

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 11**

**SWITCH CASE**



**Disusun oleh:**

**Sofwan Nuha Al Faruq**

**109082500032**

**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 hour24 int

    fmt.Print("Masukkan jam (0-23): ")
    fmt.Scan(&hour24)

    if hour24 < 0 || hour24 > 23 {
        fmt.Println("Error: Jam harus antara 0-23")
        return
    }

    var hour12 int
    var period string

    switch {
    case hour24 == 0:
        hour12 = 12
        period = "AM"
    case hour24 < 12:
```

#### Screenshot program

The screenshot shows a VS Code interface with a dark theme. In the center-left, there's a code editor with tabs for 'guided-1.go' (active), 'guided-2.go', 'guided-3.go', 'soal-1.go', 'soal-2.go', and 'soal-3.go'. The code in 'guided-1.go' is a Go program that converts a 24-hour input into a 12-hour format with AM/PM. It includes error handling for inputs outside the 0-23 range. The output window at the bottom shows the execution of the program and its results. On the right side, there's an 'EXPLORER' sidebar showing a file tree for 'MODUL-11' containing the same files. A status bar at the bottom indicates the date and time.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var hour24 int
7
8     fmt.Print("Masukkan jam (0-23): ")
9     fmt.Scan(&hour24)
10
11    if hour24 < 0 || hour24 > 23 {
12        fmt.Println("Error: Jam harus antara 0-23")
13        return
14    }
15
16    var hour12 int
17    var period string
18
19    switch {
20        case hour24 == 0:
21            hour12 = 12
22            period = "AM"
23        case hour24 < 12:
24            hour12 = hour24
25            period = "AM"
26        case hour24 == 12:
27            hour12 = 12
28            period = "PM"
29        default:
30            hour12 = hour24 - 12
31    }
32
33    fmt.Printf("Format 24 jam: %d:%02d\n", hour24, hour24%100)
34    fmt.Printf("Format 12 jam: %d:%02d %s\n", hour12, hour24%100, period)
35}
```

Output:

```
sofwanuhaifa@MacBook-Air-Sofwan Modul-11 % clear
Masukkan jam (0-23): 12
Format 24 jam: 12:00
Format 12 jam: 12:00 PM
sofwanuhaifa@MacBook-Air-Sofwan Modul-11 % go run guided-1.go
Masukkan jam (0-23): 13
Format 24 jam: 13:00
Format 12 jam: 01:00 PM
sofwanuhaifa@MacBook-Air-Sofwan Modul-11 %
```

## Deskripsi program

Program ini digunakan untuk mengubah format jam 24 jam menjadi format 12 jam (AM/PM). Program membaca input angka jam, lalu memakai switch tanpa ekspresi untuk mengecek kondisi jam tersebut. Jika jam bernilai 0, program menampilkan "12 AM". Jika jam berada antara 1–11, program menampilkan jam tersebut dengan label "AM". Jika jam bernilai 12, program menampilkan "12 PM". Jika jam lebih dari 12 sampai 23, program akan mengonversinya ke format 12 jam dan menampilkan hasilnya dalam "PM". Jika angka yang dimasukkan tidak sesuai rentang 0–23, program menampilkan bahwa jam tidak valid.

## 2. Guided 2

### Source Code

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var namaTanaman
    string

    fmt.Println("Masukkan
n nama tanaman: ")
    fmt.Scanln(&namaTa
naman)

    namaTanaman =
    strings.ToLower(nam
aTanaman)

    switch namaTanaman
    {
        case "nepenthes":
            fmt.Println("Terma
suk Tanaman
```

## Screenshot program

The screenshot shows a Mac OS X desktop environment. In the foreground, a terminal window is open with the following command and output:

```
sofwanuhaifaruq@MacBook-Air-Sofwan Modul-11 % clear
Masukkan nama tanaman: nephentes
Tanaman tidak dikenali
● sofwanuhaifaruq@MacBook-Air-Sofwan Modul-11 % go run guided-2.go
Masukkan nama tanaman: nephentes
Tersusuk Tanaman Karnivora
Asli Indonesia
● sofwanuhaifaruq@MacBook-Air-Sofwan Modul-11 % go run guided-2.go
Masukkan nama tanaman: karedok
Masukkan nama tanaman: karedok
Tidak Tersusuk Tanaman Karnivora
↳ sofwanuhaifaruq@MacBook-Air-Sofwan Modul-11 %
```

Below the terminal is a code editor window titled "Modul-11" showing a Go program named "guided-2.go". The code defines a function "main" that prompts the user for a plant name, converts it to lowercase, and then uses a switch statement to check if it's "nephentes", "venus", or "karedok". If none of these, it prints "Tidak Tersusuk Tanaman Karnivora".

```
1  package main
2
3  import (
4      "fmt"
5      "strings"
6  )
7
8  func main() {
9      var namaTanaman string
10
11     fmt.Println("Masukkan nama tanaman: ")
12     fmt.Scanln(&namaTanaman)
13
14     namaTanaman = strings.ToLower(namaTanaman)
15
16     switch namaTanaman {
17     case "nepenthes":
18         fmt.Println("Tersusuk Tanaman Karnivora")
19         fmt.Println("Asli Indonesia")
20     case "venus":
21         fmt.Println("Tersusuk Tanaman Karnivora")
22         fmt.Println("Asli Indonesia")
23     case "karedok":
24         fmt.Println("Tidak Tersusuk Tanaman Karnivora")
25     default:
26         fmt.Println("Tanaman tidak dikenali")
27     }
28 }
```

The code editor interface includes tabs for other files like "guided-1.go", "guided-3.go", "soal-1.go", etc., and an Explorer sidebar showing the file structure. A status bar at the bottom indicates the date and time.

### Deskripsi program :

Program ini adalah membuat sebuah fitur untuk mengecek apakah nama tanaman yang diinput oleh pengguna termasuk tanaman karnivora atau tidak. Program menggunakan variabel tanaman yang bertipe data *string* untuk menyimpan nama tanaman yang dimasukkan. Setelah itu, program memakai struktur switch untuk menentukan kategori dari tanaman tersebut. Jika pengguna memasukkan kata "*nepenthes*", program akan menampilkan bahwa tanaman tersebut merupakan tanaman karnivora asli Indonesia. Jika tanaman yang dimasukkan adalah "*venus*", "*Sarracenia*", "*Byblis*", atau "*Drosera*", maka program akan memberikan output bahwa tanaman tersebut merupakan tanaman karnivora tetapi bukan asli Indonesia. Selain itu, jika nama tanaman tidak ada dalam daftar tersebut, program akan menampilkan bahwa tanaman tersebut tidak termasuk tanaman karnivora.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var jenisKendaraan string
    var lamaParkir int
    var tarif int

    fmt.Println("Masukkan jenis kendaraan (Motor/Mobil/Truk) :")
    fmt.Scanln(&jenisKendaraan)
    fmt.Println("Masukkan durasi parkir (dalam jam) : ")
    fmt.Scanln(&lamaParkir)

    if lamaParkir <= 0 {
        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
        fmt.Println("Tarif Parkir: Rp 0")
        return
    }

    switch jenisKendaraan {
    case "Motor":
        if lamaParkir <= 2 {
            tarif = 7000
        } else {
            tarif = 9000
        }
        fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
    case "Mobil":
        if lamaParkir <= 2 {
            tarif = 15000
        }
    }
}
```

