

# **LAPORAN PRAKTIKUM**

## **Algoritma Pemrograman**

MODUL 11

SWITCH-CASE



**Disusun Oleh:**

MUHAMAD RAFI ALFIANSYAH

109082500191

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

# LATIHAN KELAS – GUIDED

## 1. Guided 1

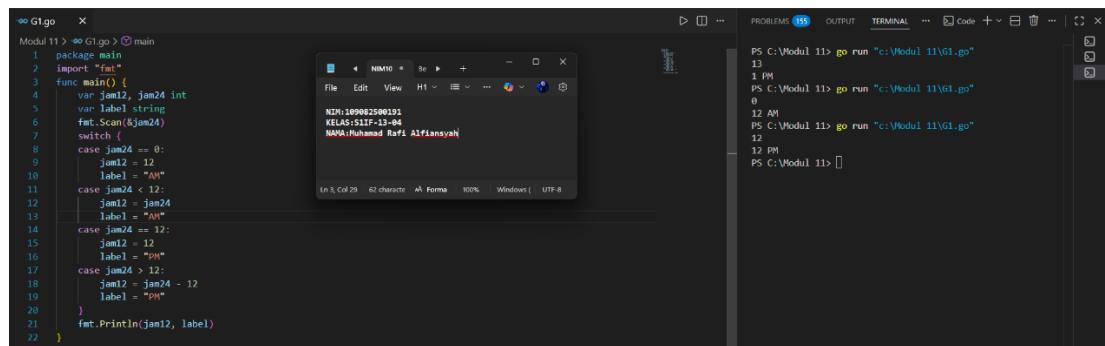
### Source Code

```
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



The screenshot shows a code editor with a Go file named G1.go open. The code implements a function to convert a 24-hour time format into a 12-hour format with AM/PM labels. A terminal window is visible in the background, showing the execution of the program and its output.

```
PS C:\Modul 11> go run "c:\Modul 11\G1.go"
13
1 PM
PS C:\Modul 11> go run "c:\Modul 11\G1.go"
0
12 AM
PS C:\Modul 11> go run "c:\Modul 11\G1.go"
12
12 PM
PS C:\Modul 11>
```

## **Deskripsi Program**

Program ini dibuat untuk mengonversi waktu dari format 24 jam menjadi format 12 jam. Program membaca input berupa satu angka jam, kemudian menentukan jam yang sesuai pada format 12 jam dan menentukan apakah waktunya AM atau PM.

- **Penjelasan Program :**

1. Program mendeklarasikan dua variabel yaitu jam12 dan jam24 untuk menyimpan nilai jam, serta variabel label untuk menyimpan output AM atau PM.
2. Program membaca input dari pengguna melalui fungsi `fmt.Scan(&jam24)` dalam bentuk angka jam 24 jam.
3. Program membaca input dari pengguna melalui fungsi `fmt.Scan(&jam24)` dalam bentuk angka jam 24 jam.
4. Program melakukan pengecekan kondisi jam:
  - > Jika  $\text{jam24} == 0$ , maka jam pada format 12 jam menjadi 12 dan diberi label AM.
  - > Jika  $\text{jam24} < 12$ , nilai jam tidak berubah dan diberi label AM.
  - > Jika  $\text{jam24} == 12$ , maka jam tetap 12 namun diberi label PM.
  - > Jika  $\text{jam24} > 12$ , maka jam dihitung ulang dengan cara  $\text{jam24} - 12$  dan diberi label PM.

## 2. Guided 2

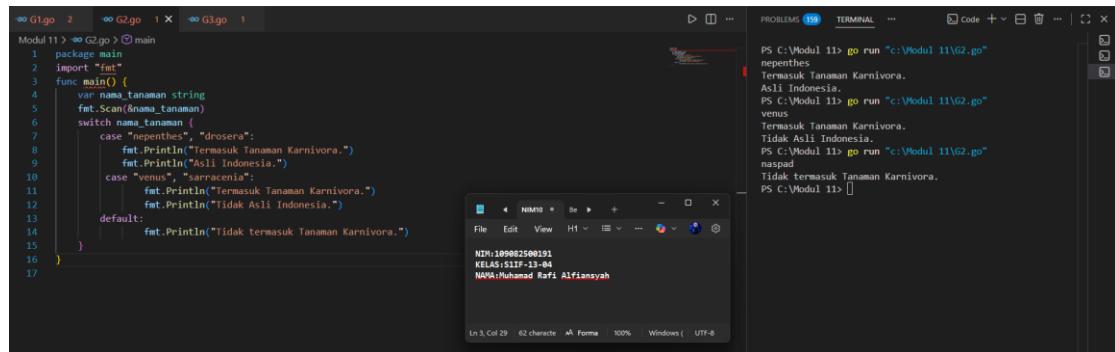
### Source Code

