

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

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



**Disusun oleh:**

**Raga Prasetyo**

**109082500028**

**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

```

alpro_11.go > guided_1.go
1 package main
2 import "fmt"
3
4 func main() {
5
6     var jam12, jam24 int
7     var label string
8
9     fmt.Scan(&jam24)
10
11    switch {
12    case jam24 == 0:
13        jam12 = 12
14        label = "AM"
15
16    case jam24 < 12:
17        jam12 = jam24
18        label = "AM"
19
20    case jam24 == 12:
21        jam12 = 12
22        label = "PM"
23
24    case jam24 > 12:
25        jam12 = jam24 - 12
26        label = "PM"
}

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```

PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_1.go"
13
1 PM
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_1.go"
0
12 AM
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_1.go"
12
12 PM
PS C:\Users\HP\.vscode>

```

## **Deskripsi program**

Program ini dirancang untuk melakukan konversi waktu dari format 24 jam menjadi format 12 jam dengan penambahan penanda waktu “AM” atau “PM”. Program menerima sebuah input berupa bilangan bulat yang merepresentasikan jam dalam format 24 jam (0–23). Selanjutnya, program menggunakan struktur kontrol *switch-case* berbasis kondisi logis untuk menentukan kategori waktu dan mengonversinya ke format 12 jam yang sesuai.

Apabila nilai jam yang dimasukkan adalah 0, program mengonversinya menjadi 12 AM sebagai representasi pukul tengah malam. Jika nilai jam berada dalam rentang 1–11, maka nilai tersebut ditampilkan kembali tanpa perubahan dan diberi label AM. Untuk nilai 12, program menampilkannya sebagai 12 PM yang menunjukkan waktu tengah hari. Sementara itu, untuk nilai jam antara 13–23, program menguranginya dengan 12 untuk memperoleh padanan waktu dalam format 12 jam serta menambahkan label PM. Hasil akhir berupa jam dalam format 12 jam beserta label waktu “AM” atau “PM” dicetak ke layar.

## **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 VS Code interface with a dark theme. In the center is a terminal window displaying Go code and its execution. The code is as follows:

```
alpro_11.go > ~∞ guided_2.go
1 package main
2 import "fmt"
3
4 func main() {
5     var nama_tanaman string
6     fmt.Scan(&nama_tanaman)
7
8     switch nama_tanaman {
9     case "nepenthes", "drosera":
10        fmt.Println("Termasuk Tanaman Karnivora.")
11        fmt.Println("Asli Indonesia.")
12     case "venus", "sarracenia":
13        fmt.Println("Termasuk Tanaman Karnivora.")
14        fmt.Println("Tidak Asli Indonesia.")
15     default:
16        fmt.Println("Tidak termasuk Tanaman Karnivora.")
17    }
18
19 }
```

Below the code, the terminal tab is selected, showing the following command-line interaction:

```
PS C:\Users\HP\.vscode>
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_2.go"
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_2.go"
venus
Termasuk Tanaman Karnivora.
Tidak Asli Indonesia.
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_2.go"
karedok
Tidak termasuk Tanaman Karnivora.
PS C:\Users\HP\.vscode>
```

## Deskripsi program

Program ini dibuat untuk menentukan jenis dan asal suatu tanaman berdasarkan nama tanaman yang dimasukkan oleh pengguna. Program menerima satu buah input berupa string yang merepresentasikan nama tanaman. Selanjutnya, program menggunakan struktur *switch-case* untuk mencocokkan input tersebut dengan beberapa kategori tanaman karnivora yang umum dikenali.

Apabila pengguna memasukkan nama "nepenthes" atau "drosera", program akan menampilkan bahwa tanaman tersebut termasuk kategori tanaman karnivora dan sekaligus merupakan tanaman asli Indonesia. Jika nama yang dimasukkan adalah "venus" atau "sarracenia", program juga mengelompokkan tanaman tersebut sebagai tanaman karnivora, namun bukan asli Indonesia. Untuk semua input selain empat nama

tersebut, program memberikan keluaran bahwa tanaman tersebut tidak termasuk kategori tanaman karnivora.

Melalui pendekatan ini, program mampu mengidentifikasi tanaman tertentu berdasarkan nama serta menampilkan informasi klasifikasi dan asalnya secara terstruktur dan jelas.

### 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

```
alpro_11.go > ~∞ guided_3.go
1 package main
2 import "fmt"
3
4 func main() {
5     var kendaraan string
6     var durasi int
7     var tarif int
8
9     fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
10    fmt.Scan(&kendaraan)
11    fmt.Print("Masukkan durasi parkir (dalam jam): ")
12    fmt.Scan(&durasi)
13
14    switch {
15        case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
16            tarif = 7000
17        case kendaraan == "Motor" && durasi > 2:
18            tarif = 9000
19        case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
20            tarif = 15000
21        case kendaraan == "Mobil" && durasi > 2:
22            tarif = 20000
23        case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
24            tarif = 25000
25        case kendaraan == "Truk" && durasi > 2:
26            tarif = 35000
27        default:
28            fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
29        }
30
31    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
32 }
```