## Screenshoot program

```
func main() {
    var jenisKendaraan string
    var lamaParkir int
    var tarif int

    fmt.Println("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
    fmt.Scanln(&jenisKendaraan)
    fmt.Println("Masukkan durasi parkir (dalam jam): ")
    fmt.Scanln(&lamaParkir)

    if lamaParkir <= 0 {
        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
        fmt.Println("Tarif Parkir: Rp 0")
        return
    }

    switch jenisKendaraan {
    case "Motor":
        if lamaParkir <= 2 {
            tarif = 7000
        } else {
            tarif = 9000
        }
        fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
    case "Mobil":
        if lamaParkir <= 2 {
            tarif = 15000
        } else {
    
```

Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk  
Masukkan durasi parkir (dalam jam): 5  
Tarif Parkir: Rp 35000  
● sofwanuhuafarug@MacBook-Air-Sofwan Modul-11 % go run guided-3.go  
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil  
Masukkan durasi parkir (dalam jam): 6  
Tarif Parkir: Rp 20000  
● sofwanuhuafarug@MacBook-Air-Sofwan Modul-11 % go run guided-3.go  
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor  
Masukkan durasi parkir (dalam jam): 2  
Tarif Parkir: Rp 7000  
● sofwanuhuafarug@MacBook-Air-Sofwan Modul-11 %

### Deskripsi program:

Pada program ini yaitu membuat sebuah sistem untuk menentukan tarif parkir berdasarkan jenis kendaraan dan lama parkir yang diinput oleh pengguna. Program ini memakai variabel kendaraan, durasi, dan tarif, di mana masing-masing digunakan untuk menyimpan jenis kendaraan, lamanya waktu parkir dalam jam, serta biaya parkir yang harus dibayar. Program ini juga menggunakan switch tanpa ekspresi, sehingga setiap kondisi pada case akan dicek sebagai perbandingan logis yang bernilai benar atau salah. Melalui kondisi tersebut, program akan menentukan tarif berbeda untuk Motor, Mobil, dan Truk, baik untuk durasi parkir 1–2 jam maupun lebih dari 2 jam. Jika jenis kendaraan tidak sesuai dengan pilihan yang tersedia, maka program akan menampilkan bahwa input tidak valid. asil akhir berupa tarif parkir yang akan ditampilkan kepada pengguna.



## **TUGAS**

### **1. Tugas 1**

#### **Source code**

```
package main

import "fmt"

func main() {
    var ph float64

    fmt.Println("masukan kadar pH: ")
    fmt.Scan(&ph)

    switch {
    case ph >= 6.5 && ph <= 8.6:
        fmt.Println("Air layak minum")
    case ph < 6.5 || ph > 8.6:
        fmt.Println("Air tidak layak minum")
    case ph > 14:
        fmt.Println("Input tidak valid")
    }
}
```

#### **Screenshot program**

The screenshot shows a Mac OS desktop environment with a VS Code window open. The code editor displays a Go file named 'soal-1.go' with the following content:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float64
7
8     fmt.Println("masukan kadar pH: ")
9     fmt.Scan(&ph)
10
11    switch {
12        case ph >= 6.5 && ph <= 8.6:
13            fmt.Println("Air layak minum")
14        case ph < 6.5 || ph > 8.6:
15            fmt.Println("Air tidak layak minum")
16        case ph > 14:
17            fmt.Println("Input tidak valid")
18    }
19 }
```

The terminal below the code editor shows the output of running the program:

```
sofwanuhaifaruq@MacBook-Air-Sofwan Modul-11 % clear
masukan kadar pH: 4
Air tidak layak minum
sofwanuhaifaruq@MacBook-Air-Sofwan Modul-11 % go run soal-1.go
masukan kadar pH: 7
Air layak minum
sofwanuhaifaruq@MacBook-Air-Sofwan Modul-11 %
```

A floating window in the center contains student information:

30 November 2025 at 20.22  
Nama : Sofwan Nuha Al Faruq  
NIM : 109082500032  
Kelas : S1-IF-02

### Deskripsi program:

Pada program ini saya membuat untuk menentukan apakah kadar pH pada air yang diinput termasuk Air yang layak untuk diminum atau tidak. Menggunakan variabel ph yang bertipe data float64, dan untuk menyimpan nilai dari ph. Program ini menggunakan struktur switch tanpa ekspresi, sehingga setiap kondisi pH bisa dicek berdasarkan rentang yang sudah ditentukan. Jika nilai pH berada pada kisaran 6.5 sampai 8.6, maka program akan menampilkan bahwa air tersebut layak minum. Jika nilai pH berada di bawah 6.5 atau berada di atas 8.6 hingga maksimal 14, program akan mengeluarkan hasil bahwa air tidak layak minum. Namun, jika nilai pH diinput melebihi batas normal skala pH, yaitu di luar 0 sampai 14, maka program akan memberikan pesan bahwa nilai pH tidak valid. Dengan cara ini, program dapat membantu mengecek kelayakan air dengan sederhana dan langsung berdasarkan nilai pH yang diberikan.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var jam int
    var tarif int
    var total int

    fmt.Print("Masukan jenis kendaraan (motor/mobil/truk) : ")
    fmt.Scan(&kendaraan)
    fmt.Print("Masukan Durasi Parkir (dalam jam) : ")
    fmt.Scan(&jam)

    if jam < 1 {
        jam = 1
    }

    switch kendaraan {
    case "motor":
        tarif = 2000
    case "mobil":
        tarif = 5000
    case "truk":
        tarif = 8000
    default:
        fmt.Println("Jenis Kendaraan Tidak Valid")
        return
    }

    total = tarif * jam

    fmt.Printf("Total Tarif Parkir: Rp %d\n", total)
}
```

## Screenshot program

The screenshot shows a macOS desktop environment. In the foreground, a terminal window is open with the following command and output:

```
sofwanhuafarug@MacBook-Air-Sofwan Modul-11 % clear
Masukan jenis kendaraan (motor/mobil/truk) : 12
Masukan Durasi Parkir (dalam jam) : 5
sofwanhuafarug@MacBook-Air-Sofwan Modul-11 % go run soal-2.go
Masukan jenis kendaraan (motor/mobil/truk) : motor
Masukan Durasi Parkir (dalam jam) : 5
Total Tarif Parkir: Rp 10000
sofwanhuafarug@MacBook-Air-Sofwan Modul-11 %
```

Below the terminal is a code editor window titled "Modul-11". It contains a Go language script named "soal-2.go". The code defines a function to calculate parking fees based on vehicle type and duration.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var jam int
8     var tarif int
9     var total int
10
11     fmt.Println("Masukan jenis kendaraan (motor/mobil/truk) : ")
12     fmt.Scan(&kendaraan)
13     fmt.Println("Masukan Durasi Parkir (dalam jam) : ")
14     fmt.Scan(&jam)
15
16     if jam < 1 {
17         jam = 1
18     }
19
20     switch kendaraan {
21     case "motor":
22         tarif = 2000
23     case "mobil":
24         tarif = 5000
25     case "truk":
26         tarif = 8000
27     default:
28         fmt.Println("Jenis Kendaraan Tidak Valid")
29     }
30 }
```

The code editor interface includes tabs for other files like "guided-1.go", "guided-2.go", "guided-3.go", "soal-1.go", and "soal-3.go". The right sidebar shows a file tree for "MODUL-11" and a "VS CODE PETS" panel.

### Deskripsi program :

Program ini berfungsi untuk menghitung tarif parkir berdasarkan jenis kendaraan dan lama parkir yang diinput pengguna. Program menggunakan variabel kendaraan untuk menyimpan jenis kendaraan, durasi untuk lama parkir dalam jam, dan tarif sebagai biaya per jam sesuai jenis kendaraan. Setelah pengguna memasukkan jenis kendaraan dan lama parkir, program akan menentukan tarif per jam melalui struktur switch. Setiap kendaraan memiliki tarif tetap per jam, lalu total biaya dihitung dari tarif × durasi. Maka, program menampilkan total tarif parkir sesuai yang diinput.