```
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.")
        default:
            fmt.Println("Tidak termasuk Tanaman
Karnivora.")
    }
}
```

### Screenshot Program



## **Deskripsi Program**

Program ini dibuat untuk mengecek apakah sebuah tanaman termasuk tanaman karnivora atau bukan, berdasarkan nama yang dimasukkan oleh pengguna. Pertama, program meminta pengguna untuk mengetikkan nama tanaman, lalu nama tersebut disimpan dalam variabel `nama_tanaman`.

Setelah itu program menggunakan struktur switch untuk membandingkan input pengguna dengan beberapa nama tanaman yang sudah ditentukan. Jika pengguna memasukkan "nepenthes" atau "drosera", program akan menampilkan bahwa tanaman tersebut adalah tanaman karnivora dan juga termasuk tanaman asli Indonesia. Kalau yang dimasukkan adalah "venus" atau "sarracenia", program tetap menganggapnya sebagai tanaman karnivora, tetapi bukan tanaman asli Indonesia.

Di luar nama-nama itu, program akan memberi output bahwa tanaman tersebut tidak termasuk tanaman karnivora. Secara sederhana, program ini hanya melakukan pengecekan nama dan memberikan informasi sesuai kategori tanaman tersebut.

### 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

```
Modul 11 > G1.go 2 > G2.go 1 > G3.go 1 x
Modul 11 > G3.go > main
1 package main
2 import "fmt"
3 func main() {
4     var kendaraan string
5     var durasi int
6     var tarif int
7     fmt.Println("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
8     fmt.Scan(&kendaraan)
9     fmt.Println("Masukkan durasi parkir (dalam jam): ")
10    fmt.Scan(&durasi)
11    switch {
12        case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
13            tarif = 7000
14        case kendaraan == "Motor" && durasi > 2:
15            tarif = 9000
16        case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
17            tarif = 15000
18        case kendaraan == "Mobil" && durasi > 2:
19            tarif = 20000
20        case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
21            tarif = 25000
22        case kendaraan == "Truk" && durasi > 2:
23            tarif = 35000
24        default:
25            fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
26        }
27        fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
28    }