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays five separate runs of the Go program, each prompting for vehicle type and parking duration, and then printing the calculated fare. The vehicle types tested are Motor, Mobil, Truk, and Sepeda.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir: Rp 9000
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\guided_3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Sepeda
Masukkan durasi parkir (dalam jam): 2
Jenis kendaraan atau durasi parkir tidak valid
Tarif Parkir: Rp 0
PS C:\Users\HP\.vscode>
```

### **Deskripsi program**

Program ini dirancang untuk menentukan besarnya tarif parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Program menerima dua input utama, yaitu jenis kendaraan berupa teks (Motor, Mobil, atau Truk) dan durasi parkir dalam satuan jam. Berdasarkan kedua input tersebut, program memproses penentuan tarif menggunakan struktur kontrol switch-case dengan kondisi logis (*boolean expressions*).

Setiap jenis kendaraan memiliki ketentuan tarif yang berbeda, dan tarif tersebut dibedakan lagi berdasarkan lama waktu parkir. Untuk kendaraan Motor, tarif yang dikenakan adalah Rp 7.000 untuk durasi 1–2 jam dan Rp 9.000 jika durasi melebihi 2 jam. Kendaraan Mobil dikenakan tarif Rp 15.000 untuk 1–2 jam dan Rp 20.000 untuk durasi lebih dari 2 jam. Sementara itu, untuk Truk, tarif ditetapkan sebesar Rp 25.000 untuk 1–2 jam, dan Rp 35.000 jika durasi parkirnya lebih dari 2 jam.

Apabila pengguna memasukkan jenis kendaraan atau durasi parkir yang tidak sesuai dengan ketentuan yang tersedia, program akan menampilkan pesan bahwa data yang dimasukkan tidak valid. Setelah seluruh proses berhasil dilakukan, program menampilkan besaran tarif parkir sesuai kategori yang telah ditentukan. Dengan demikian, program ini mampu memberikan perhitungan tarif parkir secara otomatis, konsisten, dan terstruktur.

## **TUGAS**

### **1. Tugas 1**

#### **Source code**

```
package main

import "fmt"

func main() {
    var ph float64
    fmt.Print("Masukkan kadar 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")

default:

    fmt.Println("Air tidak layak minum")

}

}
```

## Screenshoot program

The screenshot shows a VS Code interface with a terminal window open. The terminal displays Go code for a pH classification program and its execution results.

```
alpro_11.go > go soal_1.go
1 package main
2 import "fmt"
3
4 func main() {
5     var ph float64
6     fmt.Println("Masukkan kadar pH: ")
7     fmt.Scan(&ph)
8
9     switch {
10    case ph < 0 || ph > 14:
11        fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
12    case ph >= 6.5 && ph <= 8.6:
13        fmt.Println("Air layak minum")
14    default:
15        fmt.Println("Air tidak layak minum")
16    }
17 }
18
```

TERMINAL

```
PS C:\Users\HP\vscode> go run "c:\Users\HP\vscode\alpro_11.go\soal_1.go"
Masukkan kadar pH: 8.6
Air layak minum
PS C:\Users\HP\vscode> go run "c:\Users\HP\vscode\alpro_11.go\soal_1.go"
Masukkan kadar pH: 9
Air tidak layak minum
PS C:\Users\HP\vscode> go run "c:\Users\HP\vscode\alpro_11.go\soal_1.go"
Masukkan kadar pH: 16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.
PS C:\Users\HP\vscode>
```

## Deskripsi program

Program ini dirancang untuk mengklasifikasikan kelayakan air minum berdasarkan nilai pH yang dimasukkan oleh pengguna. Pengguna memberikan satu masukan berupa bilangan pecahan yang merepresentasikan kadar pH suatu sampel air. Program kemudian melakukan pemeriksaan terhadap nilai tersebut menggunakan struktur kendali *switch-case*. Apabila nilai pH berada pada rentang standar kelayakan air minum, yaitu antara 6.5 hingga 8.6, program menampilkan bahwa air tersebut layak untuk dikonsumsi. Jika nilai pH berada di luar rentang kelayakan namun masih berada dalam batas valid pH (0–14), program menyatakan bahwa air tidak layak untuk diminum. Sementara itu, apabila nilai pH berada di luar rentang valid tersebut, program memberikan keluaran bahwa input tidak valid. Dengan demikian, program ini mampu

memberikan evaluasi sederhana namun sesuai standar baku mutu air berdasarkan nilai pH.

## 2. Tugas 2

### 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)


    if durasi < 1 {

        durasi = 1

    }

    switch {

        case kendaraan == "motor":


            tarif = 2000 * durasi
```

```
case kendaraan == "mobil":  
    tarif = 5000 * durasi  
  
case kendaraan == "truk":  
    tarif = 8000 * durasi  
  
default:  
    fmt.Println("Jenis kendaraan tidak valid")  
    return  
}  
  
fmt.Printf("Total biaya parkir: Rp %d\n", tarif)  
}
```

