASOY GEBOYY\Praktikum\go\modul11_switch-case\unguided1>go run main.go
Masukkan pH air: 8.6
Air Layak Minum

D:\2. PERKULIAHAN ASOY GEBOYY\Praktikum\go\modul11_switch-case\unguided1>go run main.go
Masukkan pH air: 9
Air Tidak Layak Minum

D:\2. PERKULIAHAN ASOY GEBOYY\Praktikum\go\modul11_switch-case\unguided1>go run main.go
Masukkan pH air: 16
Input tidak valid, rentang pH 0 - 14
- Message Box:** A window titled "FIRDAUS RAMADHANA" displays the message:

```
FIRDAUS RAMADHANA
109002500151
IF - 13 - 07
```

Deskripsi program

Program ini meminta user untuk memasukkan nilai pH air dalam bentuk bilangan desimal. Setelah menerima input, program menggunakan struktur switch untuk menentukan apakah nilai pH tersebut valid dan apakah air layak untuk diminum. Jika nilai pH berada di luar rentang 0 hingga 14, program menampilkan pesan bahwa input tidak valid. Jika nilai pH berada pada kisaran 6.5 hingga 8.6, program menyatakan bahwa air tersebut layak minum. Untuk semua nilai pH lainnya yang masih berada dalam rentang valid, program menampilkan bahwa air tersebut tidak layak minum.

2. Tugas 2

Source code

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
var durasiParkir int

var jenisKendaraan string

var biayaParkir int

fmt.Print("Masukkan jenis kendaraan (mobil/motor/truk): ")

fmt.Scan(&jenisKendaraan)

fmt.Print("Masukkan durasi parkir (dalam jam): ")

fmt.Scan(&durasiParkir)

switch jenisKendaraan {

case "motor":

    biayaParkir = 2000

    if durasiParkir <= 1 {

        fmt.Println("Biaya parkir: Rp", biayaParkir * durasiParkir)

    } else {

        fmt.Println("Biaya parkir: Rp", biayaParkir * durasiParkir)

    }

case "mobil":

    biayaParkir = 5000

    if durasiParkir <= 1 {

        fmt.Println("Biaya parkir: Rp", biayaParkir * durasiParkir)

    } else {

        fmt.Println("Biaya parkir: Rp", biayaParkir * durasiParkir)

    }

case "truk":
```

```

biayaParkir = 8000

if durasiParkir <= 1 {

    fmt.Println("Biaya parkir: Rp", biayaParkir * durasiParkir)

} else {

    fmt.Println("Biaya parkir: Rp", biayaParkir * durasiParkir)

}

}

```

Screenshot program

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

- Code Editor:** The main editor window displays a Go file named `modul11_switch-case/main.go`. The code implements a switch-case statement to calculate parking fees based on vehicle type and duration.
- Terminal:** The terminal window shows the execution of the program. It prompts the user for the type of vehicle and its duration, then prints the total parking fee.
- Output:** The terminal output is as follows:


```

Masukkan jenis kendaraan (mobil/motor/truk): motor
Masukkan durasi parkir (dalam jam): 3
Biaya parkir: Rp 6000
D:\2. PERKULIAHAN AGOV GEBOY\Praktikum\go\modul11_switch-case\unguided2>go run main.go
Masukkan jenis kendaraan (mobil/motor/truk): mobil
Masukkan durasi parkir (dalam jam): 1
Biaya parkir: Rp 5000
D:\2. PERKULIAHAN AGOV GEBOY\Praktikum\go\modul11_switch-case\unguided2>go run main.go
Masukkan jenis kendaraan (mobil/motor/truk): truk
Masukkan durasi parkir (dalam jam): 5
Biaya parkir: Rp 40000
      
```

Deskripsi program

Program ini meminta user untuk memasukkan jenis kendaraan yaitu motor, mobil, atau truk serta durasi parkir dalam satuan jam. Setelah menerima input, program menentukan tarif per jam berdasarkan jenis kendaraan yang dipilih. Untuk kendaraan motor, tarifnya adalah Rp 2.000 per jam; untuk mobil, tarifnya Rp 5.000 per jam; dan untuk truk, tarifnya Rp 8.000 per jam. Program kemudian menghitung total biaya parkir dengan mengalikan tarif per jam dengan durasi parkir yang dimasukkan oleh

user. Terakhir, program menampilkan total biaya parkir sesuai dengan jenis kendaraan dan lama waktu parkir tersebut.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var num int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&num)

    switch {
    case num == 5:
        hasil := num + (num + 1)
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", num, num+1, hasil)
    case num%10 == 0:
        hasil := num / 10
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", num, hasil)
    case num%5 == 0:
        hasil := num * num
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", num, hasil)
    case num%2 == 0:
        hasil := num * (num + 1)
        fmt.Println("Kategori: Bilangan Genap")
```

```

        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d =
        %d\n", num, num+1, hasil)

    default:

        hasil := num + (num + 1)

        fmt.Println("Kategori: Bilangan Ganjil")

        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d =
        %d\n", num, num+1, hasil)

    }

}

```

Screenshoot program

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

- Editor:** Displays the source code of `main.go` which contains a switch-case statement for categorizing numbers based on their properties.
- Terminal:** Shows three command-line executions of the program:
 - Execution 1: Input 5, Output: Kategori: Bilangan Ganjil
 - Execution 2: Input 10, Output: Kategori: Bilangan Kelipatan 10
 - Execution 3: Input 25, Output: Kategori: Bilangan Kelipatan 5
- Output:** Shows the results of the program's execution in the terminal panes.

Deskripsi program

Program ini meminta user untuk memasukkan sebuah bilangan bulat, kemudian menggunakan struktur switch untuk menentukan kategori bilangan tersebut dan menghitung hasil operasi yang sesuai. Jika bilangan yang dimasukkan adalah 5, program secara khusus mengelompokkannya sebagai bilangan ganjil dan menghitung penjumlahan bilangan tersebut dengan bilangan berikutnya. Jika bilangan merupakan kelipatan 10, program membaginya dengan 10. Jika bilangan merupakan kelipatan 5 (selain 5), program menghitung hasil kuadrat dari bilangan tersebut. Untuk bilangan genap, program menghitung hasil perkalian bilangan tersebut dengan bilangan sesudahnya. Jika tidak memenuhi kondisi sebelumnya, bilangan

dianggap sebagai bilangan ganjil, dan program menampilkan hasil penjumlahan bilangan dengan bilangan berikutnya. Program kemudian menampilkan kategori bilangan beserta hasil perhitungannya.