

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**MODUL 11**  
**MODUL 11. SWITCH-CASE**



**Disusun oleh:**  
**ALMA BONITA MIA WARDHANA**  
**109082500015**  
**S1IF-13-04**

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## LATIHAN KELAS – GUIDED

### 1. Guided 1

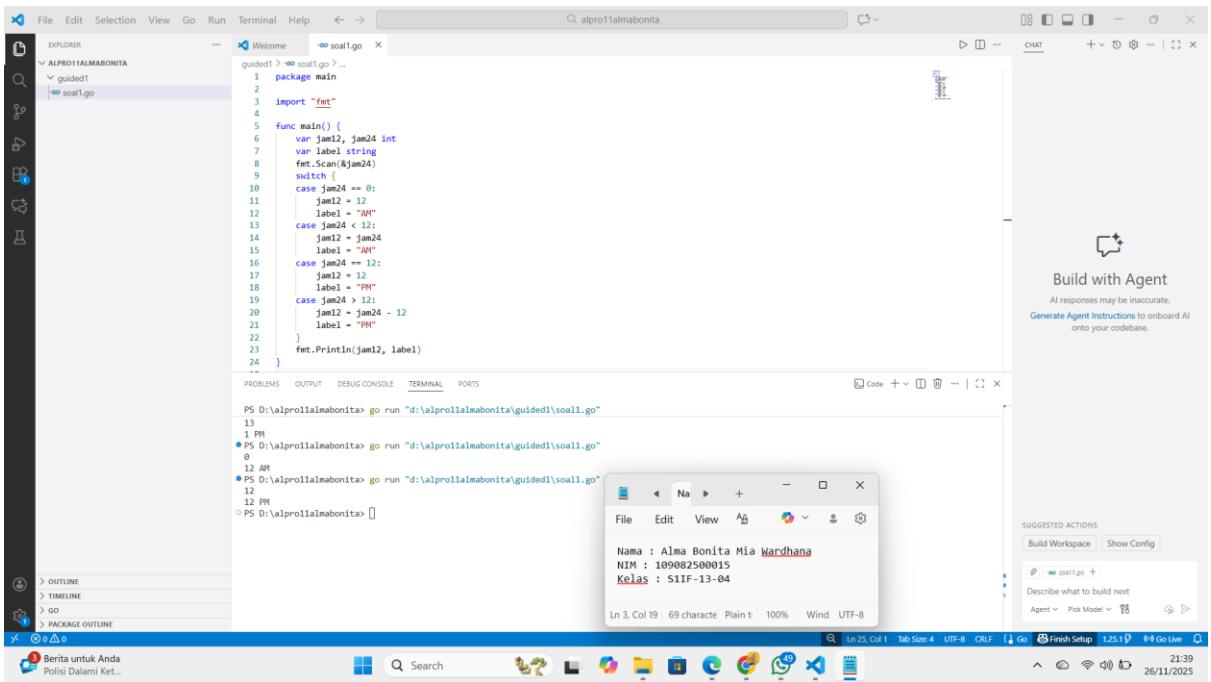
#### Source Code

```
package main

import "fmt"

func main() {
    var jam12, jam24 int
    var label string
    fmt.Scan(&jam24)
    switch {
        case jam24 == 0:
            jam12 = 12
            label = "AM"
        case jam24 < 12:
            jam12 = jam24
            label = "AM"
        case jam24 == 12:
            jam12 = 12
            label = "PM"
        case jam24 > 12:
            jam12 = jam24 - 12
            label = "PM"
    }
    fmt.Println(jam12, label)
}
```

## Screenshot program



## Deskripsi program

Program di atas bekerja dengan cara membaca sebuah angka jam dalam format 24 jam, lalu menggunakan struktur switch untuk menentukan bagaimana nilai jam tersebut harus diubah. Variabel jam24 menampung input awal dari pengguna, sedangkan jam12 dan label akan diisi berdasarkan hasil konversi. Setiap *case* pada switch memeriksa kondisi berbeda: jam24 == 0 mengubah jam menjadi 12 AM, jam24 < 12 langsung menggunakan jam yang sama dengan label AM, jam24 == 12 mengatur jam menjadi 12 PM, dan jam24 > 12 mengurangi jam dengan 12 agar sesuai format 12 jam lalu memberi label PM. Setelah salah satu kondisi terpenuhi, program mencetak hasil konversi dalam format "jam12 label". Program ini menunjukkan penggunaan *switch* untuk memecah logika konversi waktu secara terstruktur dan mudah dibaca.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    var nama_tanaman string
```

```

fmt.Scan(&nama_tanaman)

switch nama_tanaman {

case "nepenthes", "drosera":

    fmt.Println("Termasuk Tanaman Karnivora.")

    fmt.Println("Asli Indonesia.")

case "venus", "sarracenia":

    fmt.Println("Termasuk Tanaman Karnivora.")

    fmt.Println("Tidak Asli Indonesia.")

default:

    fmt.Println("Tidak termasuk Tanaman Karnivora.")

