

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

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



**Disusun oleh:**  
**FAREL JULIYANDRA RESTU HERMAWAN**  
**109082530038**  
**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 jam int

    var jamFormat int
    var perihal string

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

    switch jam
    { case 0:

        jamFormat = 12
        perihal = "AM"

    case 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11:

        jamFormat =
        jam perihal =
        "AM"

    case 12:

        jamFormat = 12
        perihal = "PM"

    case 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23:

        jamFormat = jam - 12
        perihal = "PM"

    default:

        fmt.Println("Jam tidak valid! Masukkan jam 0-
23")
    }
}
```

```

        return

    }

    fmt.Printf("%d:00 = %d %s\n", jam, jamFormat,
               perihal)

}

```

## Screenshot program

The screenshot shows a Go development environment with three main windows:

- Editor:** Displays the source code of `guided1.go`. The code defines a package named `main` with a `main` function. It reads a value from `jam` (int) and prints it in a 12-hour format (`jam12`) along with its AM/PM label (`label`). The switch statement handles values from 0 to 23, mapping them to 12-hour format and AM/PM labels.
- Terminal:** Shows the command `go run F:\ALIPRO\modul_11\guided1.go` being run, followed by the output: 12:00 PM.
- Output:** A window titled "Console" showing the output of the program: MAMA : FAREL JULIYANDRA RESTU HERMANAWI, NIM : 199802538038, KELAS : IF-13-04.

## Deskripsi program

Program Go ini berfungsi untuk mengkonversi jam dalam format 24 jam (0–23) menjadi format 12 jam dengan keterangan AM atau PM. Pengguna diminta memasukkan sebuah angka jam melalui input keyboard, kemudian program menyimpan nilai tersebut dalam variabel `jam`. Melalui pernyataan `switch`, program menentukan nilai `jamFormat` dan perihal (AM/PM): jika `jam` 0 akan ditampilkan sebagai 12 AM, `jam` 1–11 tetap sama dengan label AM, `jam` 12 menjadi 12 PM, dan `jam` 13–23 dikonversi menjadi `jamFormat = jam - 12` dengan label PM. Jika input di luar rentang 0–23, program menampilkan pesan bahwa jam tidak valid dan langsung berhenti. Terakhir, program menampilkan hasil konversi dalam format `<jam_awal>:00 = <jamFormat> <AM/PM>`.

## 2. Guided 2 Source Code

```
package main
import "fmt"

func main() {
    var Tanaman string
    var keluaran
    string

    fmt.Print("Masukkan nama tanaman: ")
    fmt.Scan(&Tanaman)

    switch Tanaman {
    case "nepenthes":
        keluaran = "Termasuk Tanaman Karnivora\nAsli Indonesia"
    case "venus":
        keluaran = "Termasuk Tanaman Karnivora\nBukan Asli Indonesia"
    case "karedok":
        keluaran = "Tidak termasuk Tanaman Karnivora"
    default:
        keluaran = "tidak diketahui"
    }

    fmt.Println(keluaran)
}
```

### Screenshot program

The screenshot shows a Go development environment. On the left, there's a file browser window titled 'GUIDED 3' showing files like 'guided 3.go', 'guided 3.go', 'guided 3.go', 'guided 3.go', and 'guided 3.go'. In the center, there's a code editor window with the following Go code:

```
package main
import "fmt"
func main() {
    var nama_tanaman string
    fmt.Scan(&nama_tanaman)
    switch nama_tanaman {
    case "nepenthes", "drosaria":
        fmt.Println("Termasuk Tanaman Karnivora dan Asli Indonesia")
    case "venus", "venusiana":
        fmt.Println("Termasuk Tanaman Karnivora, Tidak Asli Indonesia")
    default:
        fmt.Println("Tidak termasuk Tanaman Karnivora")
    }
}
```

To the right, there's a terminal window showing command-line interactions:

