

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 11  
SWITCH CASE**



**Disusun oleh:**

**NAUFAL BINTANG P**

**109082500096**

**S1IF-13-07**

**Asisten Praktikum**

Adithana dharma putra

Apri pandu wicaksono

**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)
}
```

#### Screenshoot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)  
**CONTOH TANGKAPAN LAYAR:**

The screenshot shows a Go IDE with a file named `guide1mod11.go`. The code defines a `main` function that takes two integers, `jam12` and `jam24`, and a string `label`. It uses a `switch` statement to convert `jam24` to `jam12` and determine the `label` ("AM" or "PM"). The terminal output shows the program being run multiple times, displaying the converted time and label.

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

Terminal Output:

```
PS D:\go\modul11> go run .\guide1mod11\guide1mod11.go
13
1 PM
PS D:\go\modul11> go run .\guide1mod11\guide1mod11.go
0
12 AM
PS D:\go\modul11> go run .\guide1mod11\guide1mod11.go
12
12 PM
PS D:\go\modul11> 
```

## Deskripsi program

Program ini berfungsi untuk **mengubah input waktu dalam format 24 jam (0–23) menjadi format 12 jam yang lebih umum digunakan, serta menambahkan keterangan AM atau PM berdasarkan periode waktu tertentu**. Dengan memanfaatkan struktur percabangan `switch-case`, program menentukan apakah waktu tersebut berada pada rentang tengah malam, pagi, siang, atau malam. Setelah dilakukan proses konversi, hasil berupa angka jam dan label AM/PM ditampilkan sebagai output. Program ini membantu pengguna agar lebih mudah memahami waktu dalam format yang sering dipakai pada kehidupan sehari-hari, seperti pada jam dinding, jadwal kegiatan, dan tampilan waktu digital.

## 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.")
}
}

```

### Screenshoot program

The screenshot shows a Go program in VS Code. The code is as follows:

```

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

```

The terminal output shows the program being run with the following inputs and outputs:

```

PS D:\go\modul11> go run .\guide2mod11\guide2mod11.go
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS D:\go\modul11> go run .\guide2mod11\guide2mod11.go
venus
Termasuk Tanaman Karnivora.
Tidak Asli Indonesia.
PS D:\go\modul11> go run .\guide2mod11\guide2mod11.go
karedok
Tidak termasuk Tanaman Karnivora.
PS D:\go\modul11>

```

### Deskripsi program

Program ini berfungsi untuk mengidentifikasi jenis tanaman berdasarkan nama yang dimasukkan oleh pengguna, serta menentukan apakah tanaman tersebut termasuk dalam kategori tanaman karnivora dan apakah berasal dari Indonesia atau bukan. Program menerima input berupa nama tanaman dalam bentuk string, kemudian menggunakan struktur percabangan switch-case untuk melakukan pengecekan. Jika pengguna memasukkan nama tertentu seperti *nepenthes* atau *drosera*, program akan menampilkan bahwa tanaman tersebut merupakan tanaman karnivora asli Indonesia. Untuk nama seperti *venus* atau *sarracenia*, program menampilkan bahwa tanaman tersebut merupakan tanaman karnivora tetapi bukan asli Indonesia. Apabila

nama tanaman tidak cocok dengan pilihan yang tersedia, program akan memberikan output bahwa tanaman tersebut tidak termasuk dalam tanaman karnivora. Dengan demikian, program ini membantu pengguna mengenali jenis tanaman berdasarkan klasifikasinya.

### 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\n(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

The screenshot shows a Go IDE with the following code in `guide3mod11.go`:

```

3 func main() {
4     var tarif int
5     fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
6     fmt.Scan(&kendaraan)
7     fmt.Print("Masukkan durasi parkir (dalam jam): ")
8     fmt.Scan(&durasi)
9     switch {
10    case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
11        tarif = 7000
12    case kendaraan == "Motor" && durasi > 2:
13        tarif = 9000
14    case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
15        tarif = 15000
16    case kendaraan == "Mobil" && durasi > 2:
17        tarif = 20000
18    case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
19        tarif = 25000
20    case kendaraan == "Truk" && durasi > 2:
21        tarif = 35000
22    default:
23        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
24    }
25 }

```

The terminal output shows two successful runs and one invalid input case:

```

Tarif Parkir: Rp 7000
PS D:\go\modul11> go run .\guide3mod11\guide3mod11.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000

PS D:\go\modul11> go run .\guide3mod11\guide3mod11.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS D:\go\modul11>