}

}

```

## Screenshot program

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

- Code Editor:** The main pane displays the source code for `soal2.go`. The code uses a switch statement to categorize plants based on their names.
- Terminal:** The terminal window shows command-line interactions. It runs the program with inputs "venus" and "nepenthes", both of which correctly identify them as carnivorous plants from Indonesia.
- Output Window:** A small window in the bottom right corner displays student information: Nama : Alma Bonita Mia Wardhana, NIM : 100082500015, Kelas : S1IF-13-04.

## Deskripsi program

Program di atas membaca nama sebuah tanaman dari input pengguna lalu menggunakan struktur switch untuk menentukan kategori tanaman tersebut. Variabel `nama_tanaman` menampung input, dan melalui switch, program memeriksa apakah nama tersebut cocok dengan salah satu kasus. Jika tanaman adalah “`nepenthes`” atau

“drosera”, program menampilkan bahwa tanaman tersebut termasuk tanaman karnivora dan berasal dari Indonesia. Jika yang dimasukkan adalah “venus” atau “sarracenia”, program tetap menyatakan bahwa itu tanaman karnivora tetapi bukan asli Indonesia. Jika nama tanaman tidak cocok dengan semua *case* yang tersedia, *default* akan dieksekusi untuk memberi tahu bahwa tanaman tersebut bukan tanaman karnivora. Program ini memperlihatkan penggunaan switch dengan beberapa nilai dalam satu *case* untuk mempermudah pengelompokan data berdasarkan kriteria tertentu.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi int
    var tarif int
    fmt.Print("Masukkan jenis kendaraan
(Motor/Mobil/Truk): ")
    fmt.Scan(&kendaraan)
    fmt.Print("Masukkan durasi parkir (dalam jam): ")
    fmt.Scan(&durasi)
    switch {
        case kendaraan == "Motor" && durasi >= 1 && durasi
        <= 2:
            tarif = 7000
        case kendaraan == "Motor" && durasi > 2:
            tarif = 9000
    }
}
```

```
        case kendaraan == "Mobil" && durasi >= 1 && durasi
        <= 2:
            tarif = 15000
        case kendaraan == "Mobil" && durasi > 2:
            tarif = 20000
        case kendaraan == "Truk" && durasi >= 1 && durasi
        <= 2:
            tarif = 25000
        case kendaraan == "Truk" && durasi > 2:
            tarif = 35000
        default:
            fmt.Println("Jenis kendaraan atau durasi parkir
tidak valid")
    }
    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
}
```

### Screenshoot program

```

package main

func main() {
    var kendaraan string
    var durasi int
    var tarif int

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

    switch {
    case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
        tarif = 7000
    case kendaraan == "Motor" && durasi > 2:
        tarif = 9000
    case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
        tarif = 15000
    case kendaraan == "Mobil" && durasi > 2:
        tarif = 20000
    case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
        tarif = 25000
    case kendaraan == "Truk" && durasi > 2:
        tarif = 35000
    default:
        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
    }
    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
}

```

The screenshot shows a Go code editor interface. The code defines a function `main` that takes input for vehicle type and parking duration, then calculates and prints the parking fee using a switch statement. A terminal window below the editor shows the execution of the program and its output for different inputs.