```
PS F:\VALPRO\ws\udl_1> go run ./valpro\guided_3
nepenthes
Termasuk Tanaman Karnivora dan Asli Indonesia
PS F:\VALPRO\ws\udl_1> go run ./valpro\guided_3
venus
Termasuk Tanaman Karnivora, Tidak Asli Indonesia
PS F:\VALPRO\ws\udl_1> go run ./valpro\guided_3
karedok
Tidak termasuk Tanaman Karnivora
PS F:\VALPRO\ws\udl_1>
```

Below the terminal is a small window showing student information:

NAMA	: FARREL JULIYANDRA RESTU HERMAN
NIM	: 189682530038
KELAS	: IF-13-04

### Deskripsi program

Program Go ini meminta pengguna memasukkan nama tanaman melalui keyboard dan menyimpannya ke dalam variabel Tanaman. Setelah itu, program menggunakan struktur kontrol switch untuk menentukan informasi tentang tanaman tersebut: jika input adalah "nepenthes", program menyatakan bahwa tanaman itu termasuk tanaman karnivora dan asli Indonesia; jika "venus", termasuk tanaman karnivora tetapi bukan asli Indonesia; jika "karedok", dinyatakan bukan tanaman karnivora; sedangkan untuk nama lain yang tidak dikenali, program menampilkan bahwa tanaman tersebut "tidak diketahui" melalui variabel keluaran.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var jenisKendaraan string

    var durasi int

    var tarif int

    fmt.Print("Masukkan jenis kendaraan
(Motor/Mobil/Truk): ")

    fmt.Scan(&jenisKendaraan)
```

```
fmt.Print("Masukkan durasi parkir (dalam jam):\n") fmt.Scan(&durasi)

if durasi < 1 {
    fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
    fmt.Printf("Tarif Parkir: Rp\n") return
}

switch {
    case jenisKendaraan == "Motor" && durasi >= 1 && durasi <= 2:
        tarif = 7000
    case jenisKendaraan == "Motor" && durasi > 2: tarif = 9000

    case jenisKendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
        tarif = 15000
    case jenisKendaraan == "Mobil" && durasi > 2: tarif = 20000

    case jenisKendaraan == "Truk" && durasi >= 1 && durasi <= 2:
        tarif = 25000
    case jenisKendaraan == "Truk" && durasi > 2: tarif = 35000

    default:
        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
        fmt.Printf("Tarif Parkir: Rp\n") return
}
fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
```

```
}
```

## Screenshoot program

The screenshot shows a Go development environment with several windows:

- EDITOR:** Shows the source code of `guided 3.go`. The code defines a package `main` with a `func main()` block. It prompts the user for vehicle type and parking duration, then calculates and prints the tariff based on a switch statement.
- TERMINAL:** Shows the command `go run "F:\VALPRO\modul 11\guided 3.go"` being run, followed by the output: "Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor".
- OUTPUT:** Shows the command `go run "F:\VALPRO\modul 11\guided 3.go"` being run again, followed by the output: "Masukkan durasi parkir (dalam jam): 2 Tarif Parkir: Rp. 7000".
- INFO:** A small window showing student information: NAMA : FARREL JULIYANDRA RESTU HERMANAN, NIM : 189882530838, KELAS : IF-13-04.

## Deskripsi program

Program Go ini menghitung tarif parkir berdasarkan jenis kendaraan dan lama parkir. Pengguna diminta memasukkan jenis kendaraan (Motor/Mobil/Truk) dan durasi parkir dalam jam. Jika durasi kurang dari 1 jam, program langsung menyatakan data tidak valid dan tarif Rp 0. Dengan switch, program menentukan tarif: Motor Rp 7000 (1–2 jam) atau Rp 9000 (>2 jam), Mobil Rp 15000 (1–2 jam) atau Rp 20000 (>2 jam), dan Truk Rp 25000 (1–2 jam) atau Rp 35000 (>2 jam). Jika jenis kendaraan tidak sesuai, program juga menampilkan data tidak valid dan tarif Rp 0.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
```

```
var pH float64

fmt.Print("Masukkan kadar pH: ")

fmt.Scan(&pH)

if pH < 0 || pH > 14 {

    fmt.Println("Input tidak valid, rentang pH 0 - 14")

    return
}

var kategori int

switch {

case pH < 6.5:

    kategori = 0

case pH >= 6.5 && pH <= 8.6:

    kategori = 1

case pH > 8.6:

    kategori = 2

}

switch kategori {

case 0:

    fmt.Println("Air tidak layak minum")

case 1:

    fmt.Println("Air layak minum")
```

```

case 2:

    fmt.Println("Air tidak layak minum")

