

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

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



**Disusun oleh:**

**Abyan Fito Arrasyid**

**109082500086**

**S1IF-13-04**

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

## 1. Guided Laprak Modul 11

### 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(jam, "AM")
        case jam == 12:
            fmt.Println("12 PM")
        default:
            fmt.Println(jam-12, "PM")
    }
}
```

## Screenshoot program

The screenshot shows the VS Code interface with the following details:

- Editor:** The code editor displays a Go file named `Jam.go`. The code defines a `main` package with a `main` function. It reads an integer from the user, prints it as "AM" if less than 12, or as "PM" if greater than or equal to 12. If the input is 0, it prints "12 AM".
- Terminal:** The terminal window shows the execution of the program. It runs `go run` on the file, and the output is:
  - For input 13, the output is "1 PM".
  - For input 0, the output is "12 AM".
- Status Bar:** The status bar at the bottom indicates there are 2 problems.

## Deskripsi program

Program ini membaca sebuah angka jam dalam format 24 jam, lalu mengubahnya menjadi format 12 jam dengan AM atau PM. Nilai jam dimasukkan oleh pengguna. Jika jam bernilai 0, program menampilkan “12 AM” karena 0 pada format 24 jam adalah tengah malam. Jika jam kurang dari 12, program menampilkan angka jam tersebut dengan akhiran AM. Jika jam bernilai 12, program menampilkan “12 PM” karena itu adalah tengah hari. Untuk jam di atas 12, program mengurangi jam dengan 12 dan menampilkannya sebagai PM. Program ini menggunakan struktur switch tanpa ekspresi untuk mengecek kondisi satu per satu.

## 2. Guided Laprak Modul 11

### Source code

```
package main

import "fmt"

func main() {

    var tanaman string

    fmt.Scan(&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 Karnivora")

    }

}
```

## Screenshot program

The screenshot shows a terminal window with the following content:

```
C: > Users > abyan > OneDrive > Documents > Vscode > Tanaman.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var tanaman string
7     fmt.Scan(&tanaman)
8
9     switch tanaman {
10    case "nepenthes":
11        fmt.Println("Termasuk Tanaman Karnivora \nAsli Indonesia")
12    case "venus":
13        fmt.Println("Termasuk Tanaman Karnivora \nBukan Asli Indonesia")
14    default:
15        fmt.Println("Tidak termasuk Tanaman Karnivora")
16    }
17 }
18

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\Tanaman.go"
nepenthes
PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\Tanaman.go"
venus
Termasuk Tanaman Karnivora
Bukan Asli Indonesia
PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\Tanaman.go"
karedok
Tidak termasuk Tanaman Karnivora
PS C:\Users\abyan>
```

## Deskripsi program

Program ini membaca sebuah input berupa nama tanaman dari pengguna. Nilai tersebut disimpan dalam variabel tanaman. Setelah itu, program menggunakan struktur switch untuk mengecek apakah nama tanaman cocok dengan salah satu kasus yang tersedia. Jika input adalah “nepenthes”, program menampilkan bahwa tanaman tersebut termasuk tanaman karnivora dan berasal dari Indonesia. Jika input adalah “venus”, program menampilkan bahwa tanaman itu juga termasuk tanaman karnivora tetapi bukan asli Indonesia. Jika input tidak cocok dengan kedua pilihan tersebut, program menampilkan bahwa tanaman tersebut tidak termasuk tanaman karnivora.

### 3. Guided Laprak Modul 11

#### Source Code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi int
    fmt.Println("Masukkan jenis kendaraan (Motor/mobil/truk) : ")
    fmt.Scan(&kendaraan)
    fmt.Println("Masukkan durasi parkir (dalam jam) : ")
    fmt.Scan(&durasi)

    switch kendaraan {
        case "Motor":
            if durasi >= 1 && durasi <= 2 {
                fmt.Println("Tarif parkir : Rp 7000")
            } else if durasi > 2 {
                fmt.Println("Tarif parkir : Rp 9000")
            }
        case "Mobil":
            if durasi >= 1 && durasi <= 2 {
                fmt.Println("Tarif parkir : Rp 15.000")
            } else if durasi > 2 {
                fmt.Println("Tarif parkir : Rp 20.000")
            }
        case "Truk":
            if durasi >= 1 && durasi <= 2 {
                fmt.Println("Tarif parkir : Rp 25.000")
            } else if durasi > 2 {
                fmt.Println("Tarif parkir : Rp 35.000")
            }
        default:
            fmt.Println("Jenis kendaraan atau durasi parkir tidak valid \nTarif parkir : Rp 0")
    }
}
```