```

## Deskripsi program

Program ini berfungsi untuk **menghitung dan menampilkan tarif parkir berdasarkan jenis kendaraan dan durasi parkir dalam satuan jam**. Program meminta pengguna memasukkan jenis kendaraan, seperti Motor, Mobil, atau Truk, serta lama parkir dalam jam. Setelah menerima input, program menggunakan struktur percabangan switch yang memeriksa kombinasi jenis kendaraan dan durasi untuk menentukan besarnya tarif yang harus dibayar. Tarif dibedakan berdasarkan kategori waktu, yaitu durasi parkir 1–2 jam dan lebih dari 2 jam. Apabila data yang dimasukkan tidak sesuai dengan pilihan yang tersedia, program akan menampilkan pesan bahwa input tidak valid. Hasil akhir berupa tarif parkir kemudian ditampilkan kepada pengguna dalam format rupiah.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main
```

```

import "fmt"

func main() {
    var ph float64

    fmt.Print("Masukkan kadar pH air: ")
    fmt.Scan(&ph)

    if ph < 0 || ph > 14 {
        fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0
dan 14.")
    } else if ph >= 6.5 && ph <= 8.6 {
        fmt.Println("Air layak minum")
    } else {
        fmt.Println("Air tidak layak minum")
    }
}

```

## Screenshoot program

The screenshot shows the Go program being executed in a Windows environment. The code in the editor is as follows:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float64
7
8     fmt.Print("Masukkan kadar pH air: ")
9     fmt.Scan(&ph)
10
11     if ph < 0 || ph > 14 {
12         fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
13     } else if ph >= 6.5 && ph <= 8.6 {
14         fmt.Println("Air layak minum")
15     } else {
16         fmt.Println("Air tidak layak minum")
17     }
18 }
19

```

The terminal output shows the following sequence of commands and results:

```

PS D:\go\modul11> go run .\latmod11\latmod11.go
Masukkan kadar pH air: 8.6
Air layak minum
PS D:\go\modul11> go run .\latmod11\latmod11.go
Masukkan kadar pH air: 9
Air tidak layak minum
PS D:\go\modul11> go run .\latmod11\latmod11.go
Masukkan kadar pH air: 16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.
PS D:\go\modul11>

```

A small Notepad window is also visible in the background, containing the text:

```

NAUFAL BINTANG PRATAMA
S1IF-13-07
109082500096

```

## Deskripsi program

Program ini berfungsi untuk **menentukan kelayakan air untuk dikonsumsi berdasarkan nilai pH yang diinput oleh pengguna**. Program akan meminta pengguna memasukkan angka pH air dalam bentuk bilangan desimal. Kemudian, program melakukan pengecekan menggunakan struktur percabangan **if-else** untuk memastikan apakah nilai pH tersebut berada dalam rentang valid yaitu 0 sampai 14. Jika nilai pH di luar rentang tersebut, program menampilkan pesan bahwa nilai pH tidak valid. Namun, apabila nilai pH berada pada kisaran 6,5 hingga 8,6, program akan menampilkan bahwa air tersebut layak diminum karena berada dalam standar kualitas air minum. Sedangkan air yang memiliki pH di luar rentang tersebut dinyatakan tidak layak minum karena dapat membahayakan kesehatan. Program ini membantu pengguna memahami kelayakan air berdasarkan parameter kimia pH secara sederhana.

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "strings"
)

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

    fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk): ")
    fmt.Scan(&kendaraan)
    kendaraan = strings.ToLower(kendaraan)

    fmt.Print("Masukkan durasi parkir (jam): ")
    fmt.Scan(&durasi)

    if durasi < 1 {
```



```
        durasi = 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 * durasi

    fmt.Println("Total biaya parkir: Rp", total)
}
```

**Screenshoot program**

```
1 import (
2     "fmt"
3     "strings"
4 )
5
6 func main() {
7     var kendaraan string
8     var durasi, tarif, total int
9
10    fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk): ")
11    fmt.Scan(&kendaraan)
12    kendaraan = strings.ToLower(kendaraan)
13
14    fmt.Print("Masukkan durasi parkir (jam): ")
15    fmt.Scan(&durasi)
16
17    if durasi < 1 {
18        durasi = 1
19    }
20
21    switch kendaraan {
22        case "motor":
23            tarif = 2000
24        case "mobil":
25            tarif = 5000
26        case "truk":
27            tarif = 8000
28        default:
29            fmt.Println("Jenis kendaraan tidak valid")
30            return
31    }
32
33    total = tarif * durasi
34    fmt.Println("Total biaya parkir: Rp", total)
35}
```

Terminal Output:

```
Total biaya parkir: Rp 6000
PS D:\go\modul11> go run .\lat2mod11\lat2mod11.go
Masukkan jenis kendaraan (motor/mobil/truk): mobil
Masukkan durasi parkir (jam): 1
Total biaya parkir: Rp 5000
PS D:\go\modul11> go run .\lat2mod11\lat2mod11.go
Masukkan jenis kendaraan (motor/mobil/truk): truk
Masukkan durasi parkir (jam): 5
Total biaya parkir: Rp 40000
PS D:\go\modul11> 
```

### Deskripsi program

Program ini berfungsi untuk **menghitung total biaya parkir berdasarkan jenis kendaraan dan durasi parkir dalam jam**. Pengguna diminta memasukkan jenis kendaraan seperti motor, mobil, atau truk, kemudian program akan mengubah input menjadi huruf kecil agar proses pengecekan lebih akurat. Setelah itu pengguna memasukkan durasi parkir, dan jika durasi kurang dari satu jam maka program otomatis menetapkan durasi minimum satu jam. Program menggunakan struktur switch untuk menentukan tarif per jam sesuai jenis kendaraan, yaitu Rp 2.000 untuk motor, Rp 5.000 untuk mobil, dan Rp 8.000 untuk truk. Jika jenis kendaraan tidak sesuai dengan pilihan yang tersedia, program menampilkan pesan kesalahan dan menghentikan proses. Setelah tarif ditentukan, total biaya dihitung dengan mengalikan tarif per jam dengan durasi parkir dan hasilnya ditampilkan kepada pengguna. Program ini membantu memberikan perhitungan biaya parkir secara cepat, praktis, dan akurat berdasarkan kategori kendaraan.

### 3. Tugas 3

#### 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", bilangan,
bilangan/10)
    case bilangan%2 == 0:
        fmt.Println("Kategori: Bilangan Genap")
        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d",
bilangan, bilangan+1, bilangan*(bilangan+1))
    case bilangan%2 != 0 && (bilangan%5 != 0 || bilangan < 10):
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d =
%d", bilangan, bilangan+1, bilangan+(bilangan+1))
    case bilangan%5 == 0:
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d", bilangan,
bilangan*bilangan)
    }
}

```

### Screenshoot program

The screenshot shows a Go program in VS Code. The code is as follows:

```

1 func main() {
2     fmt.Scan(&bilangan)
3     switch {
4     case bilangan%10 == 0:
5         fmt.Println("Kategori: Bilangan Kelipatan 10")
6         fmt.Printf("Hasil pembagian antara %d / 10 = %d", bilangan, bilangan/10)
7     case bilangan%2 == 0:
8         fmt.Println("Kategori: Bilangan Genap")
9         fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d", bilangan, bilangan+1, bilangan*(bilangan+1))
10    case bilangan%2 != 0 && (bilangan%5 != 0 || bilangan < 10):
11        fmt.Println("Kategori: Bilangan Ganjil")
12        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d", bilangan, bilangan+1, bilangan+(bilangan+1))
13    case bilangan%5 == 0:
14        fmt.Println("Kategori: Bilangan Kelipatan 5")
15        fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d", bilangan, bilangan*bilangan)
16    }
17 }

```

The terminal output shows the program running with the input 25:

```

PS D:\go\modul11> go run .\lat3mod11\lat3mod11.go
25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^ 2 = 625
PS D:\go\modul11> go run .\lat3mod11\lat3mod11.go
20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS D:\go\modul11>

```

A small window in the foreground shows a student ID card for NAUFAL BINTANG PRATAMA, S1IF-13-07, 109082500096.

### Deskripsi program

Program ini berfungsi untuk **mengklasifikasikan sebuah bilangan berdasarkan karakteristiknya, serta melakukan operasi matematika tertentu sesuai kategori yang dimasukkan oleh pengguna**. Setelah pengguna memasukkan sebuah bilangan, program menggunakan struktur percabangan switch untuk mengevaluasi kondisi bilangan tersebut. Jika bilangan merupakan kelipatan 10, program akan menampilkan hasil pembagian bilangan tersebut dengan angka 10. Jika bilangan genap namun bukan kelipatan 10, program menampilkan hasil perkalian antara bilangan dengan

bilangan berikutnya. Untuk bilangan ganjil yang bukan kelipatan 5 atau bilangan ganjil kurang dari 10, program menampilkan hasil penjumlahan dengan bilangan berikutnya. Sementara itu, apabila bilangan termasuk kelipatan 5, program menampilkan hasil kuadrat dari bilangan tersebut. Dengan proses ini, program membantu mengidentifikasi kategori bilangan dan memberikan hasil perhitungan yang sesuai secara otomatis.