}

}

```

### Screenshoot program

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

- EXPLORER:** Shows a file tree with files like `guided1.go`, `guided2.go`, `guided3.go`, `labeled1.go`, `labeled2.go`, and `labeled3.go`.
- CODE EDITOR:** Displays the source code for `labeled1.go`. The code reads a pH value from the user, checks if it's within the range [0, 14], and then classifies it into three categories based on its value. Category 0 (pH < 6.5) prints "Air tidak layak minum". Category 1 (6.5 ≤ pH < 8.6) prints "Air layak minum". Category 2 (pH ≥ 8.6) prints "Air tidak layak minum".
- TERMINAL:** Shows the command `go run ./labeled1` being run, followed by several test runs with different pH values (9, 10, 11, 12, 13, 14) and their corresponding outputs.
- OUTPUT:** Shows the application's output window displaying student information: NAMA : FAREL JULIYANDRA RESTU HERMAWAN, NIM : 109882538838, KELAS : IF-13-84.

### Deskripsi program

Program Go ini digunakan untuk menentukan kelayakan air minum berdasarkan nilai pH. Pengguna diminta memasukkan kadar pH, lalu program mengecek apakah nilainya berada di rentang 0–14; jika di luar itu, ditampilkan pesan “Input tidak valid” dan program berhenti. Jika valid, program mengelompokkan pH ke dalam variabel kategori: pH < 6.5 (kategori 0), 6.5–8.6 (kategori 1), dan > 8.6 (kategori 2). Berdasarkan kategori tersebut, program mencetak apakah air “layak minum” (hanya kategori 1) atau “tidak layak minum” (kategori 0 dan 2).

## 2. Tugas 2

### Source code

```

package main

import "fmt"

```

```
func main() {  
  
    var jenisKendaraan string  
  
    var durasi int  
  
    var tarifPerJam, totalBiaya int  
  
  
    fmt.Print("Masukkan jenis kendaraan  
(motor/mobil/truk): ")  
  
    fmt.Scan(&jenisKendaraan)  
  
  
    fmt.Print("Masukkan durasi parkir (jam): ")  
  
    fmt.Scan(&durasi)  
  
  
    if durasi < 1 {  
  
        durasi = 1  
  
    }  
  
  
    switch jenisKendaraan {  
  
        case "motor":  
  
            tarifPerJam = 2000  
  
        case "mobil":  
  
            tarifPerJam = 5000  
  
        case "truk":  
  
            tarifPerJam = 8000  
  
        default:  
  
            fmt.Println("Jenis kendaraan tidak valid")  
    }  
}
```

```

        return

    }

    totalBiaya = tarifPerJam * durasi

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

}

```

### Screenshot program

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

- Project Explorer:** Shows files in the "MODULE\_11" directory: parkir1.go, parkir2.go, parkir3.go, lama1.go, lama2.go, and lama3.go.
- Code Editor:** Displays the source code for "parkir1.go". The code defines a function `main` that prompts the user for vehicle type and parking duration, calculates the total cost, and prints it. It includes a switch statement for vehicle types (motor, mobil, truk) with rates of 2000, 5000, and 8000 respectively, and a default case for invalid input.
- Terminal:** Shows command-line interactions. The first two runs ask for vehicle type and duration, then print the total cost. The third run asks for vehicle type, prints an error message, and then asks for duration, printing the total cost again.
- Browser:** Shows a simple HTML page with student information: NAME: FAREL JULIYANDRA RESTU HERMANI, NIM: 189882538038, and KELAS: IF-13-B4.

### Deskripsi program

Program Go ini menghitung total biaya parkir berdasarkan jenis kendaraan dan durasi parkir. Pengguna diminta memasukkan jenis kendaraan (motor/mobil/truk) dan lama parkir dalam jam. Jika durasi yang diinput kurang dari 1 jam, program otomatis mengubahnya menjadi 1 jam. Dengan menggunakan switch, program menentukan tarif per jam: motor Rp 2000, mobil Rp 5000, dan truk Rp 8000. Jika jenis kendaraan tidak sesuai dengan ketiga pilihan tersebut, program menampilkan pesan "Jenis kendaraan tidak valid" dan berhenti. Total biaya dihitung dengan mengalikan tarif per jam dengan durasi parkir, kemudian hasilnya ditampilkan dalam format rupiah.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {
    var bilangan int

    fmt.Print("Masukkan bilangan bulat: ")
    fmt.Scan(&bilangan)

    var kategori int
    switch {
    case bilangan%10 == 0:
        kategori = 1
    case bilangan%5 == 0:
        kategori = 2
    case bilangan%2 == 1 || bilangan%2 == -1:
        kategori = 3
    case bilangan%2 == 0:
        kategori = 4
    }

    switch kategori {
    case 1:
        hasil := bilangan / 10
        fmt.Printf("Kategori: Bilangan Kelipatan
10\n")
        fmt.Printf("Hasil pembagian antara %d / 10 =
%d\n", bilangan, hasil)
    case 2:
        hasil := bilangan * bilangan
        fmt.Printf("Kategori: Bilangan Kelipatan
5\n")
        fmt.Printf("Hasil kuadrat dari %d ^2 =
%d\n", bilangan, hasil)
    case 3:
        hasil := bilangan + (bilangan + 1)
```

```

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

    case 4:
        hasil := bilangan * (bilangan + 1)
        fmt.Printf("Kategori: Bilangan Genap\n")
        fmt.Printf("Hasil perkalian dengan bilangan
berikutnya %d * %d = %d\n", bilangan, bilangan+1,
hasil)
    }
}

