

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 11**  
**“SWITCH-CASE”**



**DISUSUN OLEH:**  
**Keishin Naufa Alfaridzhi**  
**103112400061**  
**S1 IF-12-01**  
**DOSEN:**  
**Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024/2025**

## DASAR TEORI

### A. Bahasa Yang Digunakan

Pada praktikum ini bahasa pemrograman yang digunakan adalah bahasa pemrograman Go, sesuai dengan modul yang menjadi acuan praktikum. Golang (atau Go) adalah bahasa pemrograman baru, yang mulai dilirik oleh para developer karena kelebihan-kelebihan yang dimilikinya. Sudah banyak Perusahaan besar yang menggunakan bahasa ini untuk produk-produk mereka hingga di level production.

### B. Komentar

Komentar biasa dimanfaatkan untuk menyisipkan catatan pada kode program, menulis penjelasan atau deskripsi mengenai suatu blok kode, atau bisa juga digunakan untuk me-remark kode (men-non-aktifkan kode yang tidak digunakan). Komentar akan diabaikan Ketika kompilasi maupun eksekusi program.

Ada 2 jenis komentar di Golang, yaitu inline dan multiline.

#### 1. Komentar Inline

Penulisan komentar jenis ini diawali dengan tanda *double slash* (*//*) lalu diikuti pesan komentarnya. Komentar inline hanya berlaku untuk satu baris pesan saja. Jika pesan komentar lebih dari satu baris, maka tanda *double slash* harus ditulis lagi di baris selanjutnya.

#### 2. Komentar Multiline

Komentar yang cukup panjang akan lebih rapi jika ditulis menggunakan teknik komentar multiline. Ciri dari komentar jenis ini adalah penulisannya diawali dengan tanda *(/\**) dan diakhiri *(\*/)*.

### C. Variabel

Golang mengadopsi 2 jenis penulisan variabel, yang dituliskan tipe data-nya dan yang tidak. Kedua cara tersebut intinya adalah sama, pembedanya hanyalah cara penulisannya saja. Untuk penulisan variabel dengan tipe data, keyword *var* digunakan untuk deklarasi variabel kemudian diakhiri dengan tipe data misalnya *string*. Kemudian untuk penulisan variabel tanpa tipe data, variabel dideklarasikan dengan menggunakan metode type inference. Penandanya tipe data tidak dituliskan pada saat deklarasi. Pada penggunaan metode ini, operand (*=*) harus diganti dengan (*:=*) dan keyword *var* dihilangkan.

Golang memiliki aturan unik yang tidak dimiliki bahasa lain, yaitu tidak boleh ada satupun variabel yang menganggur. Artinya, semua variabel yang dideklarasikan harus digunakan. Jika terdapat variabel yang tidak digunakan tapi dideklarasikan, program akan gagal dikompilasi. Untuk mengatasi itu, golang memiliki variabel yaitu underscore. Underscore ( \_ ) adalah predefined variabel yang bisa dimanfaatkan untuk menampung nilai yang tidak dipakai.

#### **D. Tipe Data**

Golang mengenal beberapa jenis tipe data, diantaranya adalah tipe data numerik (decimal dan non-desimal), string, dan boolean.

1. Tipe Data Numerik Non-Desimal (uint, int)
2. Tipe Data Numerik Desimal (float64, float32)
3. Tipe Data Bool (true, false)
4. Tipe Data String (string, “ “)

#### **E. Operator Aritmatika**

Operator aritmatika merupakan operator yang digunakan untuk operasi yang sifatnya perhitungan. Golang mendukung beberapa operator aritmatika standar, yaitu:

1. Penjumlahan (+)
2. Pengurangan (-)
3. Perkalian (\*)
4. Pembagian (/)
5. Modulus atau sisa hasil pembagian (%)

#### **F. Seleksi Kondisi**

Seleksi kondisi pada program berguna untuk mengontrol sebuah blok kode yang akan dieksekusi. Yang dijadikan acuan oleh seleksi kondisi adalah nilai bertipe bool, bisa berasal dari variabel, ataupun hasil operasi perbandingan. Nilai tersebut menentukan blok kode mana yang akan dieksekusi. Go memiliki 2 macam keyword untuk seleksi kondisi, yaitu if else dan switch.