## Deskripsi program

Program di atas berfungsi untuk menentukan tarif parkir berdasarkan jenis kendaraan dan durasi parkir dengan memanfaatkan struktur switch yang menggunakan kondisi logis. Setelah pengguna memasukkan jenis kendaraan dan lama parkir, program mengevaluasi pasangan kondisi pada setiap *case*. Jika kendaraan adalah Motor dan durasi 1–2 jam, tarif ditetapkan menjadi 7.000, sedangkan jika lebih dari 2 jam menjadi 9.000. Untuk Mobil, tarifnya 15.000 untuk durasi 1–2 jam dan 20.000 jika lebih lama. Sementara itu, Truk dikenai tarif 25.000 untuk 1–2 jam dan 35.000 untuk durasi di atas 2 jam. Jika kombinasi input tidak sesuai dengan seluruh kondisi, *default* akan menampilkan pesan bahwa input tidak valid. Program ini menunjukkan bagaimana switch dapat digunakan untuk menangani beberapa kondisi sekaligus secara terstruktur dan mempermudah pengaturan tarif berdasarkan kategori tertentu.

## TUGAS

### 1. Tugas 1

#### Source code

```

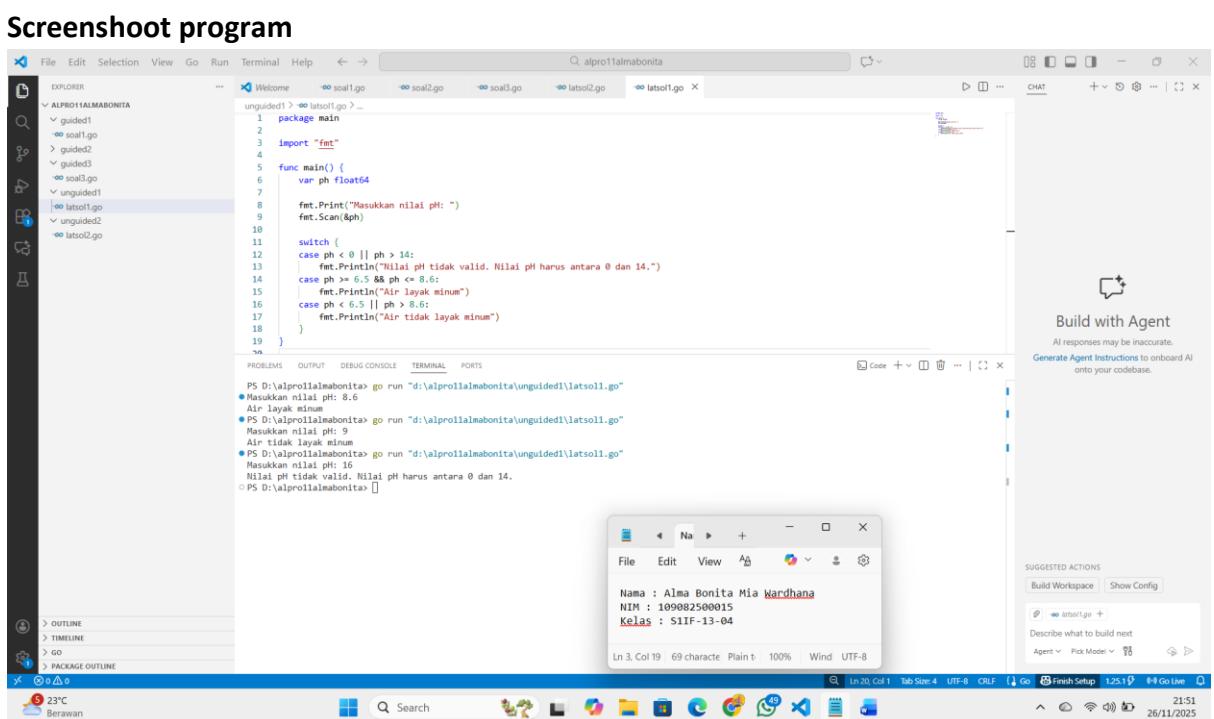
package main

import "fmt"

func main() {
    var ph float64
}
```

```
fmt.Println("Masukkan nilai pH: ")
fmt.Scan(&ph)

switch {
    case ph < 0 || ph > 14:
        fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
    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")
}
```



## Deskripsi program

Program di atas digunakan untuk mengecek kelayakan air berdasarkan nilai pH menggunakan struktur switch dengan kondisi logis. Setelah pengguna memasukkan nilai pH, program memeriksa beberapa kondisi dalam *switch*: jika nilai pH kurang dari