```

## Screenshot program

The screenshot shows a Go code editor interface with a file named `bilangan.go` open. The code implements a function to categorize a given number based on its properties (odd/even, divisible by 10, square, etc.) and prints the results. Below the code editor is a terminal window displaying the program's output for a user input of 10. The status bar at the bottom right provides information about the current file and session.

```

package main

import "fmt"

func main() {
    var bilangan int

    fmt.Println("Masukkan bilangan bulat: ")
    fmt.Scan(&bilangan)

    var kategori int
    switch {
    case bilangan%10 == 0:
        kategori = 1
    case bilangan%5 == 0:
        kategori = 2
    case bilangan%2 == 0 || bilangan%2 == -1:
        kategori = 3
    case bilangan%3 == 0:
        kategori = 4
    }

    switch kategori {
    case 1:
        hasil := bilangan / 10
        fmt.Printf("Kategori: Bilangan Kelipatan 10\n")
        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", bilangan, hasil)
    case 2:
        hasil := bilangan * bilangan
        fmt.Printf("Kategori: Bilangan Kelipatan 5\n")
        fmt.Printf("Hasil kuadrat dari %d * %d = %d\n", bilangan, hasil)
    case 3:
        hasil := bilangan * (bilangan + 1)
        fmt.Printf("Kategori: Bilangan genap\n")
    }
}

```

Terminal Output:

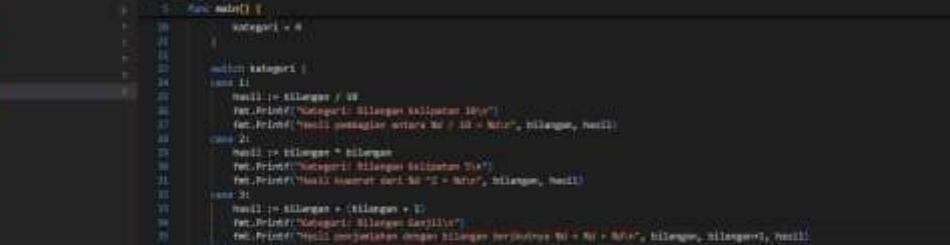
```

Masukkan bilangan bulat: 10
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 10 / 10 = 1

```

Status Bar:

- File: bilangan.go
- Line: 14
- Column: 17
- Character: 14 characters
- Mode: Plain
- Font: 100%
- Width: 1280px
- Height: 720px
- Encoding: UTF-8



```
public class Kategori {
    public static void main(String[] args) {
        int[] bilangan = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
        System.out.println("Rata-rata bilangan bulat: " + rataRata(bilangan));
        System.out.println("Median bilangan bulat: " + median(bilangan));
    }

    public static double rataRata(int[] arr) {
        double sum = 0;
        for (int i : arr) {
            sum += i;
        }
        return sum / arr.length;
    }

    public static int median(int[] arr) {
        Arrays.sort(arr);
        int n = arr.length;
        if (n % 2 == 0) {
            return (arr[n / 2] + arr[n / 2 - 1]) / 2;
        } else {
            return arr[n / 2];
        }
    }
}
```

## Deskripsi program

Program Go ini mengklasifikasikan sebuah bilangan bulat ke beberapa kategori lalu melakukan operasi berbeda sesuai kategorinya. Setelah pengguna memasukkan bilangan, switch pertama menentukan kategori: jika kelipatan 10 → kategori 1, jika kelipatan 5 → kategori 2, jika ganjil (termasuk negatif ganjil) → kategori 3, dan jika genap → kategori 4. switch kedua menjalankan operasi: kategori 1 membagi bilangan dengan 10, kategori 2 menghitung kuadrat, kategori 3 menjumlahkan bilangan dengan bilangan berikutnya, kategori 4 mengalikan bilangan dengan bilangan berikutnya, lalu menampilkan hasilnya.