##### **1. If Expression**

If adalah salah satu kata kunci yang digunakan dalam percabangan. Percabangan artinya kita bisa mengeksekusi kode program tertentu ketika suatu kondisi terpenuhi. Hampir semua bahasa pemrograman mendukung if expression.

##### **2. Else if expression**

Terkadang kita butuh membuat beberapa kondisi. Kasus seperti ini dapat menggunakan else if expression. If mendukung short statement sebelum kondisi.

Hal ini sangat cocok untuk membuat statement yang sederhana sebelum melakukan pengecekan terhadap kondisi.

### 3. Switch-Case

Switch merupakan seleksi kondisi yang sifatnya fokus pada satu variabel, lalu kemudian di-cek nilainya. Contoh sederhananya seperti penentuan apakah nilai variabel x adalah: 1, 2, 3, atau lainnya. Perlu diketahui, switch pada pemrograman Go memiliki perbedaan dibanding bahasa lain. Di Go, ketika sebuah case terpenuhi, tidak akan dilanjutkan ke pengecekan case selanjutnya, meskipun tidak ada keyword “break” di situ. Konsep ini berkebalikan dengan switch pada umumnya pemrograman lain (yang ketika sebuah case terpenuhi, maka akan tetap dilanjutkan mengecek case selanjutnya kecuali ada keyword “break”).

## G. Perulangan

Perulangan merupakan proses mengulang dan mengeksekusi blok kode tanpa henti sesuai dengan kondisi yang dijadikan acuan. Biasanya disiapkan variabel untuk iterasi atau penanda kapan perulangan akan dihentikan.

### a. For Loop

For loop merupakan statement perulangan dasar dan cukup sering ditemui. Format for loop yaitu sebagai berikut.

- *Init Statement*: bagian ini akan dieksekusi sebelum perulangan dimulai. Biasanya diisi dengan mendeklarasi variabel iterasi.
- *Condition Expression*: bagian ini akan dicek dan dieksekusi setiap perulangan yang dilakukan, jika true maka perulangan akan terus berjalan hingga kondisi bernilai false.
- *Post Statement*: statement ini akan dieksekusi pada akhir iterasi. Jika terdapat range, maka perulangan akan dieksekusi untuk setiap item pada range.

## CONTOH SOAL

1. Sebuah algoritma digunakan untuk melakukan konversi waktu dari bentuk 24 jam ke dalam bentuk 12 jam.

**Masukan** terdiri dari satu bilangan bulat dengan interval 0 s.d. 23 (inklusif) yang merepresentasikan jam dalam bentuk 24 jam.

**Keluaran** terdiri dari teks yang menyatakan format jam pada masukan dalam bentuk 12 jam.

### 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)
}
```

### Output:

```
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso1\te
13
1 PM

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso1\te
0
12 AM

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso1\te
12
12 PM
```

2. Buatlah program dengan bahasa Go yang digunakan untuk menentukan apakah tanaman tersebut termasuk tanaman karnivora atau tidak. Jika ya, apakah tanaman tersebut asli indonesia atau tidak.

**Masukan** terdiri dari satu string, yakni nama tanaman tersebut.

**Keluaran** berupa teks "Termasuk Tanaman Karnivora" atau "Tidak termasuk Tanaman Karnivora", dan dilanjutkan "Asli Indonesia" atau "Bukan Asli Indonesia". Keluaran sesuai dengan masukan yang diberikan.

**Source Code:**