## Screenshot Program

The screenshot shows a VS Code interface with a terminal window open. The terminal shows the execution of a Go program named Parkir.go. The program prompts the user for the type of vehicle (Motor, Mobil, or Truk) and the duration of parking (in hours). It then calculates and prints the parking fee based on the input. If the input is invalid, it prints a message indicating that the input is not valid and sets the fee to 0.

```
C:\> Users > abyan > OneDrive > Documents > Vscode > Parkir.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var durasi int
8     fmt.Print("Masukkan jenis kendaraan (Motor/mobil/truk) : ")
9     fmt.Scan(&kendaraan)
10    fmt.Print("Masukkan durasi parkir (dalam jam) : ")
11    fmt.Scan(&durasi)
12
13    switch kendaraan {
14        case "Motor":
15            if durasi >= 1 && durasi <= 2 {
16                fmt.Println("Tarif parkir : Rp 7000")
17            } else if durasi > 2 {
18                fmt.Println("Tarif parkir : Rp 9000")
19            }
20        case "Mobil":
21            if durasi >= 1 && durasi <= 2 {
22                fmt.Println("Tarif parkir : Rp 15.000")
23            } else if durasi > 2 {
24                fmt.Println("Tarif parkir : Rp 20.000")
25            }
26        case "Truk":
27            if durasi >= 1 && durasi <= 2 {
28                fmt.Println("Tarif parkir : Rp 25.000")
29            } else if durasi > 2 {
30                fmt.Println("Tarif parkir : Rp 35.000")
31            }
32        default:
33            fmt.Println("Jenis kendaraan atau durasi parkir tidak valid \nTarif parkir : Rp 0")
34    }
35 }
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscod Parkir.go"
Masukkan jenis kendaraan (Motor/mobil/truk) : Mobil
Masukkan durasi parkir (dalam jam) : 4
Tarif parkir : Rp 20.000
PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscod Parkir.go"
Masukkan jenis kendaraan (Motor/mobil/truk) : Motor
Masukkan durasi parkir (dalam jam) : 3
Tarif parkir : Rp 9000
PS C:\Users\abyan>
```

## Deskripsi Program

Program ini meminta pengguna memasukkan jenis kendaraan dan durasi parkir dalam jam. Jenis kendaraan disimpan sebagai string, sedangkan durasi disimpan sebagai angka. Setelah itu, program menggunakan switch untuk menentukan tarif parkir berdasarkan jenis kendaraan.

Jika kendaraan adalah Motor, tarifnya Rp 7000 untuk durasi 1–2 jam, dan Rp 9000 jika lebih dari 2 jam. Jika kendaraan adalah Mobil, tarifnya Rp 15.000 untuk 1–2 jam, dan Rp 20.000 jika lebih dari 2 jam. Jika kendaraan adalah Truk, tarifnya Rp 25.000 untuk 1–2 jam, dan Rp 35.000 jika lebih dari 2 jam. Jika input tidak cocok dengan ketiga jenis kendaraan tersebut, program akan menampilkan bahwa input tidak valid dan menetapkan tarif Rp 0.

## 1. Soal Latihan Modul 11

### 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 >= 0 && ph < 6.5 || ph > 8.6 && ph <= 14:
        fmt.Println("Air tidak layak minum")
    default:
        fmt.Println("Nilai pH tidak valid. Nilai pH harus
antara 0 dan 14")
    }
}
```

### Screenshot Program

The screenshot shows a terminal window with the following content:

```
C:\> Users > abyan > OneDrive > Documents > Vscode > Ph air.go > ...
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 >= 0 && ph < 6.5 || ph > 8.6 && ph <= 14:
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 }
```

Below the code, the terminal shows the execution of the program:

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\Ph air.go"
8.6
Air layak minum
PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\Ph air.go"
● 9
Air tidak layak minum
● PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\Ph air.go"
16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14
✧ PS C:\Users\abyan>
```

## **Deskripsi Program**

Program ini membaca nilai pH dari input pengguna dalam bentuk angka desimal. Setelah nilai pH dimasukkan, program menggunakan switch tanpa ekspresi untuk mengecek kondisi tertentu.

Jika nilai pH berada dalam rentang 6.5 hingga 8.6, program menampilkan air layak minum. Jika nilai pH berada di bawah 6.5 atau di atas 8.6 tetapi masih dalam rentang 0 sampai 14, program menampilkan bahwa air tidak layak minum. Jika nilai pH di luar rentang 0 sampai 14, program menampilkan bahwa nilai pH tidak valid karena pH hanya boleh berada antara 0 hingga 14.

## 2. Soal Latihan Modul 11

### Source Code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var motor, mobil, truk, durasi int
    fmt.Scan(&kendaraan, &durasi)

    motor = durasi * 2000
    mobil = durasi * 5000
    truk = durasi * 8000

    switch kendaraan {
        case "motor":
            fmt.Println("Rp", motor)
        case "mobil":
            fmt.Println("Rp", mobil)
        case "truk":
            fmt.Println("Rp", truk)
    }
}
```

## Screenshot Program

The screenshot shows a VS Code interface with a terminal window. The terminal displays Go code for calculating parking fees based on vehicle type and duration. It includes imports for `fmt`, defines a `main` function, reads input for vehicle type and duration, and uses a switch statement to calculate and print the fee for each vehicle type (motor, mobil, truk) at rates of 2000, 5000, and 8000 respectively per hour.

