

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



**DISUSUN OLEH:  
FEROS PEDROSA VALENTINO  
103112400055  
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

Switch merupakan seleksi kondisi yang sifatnya fokus pada satu variabel, lalu kemudian dicek nilainya. 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 dilanjut mengecek case selanjutnya kecuali ada keyword break).

Sebuah case dapat menampung banyak kondisi. Cara penerapannya yaitu dengan menuliskan nilai pembanding-pembanding variabel yang di-switch setelah keyword case dipisah dengan tanda koma.

Tanda kurung kurawal ( { } ) bisa diterapkan pada keyword case dan default. Tanda ini opsional, boleh dipakai boleh tidak. Bagus jika dipakai pada blok kondisi yang di dalamnya ada banyak statement, dengannya kode akan terlihat lebih rapi.

### Karakteristik Switch-Case

Pada dasarnya switch-case merupakan alternatif dari penggunaan else-if, tetapi pemilihan aksi tidak dilakukan berdasarkan kondisi, tetapi suatu nilai atau operasi tipe data dasar yang tidak menghasilkan boolean.

Penulisan switch case terdiri dari komponen berikut:

- Ekspresi, merupakan operasi tipe data yang menghasilkan nilai selain tipe data boolean.
- Nilai, pilihan case dari ekspresi yang dideklarasikan pada instruksi switch-case.
- Aksi, merupakan kumpulan instruksi yang akan dieksekusi sesuai dengan nilai yang dihasilkan dari ekspresi yang terdapat pada switch-case. Aksi lain hanya akan dieksekusi apabila tidak ada pilihan nilai yang sesuai dengan hasil ekspresi.

Kata kunci default digunakan untuk menyatakan pilihan untuk aksi lain. Selain itu terdapat juga penggunaan switch-case sebagai pengganti if-then, yang mana pada setiap case yang didefinisikan adalah kondisi dari struktur kontrol percabangan.

## CONTOH SOAL

### 1. Coso1

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:

```
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso1\coso1.go"
13
1 PM
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso1\coso1.go"
0
12 AM
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso1\coso1.go"
12
12 PM
```

Deskripsi program:

Program di atas adalah program yang ditulis dalam bahasa Go yang digunakan untuk mengkonversi waktu dari format 24 jam menjadi format 12 jam dengan penambahan label waktu AM atau PM. Program dimulai dengan mendeklarasikan dua variabel, yaitu variabel jam12, jam24 yang bertipe integer dan variabel label yang bertipe string. Selanjutnya, pengguna diminta untuk memasukkan waktu dalam format 24 jam sebagai input. Berdasarkan input tersebut, program akan menentukan nilai jam dalam format 12 jam dan memberikan label waktu yang sesuai:

1. Jika waktu adalah 00:00, maka dikonversi menjadi 12 AM.

2. Jika waktu berada di antara 01:00 hingga 11:00, waktu tetap sama dengan label AM.
3. Jika waktu adalah 12:00, maka waktu tetap menjadi 12 PM.
4. Jika waktu berada di antara 13:00 hingga 23:00, nilai waktu akan dikurangi 12 untuk mendapatkan format 12 jam, dengan label PM.

Setelah perhitungan selesai, hasil konversi waktu ditampilkan dalam format jam12, label menggunakan fungsi `fmt.Println`.

## 2. Coso2

Source code:

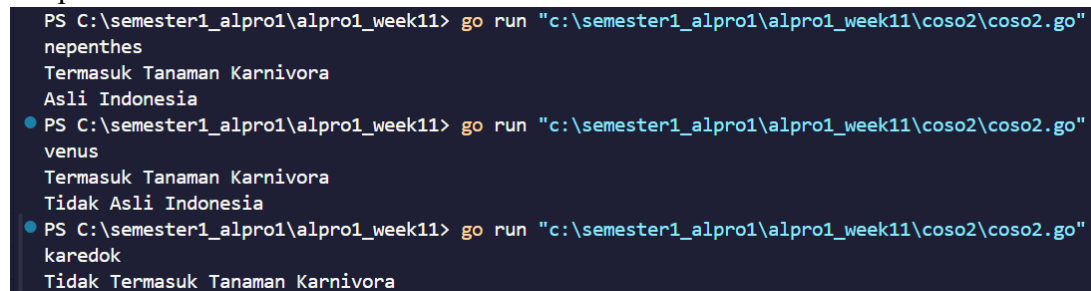
```
package main

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")
    }
}
```

Output:



```
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso2\coso2.go"
nepenthes
Termasuk Tanaman Karnivora
Asli Indonesia
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso2\coso2.go"
venus
Termasuk Tanaman Karnivora
Tidak Asli Indonesia
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso2\coso2.go"
karedok
Tidak Termasuk Tanaman Karnivora
```

Deskripsi program:

Program di atas adalah program yang ditulis dalam bahasa Go yang digunakan untuk mengidentifikasi jenis tanaman berdasarkan nama yang diinputkan oleh pengguna. Pertama deklarasi variabel `nama_tanaman` yang bertipe `string` digunakan untuk menampung nama tanaman yang diinputkan oleh pengguna. Selanjutnya, program akan menerima inputan nama tanaman yang dimasukkan pengguna, program akan menyimpannya dalam variabel `nama_tanaman`. Lalu program memeriksa nilai variabel `nama_tanaman` menggunakan seleksi kondisi `switch`:

1. Jika nama tanaman yang dimasukkan adalah "nepenthes" atau "drosera", program mencetak bahwa tanaman tersebut "Termasuk Tanaman Karnivora" dan "Asli Indonesia".
2. Jika nama tanaman yang dimasukkan adalah "venus" atau "sarracenia", program mencetak bahwa tanaman tersebut "Termasuk Tanaman Karnivora" tetapi "Tidak Asli Indonesia".
3. Untuk nama tanaman selain yang disebutkan di atas, program mencetak bahwa tanaman tersebut "Tidak Termasuk Tanaman Karnivora".

Terakhir, hasil identifikasi dicetak ke layar menggunakan fungsi `fmt.Println`.

### 3. Coso3

Source code:

```
package main

import "fmt"

func main() {
    var tipe_kendaraan string
    var durasi, tarif int

    fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/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", tarif)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso3\coso3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso3\coso3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir : Rp 20000
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso3\coso3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir : Rp 9000
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso3\coso3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir : Rp 25000
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\coso3\coso3.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
```

### Deskripsi program:

Program ini adalah program yang ditulis dengan bahasa Go yang digunakan untuk menghitung tarif parkir berdasarkan jenis kendaraan dan durasi parkir yang diinputkan oleh pengguna. Pertama, deklarasikan variabel tipe\_kendaraan bertipe string, dan variabel durasi, tarif bertipe integer. Selanjutnya, pengguna diminta untuk memasukkan jenis kendaraan yang akan diparkir, yang dapat berupa Motor, Mobil, atau Truk, serta durasi parkir dalam jam. Setelah menerima input, program akan menentukan tarif parkir berdasarkan kombinasi jenis kendaraan dan durasi yang diberikan. Terdapat beberapa aturan tarif: untuk Motor, tarif Rp 7.000 untuk durasi 1 hingga 2 jam, dan Rp 9.000 untuk durasi lebih dari 2 jam; untuk Mobil, tarif Rp 15.000 untuk durasi 1 hingga 2 jam, dan Rp 20.000 untuk durasi lebih dari 2 jam; sedangkan untuk Truk, tarif Rp 25.000 untuk durasi 1 hingga 2 jam, dan Rp 35.000 untuk durasi lebih dari 2 jam. Jika input yang diberikan tidak sesuai dengan jenis kendaraan yang valid atau durasi parkir yang ditentukan, program akan menampilkan pesan bahwa “Jenis kendaraan atau durasi parkir tidak valid”. Setelah perhitungan selesai, program akan mencetak tarif parkir yang sesuai.

## LATIHAN SOAL

### 1. Latihan soal 1

Source code:

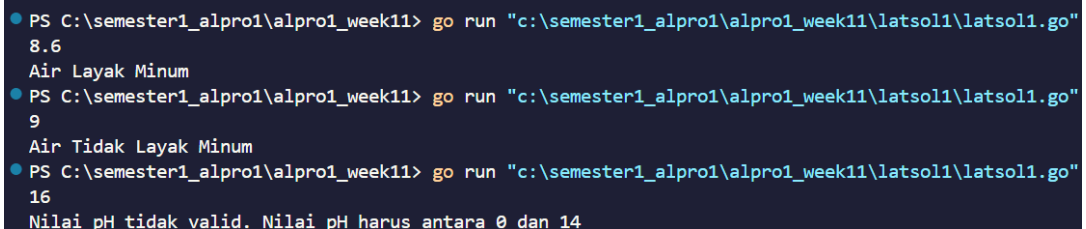
```
package main

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 <= 14 && ph > 8.6):
        fmt.Println("Air Tidak Layak Minum")
    default:
        fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14")
    }
}
```

Output:



```
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol1\latsol1.go"
8.6
Air Layak Minum
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol1\latsol1.go"
9
Air Tidak Layak Minum
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol1\latsol1.go"
16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14
```

Deskripsi program:

Program di atas ditulis dalam bahasa pemrograman Go dan digunakan untuk mengevaluasi kualitas air berdasarkan nilai pH yang diinputkan pengguna. Pertama, deklarasi variabel `ph` bertipe `float64` untuk menyimpan nilai pH. Selanjutnya, pengguna diminta untuk memasukkan nilai pH. Kemudian, program menggunakan seleksi kondisi `switch` untuk menentukan kategori kualitas air berdasarkan nilai pH yang diberikan. Jika nilai pH berada dalam rentang 6.5 hingga 8.6, program akan menampilkan output “Air Layak Minum”. Namun, jika nilai pH berada di luar rentang tersebut (yaitu lebih dari 0 hingga kurang dari 6.5 atau lebih dari 8.6 hingga 14), program akan menampilkan output “Air Tidak Layak Minum”. Jika nilai pH yang dimasukkan tidak valid (misalnya kurang dari 0 atau lebih dari 14), program akan menampilkan output “Nilai pH tidak valid. Nilai pH harus antara 0 dan 14”.