```
package main
import "fmt"

func main() {
    var namatanaman string
    fmt.Scan(&namatanaman)

    switch namatanaman {
    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")
    }
}
```

**Output:**

```
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso2\coso2.go"
nepenthes
Termasuk tanaman karnivora
Asli Indonesia

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso2\coso2.go"
venus
Termasuk tanaman karnivora
Tidak Asli Indonesia

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso2\coso2.go"
karedok
Tidak termasuk tanaman karnivora
```

3. Buatlah program dalam bahasa Go yang menentukan tarif parkir berdasarkan jenis kendaraan dan durasi parkir. Terdapat tiga jenis kendaraan yang dapat diparkir dengan tarif berbeda berdasarkan durasi parkir. Untuk motor, tarifnya adalah Rp 7.000 untuk durasi 1-2 jam, dan Rp 9.000 jika durasi lebih dari 2 jam. Mobil dikenakan tarif Rp

15.000 untuk 1-2 jam, dan Rp 20.000 untuk durasi lebih dari 2 jam. Sementara itu, truk dikenakan tarif Rp 25.000 untuk 1-2 jam, dan Rp 35.000 jika lebih dari 2 jam.

**Masukan** terdiri dari 2 baris, baris pertama adalah string yang menyatakan jenis kendaraan sedangkan baris kedua merupakan bilangan bulat yang menyatakan durasi parkir dalam jam.

**Keluaran** berupa string yang menampilkan tarif parkir berdasarkan jenis kendaraan dan durasi parkir kendaraan atau "Jenis kendaraan atau durasi parkir tidak valid" jika jenis kendaraan selain dari motor, mobil dan truk.

**Source Code:**

```
package main
import "fmt"

func main() {
    var (
        tipe_kendaraan string
        durasi, tarif int
    )
    fmt.Print("Masukkan jenis kendaraan (Mobil/Motor/Truk): ")
    fmt.Scan(&tipe_kendaraan)
    fmt.Print("Masukkan durasi parkir (dalam jam): ")
    fmt.Scan(&durasi)

    switch {
    case tipe_kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
        tarif = 7000
    case tipe_kendaraan == "Motor" && durasi > 2:
        tarif = 9000
    case tipe_kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
        tarif = 15000
    case tipe_kendaraan == "Mobil" && durasi > 2:
        tarif = 20000
    case tipe_kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
        tarif = 25000
    case tipe_kendaraan == "Truk" && durasi > 2:
        tarif = 35000
    default:
        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
    }

    fmt.Printf("Tarif parkir : Rp. %d\n", tarif)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso3\coso3.go"
Masukkan jenis kendaraan (Mobil/Motor/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif parkir : Rp. 7000

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso3\coso3.go"
Masukkan jenis kendaraan (Mobil/Motor/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif parkir : Rp. 20000

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso3\coso3.go"
Masukkan jenis kendaraan (Mobil/Motor/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif parkir : Rp. 9000

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso3\coso3.go"
Masukkan jenis kendaraan (Mobil/Motor/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif parkir : Rp. 25000

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\coso\coso3\coso3.go"
Masukkan jenis kendaraan (Mobil/Motor/Truk): Sepeda
Masukkan durasi parkir (dalam jam): 2
Jenis kendaraan atau durasi parkir tidak valid
Tarif parkir : Rp. 0
```

## LATIHAN SOAL

### 1. Latihan1

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.")
    }
}
```



Output:

```
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\satu\satu.go"
8.6
Air Layak Minum

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\satu\satu.go"
9
Air Tidak Layak Minum

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\satu\satu.go"
16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.
```

Deskripsi Program:

Program untuk menentukan air minum apakah layak minum atau tidak dengan dilihat dari pH air. Menggunakan seleksi kondisi Switch-Case untuk menentukan air minum berdasarkan pH. Case pertama dengan kondisi  $\text{pH} \geq 6.5$  dan  $\text{pH} \leq 8.5$  ini akan menghasilkan output “Air Layak Minum”, lalu Case 2, karena nilai pH harus antara 0 dan 14, maka kondisi untuk output “Air Tidak Layak Minum” adalah  $\text{pH} \geq 0$  dan  $\text{pH} < 6.5$  atau  $\text{pH} > 8.6$  dan  $\text{pH} \leq 14$ . Kemudian untuk selain kedua kondisi sebelumnya, kita berikan nilai default yaitu “Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.”