PS C:\Modul 11> go run "c:\Modul 11\G3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS C:\Modul 11> go run "c:\Modul 11\G3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS C:\Modul 11> go run "c:\Modul 11\G3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS C:\Modul 11> go run "c:\Modul 11\G3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Sepeda
Masukkan durasi parkir (dalam jam): 90
Jenis kendaraan atau durasi parkir tidak valid
Tarif Parkir: Rp 0
PS C:\Modul 11>
```

## Deskripsi Program

Program ini digunakan untuk menghitung tarif parkir berdasarkan jenis kendaraan dan lama waktu parkir. Pertama, program meminta pengguna memasukkan dua data, yaitu jenis kendaraan (Motor, Mobil, atau Truk) dan durasi parkir dalam satuan jam. Kedua input ini disimpan masing-masing ke variabel kendaraan untuk jenis kendaraan, serta durasi untuk lama parkir.

Setelah data berhasil diinput, program menentukan tarif dengan menggunakan struktur switch tanpa ekspresi, yang diisi beberapa kondisi case untuk mencocokkan jenis kendaraan sekaligus durasi parkirnya. Setiap kondisi sudah diatur agar tarif berbeda-beda tergantung jenis kendaraan dan lamanya parkir.

Jika kendaraan adalah Motor dan durasinya 1–2 jam, tarif yang dikenakan adalah Rp 7.000. Bila lebih dari 2 jam, tarif motor menjadi Rp 9.000. Untuk kendaraan Mobil, tarifnya Rp 15.000 untuk 1–2 jam, dan Rp 20.000 jika lebih dari 2 jam. Sedangkan Truk memiliki tarif paling tinggi: Rp 25.000 untuk 1–2 jam dan Rp 35.000 bila lebih dari 2 jam.

Jika pengguna memasukkan jenis kendaraan di luar pilihan tersebut atau memberikan angka durasi yang tidak valid, program akan menampilkan pesan bahwa input tidak sesuai. Setelah kondisi yang cocok ditemukan, program mencetak hasil tarif yang harus dibayar melalui `fmt.Printf`.

# TUGAS

## 1. Tugas 1

### Source Code

```
package main

import "fmt"

func main() {

    var ph float64

    fmt.Scan(&ph)

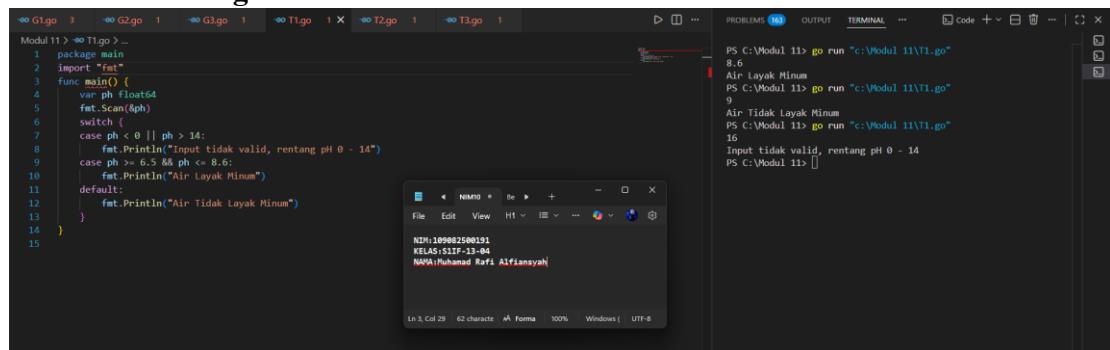
    switch {

    case ph < 0 || ph > 14:
        fmt.Println("Input tidak valid, rentang pH 0 - 14")

    case ph >= 6.5 && ph <= 8.6:
        fmt.Println("Air Layak Minum")