0 atau lebih dari 14, program menampilkan bahwa nilai tersebut tidak valid karena berada di luar rentang pH yang seharusnya. Jika pH berada pada kisaran 6.5 hingga 8.6, air dinyatakan layak minum. Sedangkan jika pH berada di bawah 6.5 atau di atas 8.6, maka air dianggap tidak layak minum. Dengan menggunakan switch tanpa ekspresi, program dapat menangani berbagai kondisi secara lebih rapi dan mudah dipahami.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var jenis string

    var jam int

    fmt.Scan(&jenis)

    fmt.Scan(&jam)

    if jam < 1 {

        jam = 1

    }

    var tarifPerJam int

    switch jenis {

        case "motor":


            tarifPerJam = 2000
```

```

case "mobil":

    tarifPerJam = 5000

case "truk":

    tarifPerJam = 8000

default:

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

return

}

total := tarifPerJam * jam

fmt.Printf("Total biaya parkir: Rp %d\n",
total)

}

```

## Screenshot program

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

- Code Editor:** The main window displays the Go code for calculating parking fees.
- Terminal:** The terminal window shows the command `go run` being used to execute the program, followed by the output which includes the student's name, NIM, and Kelas.
- Tooltip:** A tooltip provides information about the student: Name (Alma Bonita Mia Wardhana), NIM (109082500015), and Kelas (SIIF-13-04).
- IDE UI:** The interface includes standard windows for File, Edit, Selection, View, Go, Run, Terminal, Help, and various toolbars.

### **Deskripsi program**

Program di atas digunakan untuk menghitung total biaya parkir berdasarkan jenis kendaraan dan lamanya waktu parkir. Pengguna memasukkan jenis kendaraan dan jumlah jam, lalu program memastikan bahwa durasi parkir minimal dihitung sebagai 1 jam. Setelah itu, melalui struktur switch, program menentukan tarif per jam sesuai jenis kendaraan: motor dikenakan 2.000, mobil 5.000, dan truk 8.000. Jika input tidak cocok dengan salah satu jenis kendaraan tersebut, program menampilkan pesan bahwa jenis kendaraan tidak valid dan menghentikan proses. Setelah tarif per jam ditetapkan, program menghitung total biaya dengan mengalikan tarif per jam dengan jumlah jam, lalu menampilkan hasilnya. Program ini menunjukkan penggunaan switch untuk memilih tarif berdasarkan kategori input secara sederhana dan jelas.

### **3. Tugas 3**

#### **Source code**

```
package main

import "fmt"

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

    switch {
    case n%2 != 0:
        fmt.Println("Kategori: Bilangan Ganjil")
        hasil := n + (n + 1)
        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n", n, n+1, hasil)

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

    case n%5 == 0 && n%10 != 0:
        fmt.Println("Kategori: Bilangan Kelipatan 5")
```

```

        hasil := n * n
        fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n",
n, hasil)

        case n%10 == 0:
            fmt.Println("Kategori: Bilangan Kelipatan 10")
            hasil := n / 10
            fmt.Printf("Hasil pembagian antara %d / 10 =
%d\n", n, hasil)
        }
    }
}

```

### Screenshot program

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

- File Explorer:** Shows files in the 'ALPRO11ALMABONITA' directory, including 'guided1', 'guided2', 'guided3', 'unguided1', 'unguided2', 'unguided3', and 'latso1.go' through 'latso3.go'.
- Code Editor:** Displays the source code for 'latso3.go'.
- Terminal:** Shows command-line interactions in a Go workspace. It includes commands like 'go run' and outputs related to user input and program execution.
- Suggested Actions:** A sidebar provides AI-related suggestions such as 'Build with Agent' and 'Generate Agent Instructions'.

### Deskripsi program

Program di atas digunakan untuk mengelompokkan sebuah bilangan ke dalam beberapa kategori menggunakan struktur switch dengan kondisi logis. Setelah pengguna memasukkan nilai n, program memeriksa berbagai kondisi: jika n tidak habis dibagi 2, program menggolongkannya sebagai bilangan ganjil dan menghitung penjumlahan antara n dan bilangan berikutnya. Jika n genap tetapi bukan kelipatan 5, program mengklasifikasikannya sebagai bilangan genap dan melakukan perkalian dengan bilangan berikutnya. Jika n merupakan kelipatan 5 namun bukan kelipatan 10, program menghitung kuadrat dari bilangan tersebut.

Terakhir, jika n adalah kelipatan 10, program membaginya dengan 10. Setiap *case* menampilkan kategori bilangan beserta hasil operasi yang sesuai. Program ini memperlihatkan bagaimana switch dapat digunakan untuk menangani berbagai kondisi matematika secara terstruktur dan mudah dipahami.