## 2. Latihan2

Source Code:

```
package main
import "fmt"

func main() {
    var (
        kendaraan, waktu string
        durasi, biaya    int
    )

    fmt.Scanf("%s %d %s", &kendaraan, &durasi, &waktu)
    if waktu == "menit" && durasi <= 60 {
        durasi = 1
    } else if waktu == "menit" && durasi > 60 {
        durasi = durasi/60 + 1
    }

    switch {
    case (kendaraan == "mobil" || kendaraan == "Mobil") && (waktu == "jam" || waktu == "menit"):
        for i := 0; i < durasi; i++ {
            biaya += 5000
        }
    }
```

```

    }
    fmt.Printf("Rp %d", biaya)
    case (kendaraan == "motor" || kendaraan == "Motor") && (waktu == "jam" ||
waktu == "menit"):
        for i := 0; i < durasi; i++ {
            biaya += 2000
        }
        fmt.Printf("Rp %d", biaya)
    case (kendaraan == "truk" || kendaraan == "Truk") && (waktu == "jam" || waktu
== "menit"):
        for i := 0; i < durasi; i++ {
            biaya += 8000
        }
        fmt.Printf("Rp %d", biaya)
    default:
        fmt.Print("Invalid Input")
    }
}
}

```

Output:

```

D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\dua\dua.go"
motor 3 jam
Rp 6000
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\dua\dua.go"
mobil 1 jam
Rp 5000
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\dua\dua.go"
truk 5 jam
Rp 40000
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\dua\dua.go"
motor 25 menit
Rp 2000
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\dua\dua.go"
mobil 90 menit
Rp 10000

```

#### Deskripsi Program:

Program untuk menghitung biaya parkir perjam untuk kendaraan motor, mobil, dan truk dengan biaya motor adalah 2000 perjam, mobil 5000 perjam, dan truk 8000 perjam. Pada soal, terdapat logika “Jika durasi parkir kurang dari 1 jam, maka durasi tetap dianggap 1 jam”, oleh karena itu kita akan gunakan menit untuk kasus ini. Program memiliki 2 variabel string (kendaraan dan waktu) dan 2 variabel integer (durasi dan biaya). Variabel waktu untuk menentukan durasi dalam satuan jam atau menit.

Pertama kita memiliki seleksi-kondisi if untuk membulatkan menit menjadi jam. Kedua kita memiliki seleksi-kondisi switch-case untuk logik utamanya. Case 1 adalah untuk kendaraan mobil dan terdapat kondisi waktu = jam atau waktu = menit (seluruh Case akan memiliki kondisi ini), kemudian jika kondisi terpenuhi maka Case 1 akan mengeksekusi code untuk biaya perjam, menggunakan perulangan untuk menambah biaya parkir tiap jam, lalu print total biaya. Case 2 untuk kendaraan motor, kurang lebih akan sama dengan Case 1. Kemudian Case 3 untuk kendaraan truk juga kurang

lebih sama dengan Case 1. Lalu terdapat default untuk jika seluruh kondisi tidak ada yang bernilai true pada Case 1, 2 dan 3, akan output “Invalid”.