## 2. Latihan soal 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 kendaraan {
    case "motor":
        tarif = durasi * 2000

    case "mobil":
        tarif = durasi * 5000

    case "truk":
        tarif = durasi * 8000

    default:
        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
    }
    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol2\latsol2.go"
Masukkan jenis kendaraan (motor/mobil/truk): motor
Masukkan durasi parkir (dalam jam): 3 jam
Tarif Parkir: Rp 6000
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol2\latsol2.go"
Masukkan jenis kendaraan (motor/mobil/truk): mobil
Masukkan durasi parkir (dalam jam): 1 jam
Tarif Parkir: Rp 5000
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol2\latsol2.go"
Masukkan jenis kendaraan (motor/mobil/truk): truk
Masukkan durasi parkir (dalam jam): 5 jam
Tarif Parkir: Rp 40000
```

Deskripsi program:

Program di atas ditulis dalam bahasa pemrograman Go dan digunakan untuk menghitung tarif parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Pertama, deklarasikan tiga variabel: kendaraan bertipe string untuk menyimpan jenis kendaraan, durasi bertipe int untuk menyimpan lama waktu parkir dalam jam, dan tarif bertipe int untuk menyimpan total biaya parkir.

Pengguna diminta untuk memasukkan jenis kendaraan (motor, mobil, atau truk) serta durasi parkir dalam jam. Jika durasi parkir yang dimasukkan kurang dari 1 jam, program secara otomatis menetapkan durasi menjadi 1 jam untuk memastikan bahwa biaya parkir tetap dihitung. Program kemudian menggunakan seleksi kondisi switch untuk menentukan tarif parkir berdasarkan jenis kendaraan yang dipilih: Rp 2.000 per jam untuk motor, Rp 5.000 per jam untuk mobil, dan Rp 8.000 per jam untuk truk. Jika pengguna memasukkan jenis kendaraan yang tidak valid, program akan menampilkan output “Jenis kendaraan atau durasi parkir tidak valid”. Terakhir, program mencetak total tarif parkir yang harus dibayar berdasarkan perhitungan yang telah dilakukan.

### 3. Latihan soal 3

Source code:

```
package main

import "fmt"

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

Output:

```
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol3\latsol3.go"
5
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol3\latsol3.go"
8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol3\latsol3.go"
25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^2 = 625
PS C:\semester1_alpro1\alpro1_week11> go run "c:\semester1_alpro1\alpro1_week11\latsol3\latsol3.go"
20
Kategori: Bilangan Keliapatan 10
Hasil pembagian antara 20 / 10 = 2
```

Deskripsi program:

Program di atas ditulis dalam bahasa pemrograman Go dan digunakan untuk mengkategorikan bilangan yang dimasukkan oleh pengguna serta melakukan perhitungan berdasarkan kategori tersebut. Pertama, deklarasikan variabel bilangan, hasil yang bertipe int. Selanjutnya pengguna diminta untuk memasukkan suatu bilangan. Program kemudian menggunakan seleksi kondisi switch untuk memeriksa apakah bilangan tersebut genap atau ganjil. Jika bilangan tersebut ganjil dan merupakan kelipatan 5 serta lebih besar dari 5, program akan menampilkan output “Kategori: Bilangan Kelipatan 5” dan program akan menghitung kuadrat bilangan tersebut serta menampilkan hasil kuadratnya. Jika bilangan ganjil tetapi tidak memenuhi syarat kelipatan 5, program menampilkan output “Kategori: Bilangan Ganjil” dan program akan menghitung hasil penjumlahan bilangan dengan bilangan berikutnya serta menampilkan hasil penjumlahannya. Sebaliknya, jika bilangan tersebut genap dan merupakan kelipatan 10 serta lebih besar dari 10, program akan menampilkan output “Kategori: Bilangan Kelipatan 10” dan program akan menghitung hasil bagi bilangan tersebut dengan 10 serta menampilkan hasil baginya. Jika bilangan genap tetapi tidak memenuhi syarat kelipatan 10, program akan menampilkan output “Kategori: Bilangan Genap” dan program akan menghitung hasil perkalian bilangan dengan bilangan berikutnya serta menampilkan hasil perkaliannya. Jika tidak ada kondisi yang cocok, program akan menampilkan output “Masukkan angka.”.

## DAFTAR PUSTAKA

A.13. Seleksi Kondisi. (n.d.). Retrieved from  
<https://dasarpemrogramangolang.novalagung.com/A-seleksi-kondisi.html>

Prasti Eko Yunanto, S.T., M.Kom. (2004). MODUL PRAKTIKUM 11 - SWITCH-CASE  
ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA (MODUL 11)