    default:
        fmt.Println("Air Tidak Layak Minum")
    }
}
```

### Screenshot Program



### Deskripsi Program

Program ini digunakan untuk menentukan apakah sebuah sampel air termasuk layak minum atau tidak berdasarkan nilai pH yang dimasukkan pengguna. Pada awalnya, pengguna diminta memberikan sebuah nilai berupa angka desimal. Angka ini kemudian disimpan dalam variabel ph yang mewakili tingkat keasaman atau kebasaan air.

Setelah nilai pH diterima, program mulai melakukan pengecekan melalui beberapa kondisi berurutan. Pengecekan pertama memastikan terlebih dahulu apakah nilai pH masih berada dalam rentang normal, yaitu 0 hingga 14. Jika angka yang dimasukkan berada di luar rentang tersebut, program langsung memberikan informasi bahwa input tidak valid, karena nilai pH tidak mungkin kurang dari 0 atau lebih dari 14 dalam kondisi normal.

Jika nilai pH masih berada dalam batas wajar, program melanjutkan pengecekan untuk menentukan apakah air tersebut termasuk kategori layak diminum. Air akan dianggap layak jika nilai pH berada pada kisaran 6.5 sampai 8.6. Rentang ini merupakan kisaran umum air netral hingga sedikit basa yang aman untuk dikonsumsi manusia. Jika nilai pH jatuh tepat di dalam rentang tersebut, program langsung menampilkan pesan bahwa air dikategorikan sebagai air layak minum.

Untuk nilai pH yang masih valid tetapi berada di luar rentang aman, program menganggap bahwa air tersebut tidak layak diminum. Hal ini bisa terjadi jika nilai pH terlalu asam atau terlalu basa, yang dapat membahayakan tubuh apabila dikonsumsi.

Secara keseluruhan, program ini berfungsi sebagai alat sederhana untuk mengklasifikasikan kualitas air berdasarkan pH, dengan langkah-langkah berupa penerimaan input, validasi rentang nilai, dan penentuan kategori kelayakan air.

## 2. Tugas 2

### Source Code

```
package main

import "fmt"

func main() {
    var j,jam string
    var w,t int
    fmt.Scan(&j,&w,&jam)
    switch j{
        case "motor":
            t = w * 2000
        case "mobil":
            t = w * 5000
        case "truk":
            t = w * 8000
    }
    fmt.Println("Rp",t)
}
```

### Screenshot Program

```
C:\Modul 11\T2.go: 5 problems in this file
1 package main
2 import "fmt"
3 func main(){
4     var j,jam string
5     var w,t int
6     fmt.Scan(&j,&w,&jam)
7     switch j{
8         case "motor":
9             t = w * 2000
10        case "mobil":
11            t = w * 5000
12        case "truk":
13            t = w * 8000
14    }
15    fmt.Println("Rp",t)
16 }

PS C:\Modul 11> go run "c:\Modul 11\T2.go"
motor 3 jam
Rp 6000
PS C:\Modul 11> go run "c:\Modul 11\T2.go"
mobil 1 jam
Rp 5000
PS C:\Modul 11> go run "c:\Modul 11\T2.go"
truk 5 jam
Rp 40000
PS C:\Modul 11>
```

## **Deskripsi Program**

Program ini digunakan untuk menghitung biaya parkir berdasarkan jenis kendaraan dan lama waktu parkir. Di awal, pengguna diminta memasukkan tiga data sekaligus, yaitu jenis kendaraan ( $j$ ), durasi parkir dalam satuan jam ( $w$ ), serta sebuah input tambahan berupa string jam masuk atau informasi lain yang tidak berpengaruh pada proses perhitungan. Seluruh input tersebut dibaca secara berurutan melalui satu baris.

Setelah semua data diterima, program mulai menentukan tarif menggunakan struktur percabangan. Pemilihan tarif dilakukan berdasarkan jenis kendaraan yang dimasukkan pengguna. Untuk kendaraan motor, biaya parkir dihitung dengan mengalikan lama parkir dengan tarif Rp 2.000 per jam. Jika kendaraan yang dipilih adalah mobil, tarifnya lebih tinggi, yaitu Rp 5.000 per jam. Sedangkan untuk jenis truk, biaya per jamnya adalah yang paling besar, yaitu Rp 8.000 per jam.

Setelah jenis kendaraan dikenali dan tarif per jam ditetapkan, program menghitung total biaya parkir dengan cara mengalikan durasi ( $w$ ) dengan tarif sesuai kategori kendaraan. Hasil perhitungan tersebut kemudian ditampilkan kepada pengguna dalam bentuk “Rp <nominal>”.

Secara keseluruhan, program ini berfungsi sebagai sistem sederhana yang menentukan total biaya parkir berdasarkan kategori kendaraan dan lama pengguna memparkirkan kendaraannya. Program ini menunjukkan proses pengambilan keputusan menggunakan percabangan serta perhitungan memakai operasi aritmatika dasar.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {
    var b int
    fmt.Scan(&b)
    switch{
        case b % 10 ==0:
            fmt.Println("Kategori : Bilangan Kelipatan 10")
            fmt.Printf("Hasil pembagian antara %d / 10 = %d",b, b/10)
        case b % 5 == 0 && b != 5:
            fmt.Println("Kategori : Bilangan Kelipatan 5")
            fmt.Printf("Hasil kuadrat dari %d ^2 = %d",b, b*b)
        case b % 2 == 1:
            fmt.Println("Kategori : Bilangan Ganjil")
            fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d",b, b+1, b+(b+1))
        case b % 2 == 0:
            fmt.Println("Kategori : Bilangan Genap")
            fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d",b, b+1, b*(b+1))
    }
}
```