### 3. Latihan3

Source Code:

```
package main
import "fmt"

func main() {
    var bilangan, b1, b2 int
    var kategori string
    fmt.Scan(&bilangan)

    switch {
    case bilangan%2 != 0 && bilangan == 5:
        b1 = bilangan
        b2 = bilangan + 1
        bilangan = b1 + b2
        kategori = "Ganjil"
        fmt.Printf("Kategori : Bilangan %s\nHasil penjumlahan dengan bilangan
berikutnya %d + %d = %d", kategori, b1, b2, bilangan)

    case bilangan%2 == 0 && bilangan%10 != 0:
        b1 = bilangan
        b2 = bilangan + 1
        bilangan = b1 * b2
        kategori = "Genap"
        fmt.Printf("Kategori : Bilangan %s\nHasil perkalian dengan bilangan berikutnya
%d * %d = %d", kategori, b1, b2, bilangan)

    case bilangan%5 == 0 && bilangan%10 != 0:
        b1 = bilangan
        bilangan = b1 * b1
        kategori = "Kelipatan 5"
        fmt.Printf("Kategori : Bilangan %s\nHasil kuadrat dari %d ^2 = %d", kategori,
b1, bilangan)

    case bilangan%10 == 0:
        b1 = bilangan
        bilangan = b1 / 10
        kategori = "Kelipatan 10"
        fmt.Printf("Kategori : Bilangan %s\nHasil pembagian antara %d / 10 = %d",
kategori, b1, bilangan)
    }
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\tiga\tiga.go"
5
Kategori : Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\tiga\tiga.go"
8
Kategori : Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\tiga\tiga.go"
25
Kategori : Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^2 = 625
D:\Tugas\Alpro\Praktikum\modul11>go run "d:\Tugas\Alpro\Praktikum\modul11\latso\tiga\tiga.go"
20
Kategori : Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
```

Deskripsi Program:

Program operasi aritmatika untuk menentukan suatu bilangan apakah bilangan tersebut merupakan bilangan ganjil, bilangan genap, bilangan kelipatan 5, atau bilangan kelipatan 10. Terdapat bilangan yang memenuhi beberapa kondisi seperti 25 yang merupakan bilangan ganjil dan juga bilangan kelipatan 5, oleh karena itu kita memerlukan kondisi tambahan pada setiap case untuk mengecualikan kondisi dari suatu bilangan yang memiliki lebih dari 1 kategori. Pada Case 1, kita memerlukan 2 kondisi yaitu  $\text{bilangan} \% 2 \neq 0$  untuk cek bilangan ganjil, dan  $\text{bilangan} \neq 5$  agar bilangan 5 tidak ikut kategori “bilangan kelipatan 5”. Pada Case 2, kita menggunakan 2 kondisi yaitu  $\text{bilangan} \% 2 = 0$  untuk cek bilangan genap, dan  $\text{bilangan} \% 10 \neq 0$  untuk mengecualikan bilangan dengan kategori “bilangan kelipatan 10”. Pada Case 3, kita juga menggunakan 2 kondisi yaitu  $\text{bilangan} \% 5 = 0$  untuk cek bilangan kelipatan 5, dan  $\text{bilangan} \% 10 \neq 0$  untuk mengecualikan bilangan dengan kategori “bilangan kelipatan 10”. Pada Case 4, kita hanya memerlukan 1 kondisi yaitu  $\text{bilangan} \% 10 = 0$  untuk cek kategori “bilangan kelipatan 10” karena pengecualian kondisi sudah dilakukan pada ketiga Case sebelumnya.

## DAFTAR PUSTAKA

Noval Agung Prayogo. *Dasar Pemrograman Golang*. Diakses pada 01 Oktober 2024.  
<https://dasarpemrogramangolang.novalagung.com>

Annisa Nur Isnaeni. *Golang — Seleksi Kondisi*. Diakses pada 01 Oktober 2024.  
<https://medium.com/@annisaisna/golang-seleksi-kondisi-f988ead004b4>

Parvez Alam, *Golang for loop example | Golang Loops Tutorial – Phpflow.com*  
<https://medium.com/@parvez1487/golang-for-loop-example-golang-loops-tutorial-phpflow-com-f4b2b0e57944>