```
C: > Users > abyans > OneDrive > Documents > Vscode > tarif parkir.go > main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var motor, mobil, truk, durasi int
8     fmt.Scan(&kendaraan, &durasi)
9
10    motor = durasi * 2000
11    mobil = durasi * 5000
12    truk = durasi * 8000
13
14    switch kendaraan {
15        case "motor":
16            fmt.Println("Rp", motor)
17        case "mobil":
18            fmt.Println("Rp", mobil)
19        case "truk":
20            fmt.Println("Rp", truk)
21    }
22 }
23
```

Below the code, the terminal shows the execution of the program with three different inputs:

- PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\tarif parkir.go"  
motor 3 jam  
Rp 6000
- PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\tarif parkir.go"  
mobil 1 jam  
Rp 5000
- PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\tarif parkir.go"  
truk 5 jam  
Rp 40000

## Deskripsi Program

Program ini menerima dua input dari pengguna, yaitu jenis kendaraan dan durasi parkir dalam jam. Setelah data dimasukkan, program menghitung tarif parkir berdasarkan jenis kendaraan dengan mengalikan durasi dengan tarif per jam untuk masing-masing kendaraan. Tarif motor dihitung dengan  $\text{durasi} \times 2000$ , mobil dengan  $\text{durasi} \times 5000$ , dan truk dengan  $\text{durasi} \times 8000$ .

Setelah perhitungan selesai, program menggunakan struktur `switch` untuk menampilkan tarif sesuai jenis kendaraan yang dimasukkan. Jika input adalah “motor”, program menampilkan hasil perhitungan tarif motor. Jika “mobil”, program menampilkan tarif mobil. Jika “truk”, program menampilkan tarif truk.

### 3. Soal Latihan Modul 11

#### Source Code

```
package main
import "fmt"
func main() {
    var bilangan int
    fmt.Scan(&bilangan)

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

    case bilangan%5 == 0 && bilangan != 5:
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d\n",
bilangan, bilangan*bilangan)

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

    case (bilangan-1)%2 == 0:
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n", bilangan, bilangan+1,
bilangan+(bilangan+1))
    }
}
```

## Screenshot Program

The screenshot shows a VS Code interface with a terminal window open. The terminal displays Go code for categorizing numbers based on their divisibility by 10, 5, or 2. It also shows the execution of the program with various inputs and the resulting output categories (Bilangan Ganjil, Bilangan Kelipatan 5, Bilangan Genap).

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7     fmt.Scan(&bilangan)
8
9     switch {
10    case bilangan%10 == 0:
11        fmt.Println("Kategori: Bilangan Kelipatan 10")
12        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", bilangan, bilangan/10)
13
14    case bilangan%5 == 0 && bilangan != 5:
15        fmt.Println("Kategori: Bilangan Kelipatan 5")
16        fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d\n", bilangan, bilangan*bilangan)
17
18    case bilangan%2 == 0:
19        fmt.Println("Kategori: Bilangan Genap")
20        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", bilangan, bilangan+1, bilangan*(bilangan+1))
21
22    case (bilangan-1)%2 == 0:
23        fmt.Println("Kategori: Bilangan Ganjil")
24        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", bilangan, bilangan+1, bilangan+(bilangan+1))
25
26 }
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\pola aritmatika.go" 5  
Kategori: Bilangan Ganjil  
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
- PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\pola aritmatika.go" 8  
Kategori: Bilangan Genap  
Hasil perkalian dengan bilangan berikutnya 8 \* 9 = 72
- PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\pola aritmatika.go" 25  
Kategori: Bilangan Kelipatan 5  
Hasil kuadrat dari 25 ^ 2 = 625
- PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\pola aritmatika.go" 20  
Kategori: Bilangan Genap  
Hasil perkalian dengan bilangan berikutnya 8 \* 9 = 72
- PS C:\Users\abyan> go run "c:\Users\abyan\OneDrive\Documents\Vscode\pola aritmatika.go" 25  
Kategori: Bilangan Kelipatan 5  
Hasil kuadrat dari 25 ^ 2 = 625  
Kategori: Bilangan Genap

## Deskripsi Program

Program ini membaca sebuah bilangan integer dari pengguna, lalu menentukan kategori bilangan tersebut menggunakan beberapa kondisi dalam switch tanpa ekspresi.

Jika bilangan habis dibagi 10, program mengategorikannya sebagai bilangan kelipatan 10 dan menampilkan hasil pembagian bilangan tersebut dengan 10. Jika bukan kelipatan 10 tetapi habis dibagi 5 dan tidak sama dengan 5, program mengategorikannya sebagai bilangan kelipatan 5 dan menampilkan hasil kuadrat dari bilangan itu. Jika bukan keduanya tetapi bilangan genap (habis dibagi 2), program menampilkan bahwa bilangan tersebut adalah bilangan genap dan menampilkan hasil perkalian antara bilangan itu dengan bilangan setelahnya. Jika tidak termasuk kategori sebelumnya, program menganggapnya sebagai bilangan ganjil dan menampilkan hasil penjumlahan antara bilangan tersebut dengan bilangan berikutnya.

Program mengevaluasi kondisi secara berurutan, dan hanya kondisi pertama yang terpenuhi yang akan dijalankan. “||” ini disebut operator “OR” digunakan bila salah satu benar. Contoh, (True || False = True), sebaliknya (False || False = False)