#### Screenshot program

```
PS C:\Modul 11> go run "c:\Modul 11\T3.go"
5
Kategori : Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS C:\Modul 11> go run "c:\Modul 11\T3.go"
8
Kategori : Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Modul 11> go run "c:\Modul 11\T3.go"
25
Kategori : Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^2 = 625
PS C:\Modul 11> go run "c:\Modul 11\T3.go"
10
Kategori : Bilangan Kelipatan 10
Hasil pembagian antara 10 / 10 = 1
PS C:\Modul 11>
```

## **Deskripsi Program**

Program ini dibuat untuk mengklasifikasikan sebuah bilangan bulat yang dimasukkan oleh pengguna, lalu memberikan operasi perhitungan tertentu sesuai kategori bilangan tersebut. Pada awal eksekusi, pengguna diminta memasukkan satu nilai bilangan bulat, yang kemudian disimpan dalam variabel b.

Setelah bilangan diterima, program melakukan serangkaian pengecekan kondisi secara berurutan menggunakan struktur percabangan. Pengecekan pertama adalah apakah bilangan tersebut merupakan kelipatan 10, yang ditentukan dengan memeriksa apakah sisa hasil bagi  $b \% 10$  bernilai 0. Jika benar, bilangan langsung dikategorikan sebagai Bilangan Kelipatan 10, dan program menampilkan hasil pembagian bilangan tersebut dengan 10.

Jika bilangan bukan kelipatan 10, program melanjutkan ke kondisi berikutnya, yaitu memeriksa apakah bilangan tersebut merupakan kelipatan 5 (dengan  $b \% 5 == 0$ ), namun bukan angka 5 itu sendiri. Jika masuk kategori ini, program menampilkannya sebagai Bilangan Kelipatan 5 dan menghitung hasil kuadrat dari bilangan tersebut.

Apabila bilangan tidak masuk kedua kategori di atas, program memeriksa apakah bilangan tersebut ganjil dengan melihat apakah  $b \% 2 == 1$ . Jika bilangan adalah bilangan ganjil, program menampilkan kategorinya dan menghitung hasil penjumlahan antara bilangan tersebut dengan bilangan berikutnya ( $b + (b + 1)$ ).

Jika semua kondisi sebelumnya tidak terpenuhi, maka bilangan yang dimasukkan sudah pasti merupakan bilangan genap. Dalam kategori ini, program menampilkan bahwa bilangan tersebut adalah bilangan genap, lalu menghitung hasil perkalian bilangan tersebut dengan bilangan sesudahnya.

Secara keseluruhan, program ini berfungsi sebagai alat untuk mengidentifikasi jenis suatu bilangan (kelipatan 10, kelipatan 5, ganjil, atau genap), sekaligus memberikan output tambahan berupa operasi matematika yang berbeda untuk setiap kategori. Program ini menekankan pemahaman tentang percabangan, sisa pembagian, serta pengolahan bilangan menggunakan operasi aritmatika dasar.