## Screenshoot program

```
alpro_11.go > ~soal_2.go
 1  package main
 2  import "fmt"
 3
 4  func main() {
 5      var kendaraan string
 6      var durasi int
 7      var tarif int
 8
 9      fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk): ")
10      fmt.Scan(&kendaraan)
11      fmt.Print("Masukkan durasi parkir (dalam jam): ")
12      fmt.Scan(&durasi)
13
14      if durasi < 1 {
15          durasi = 1
16      }
17
18      switch {
19      case kendaraan == "motor":
20          tarif = 2000 * durasi
21
22      case kendaraan == "mobil":
23          tarif = 5000 * durasi
24
25      case kendaraan == "truk":
26          tarif = 8000 * durasi
27
28      default:
29          fmt.Println("Jenis kendaraan tidak valid")
30          return
31      }
32
33      fmt.Printf("Total biaya parkir: Rp %d\n", tarif)
34  }
```

```
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\soal_2.go"
Masukkan jenis kendaraan (motor/mobil/truk): motor
Masukkan durasi parkir (dalam jam): 3
Total biaya parkir: Rp 6000
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\soal_2.go"
Masukkan jenis kendaraan (motor/mobil/truk): mobil
Masukkan durasi parkir (dalam jam): 1
Total biaya parkir: Rp 5000
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\soal_2.go"
Masukkan jenis kendaraan (motor/mobil/truk): truk
Masukkan durasi parkir (dalam jam): 5
Total biaya parkir: Rp 40000
PS C:\Users\HP\.vscode>
```

### **Deskripsi program**

Program ini dibuat untuk menghitung total biaya parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Jenis kendaraan yang dapat dipilih meliputi motor, mobil, dan truk, masing-masing dengan tarif per jam yang telah ditentukan, yaitu Rp 2.000 untuk motor, Rp 5.000 untuk mobil, dan Rp 8.000 untuk truk. Pengguna juga memasukkan durasi parkir dalam satuan jam, dan program memastikan bahwa durasi yang bernilai kurang dari satu jam tetap dihitung sebagai satu jam penuh. Penggunaan struktur *switch-case* memungkinkan program menentukan tarif per jam berdasarkan kategori kendaraan secara lebih terstruktur. Setelah tarif ditentukan, tarif tersebut dikalikan dengan durasi parkir untuk memperoleh total biaya parkir. Program ini membantu memberikan perhitungan tarif parkir secara otomatis dan konsisten sesuai dengan ketentuan tarif yang berlaku.

### **3. Tugas 3**

#### **Source code**

```
package main
import "fmt"

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

    next := n + 1
    hasil := 0

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

    case n%5 == 0 && n >= 10:
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        hasil = n * n
        fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n",
n, hasil)

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

    default:
        fmt.Println("Kategori: Bilangan Ganjil")
```

```

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

```

## Screenshoot program

```

alpro_11.go > soal_3.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("Masukkan bilangan: ")
8     fmt.Scan(&n)
9
10    next := n + 1
11    hasil := 0
12
13    switch {
14        case n%10 == 0:
15            fmt.Println("Kategori: Bilangan Kelipatan 10")
16            hasil = n / 10
17            fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, hasil)
18
19        case n%5 == 0 && n >= 10:
20            fmt.Println("Kategori: Bilangan Kelipatan 5")
21            hasil = n * n
22            fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", n, hasil)
23
24        case n%2 == 0:
25            fmt.Println("Kategori: Bilangan Genap")
26            hasil = n * next
27            fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", n, next, hasil)
28
29        default:
30            fmt.Println("Kategori: Bilangan Ganjil")
31            hasil = n + next
32            fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, next, hasil)
33    }
34 }

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\soal_3.go"
Masukkan bilangan: 5
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\soal_3.go"
Masukkan bilangan: 8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\soal_3.go"
Masukkan bilangan: 25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^2 = 625
PS C:\Users\HP\.vscode> go run "c:\Users\HP\.vscode\alpro_11.go\soal_3.go"
Masukkan bilangan: 20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS C:\Users\HP\.vscode>

```

### **Deskripsi program**

Program ini berfungsi untuk mengidentifikasi kategori suatu bilangan serta melakukan operasi aritmatika sesuai dengan kategori bilangan tersebut. Program menerima satu masukan berupa bilangan bulat, kemudian menentukan kategorinya melalui struktur *switch-case*. Kategori bilangan yang digunakan meliputi bilangan ganjil, bilangan genap, bilangan kelipatan lima, dan bilangan kelipatan sepuluh. Setelah kategori ditentukan, program melakukan operasi aritmatika yang berkaitan dengan kategori tersebut: bilangan ganjil dijumlahkan dengan bilangan berikutnya, bilangan genap dikalikan dengan bilangan berikutnya, bilangan kelipatan lima dikuadratkan, dan bilangan kelipatan sepuluh dibagi dengan angka sepuluh. Hasil perhitungan ditampilkan secara terstruktur sesuai dengan format keluaran yang telah ditetapkan. Program ini menunjukkan penerapan logika seleksi kondisi dalam pengolahan bilangan untuk menghasilkan keluaran yang spesifik sesuai kategori bilangan.