

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11

SWITCH - CASE



Disusun oleh:

Manggala Patra Raditya

109082500179

S1IF-13-02

Asisten Praktikum

Adithana dharma putra

Alfin Ilham Berlianto

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

1. Guided 1 Source Code

```
package main

import "fmt"

func main() {
    var jam int
    fmt.Scan(&jam)
    var konversi int
    var periode string
    switch (jam) {
        case 0:
            konversi = 12;
            periode = "AM";
        case 12:
            konversi = 12;
            periode = "PM"
        default:
            if (jam < 24){
                konversi = jam - 12;
                periode = "PM"
            }else{
                konversi = jam
                periode = "AM"
            }
    }
    fmt.Println(konversi, periode)
}
```

Screenshoot program

The screenshot shows a code editor interface with several tabs open. The active tab is 'Guide1.go' which contains the following Go code:

```
func main() {
    var jam int
    fmt.Scan(&jam)
    var konversi int
    var periode string
    switch (jam) {
        case 0:
            konversi = 12;
            periode = "AM";
        case 12:
            konversi = 12;
            periode = "PM";
        default:
    }
}
```

Below the code editor is a terminal window showing the execution of the program:

```
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide1\Guide1.go"
0
12 AM
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide1\Guide1.go"
12
12 PM
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11>
```

A floating window titled 'Build with Agent' is visible on the right side of the screen.

Deskripsi program

Program ini dibuat untuk mengubah jam dari format 24 jam menjadi format 12 jam lengkap dengan keterangan AM atau PM. Pertama, program meminta pengguna memasukkan sebuah angka sebagai jam. Angka tersebut kemudian disimpan dalam variabel **Jam**. Setelah itu, program menyiapkan dua variabel lain yaitu **konversi** untuk hasil jam setelah diubah, dan **periode** untuk menentukan apakah waktunya AM atau PM.

Proses pengecekan dilakukan dengan **switch**. Kalau jam yang dimasukkan adalah 0, program otomatis mengubahnya menjadi 12 AM karena jam 0 dalam format 24 jam itu sama dengan jam 12 malam. Kalau jamnya 12, maka hasilnya tetap 12 tetapi periodenya menjadi PM karena itu adalah jam 12 siang.

Untuk jam yang bukan 0 dan 12, program masuk ke bagian default. Jika jam kurang dari 24, maka program akan mengonversinya menjadi jam PM dengan cara mengurangi nilai jam tersebut dengan 12. Misalnya jika inputnya 15, hasilnya menjadi 3 PM. Namun kalau input jam lebih dari 24, program tidak mengonversinya dan hanya menampilkan kembali jam aslinya sambil memberi periode AM, meskipun sebenarnya nilai tersebut sudah tidak valid sebagai jam.

Di bagian akhir, program menampilkan hasil konversinya. Intinya, program ini bertugas mengubah jam 24 jam menjadi format 12 jam agar lebih mudah dipahami, lengkap dengan penanda AM atau PM.

2. Guided 2 Source Code

```
package main

import "fmt"

func main() {
    var tanaman string

    fmt.Scan(&tanaman)

    switch tanaman{
        case "nepenthes":
            fmt.Println("termasuk tanaman Karnivora Asli
Indonesia")
        case "venus":
            fmt.Println("termasuk tanaman Karnivora bukan
asli Indonesia")
        default:
            fmt.Println("Tidak termasuk tanaman
Karnivora")
    }
}
```

Screenshot program

The screenshot shows a Go code editor interface. The left sidebar displays a file tree under 'MODUL 11' with files like Guide1.go, Guide2.go, and Soal1.go. The main editor window contains the following Go code:

```
func main() {
    fmt.Scan(&tanaman)

    switch tanaman{
        case "nepenthes":
            fmt.Println("termasuk tanaman Karnivora Asli Indonesia")
        case "venus":
            fmt.Println("termasuk tanaman Karnivora bukan asli Indonesia")
        default:
            fmt.Println("Tidak termasuk tanaman Karnivora")
    }
}
```

Below the editor is a terminal window showing command-line interactions:

```
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide2\Guide2.go"
nepenthes
termasuk tanaman Karnivora Asli Indonesia
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide2\Guide2.go"
venus
termasuk tanaman Karnivora bukan asli Indonesia
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide2\Guide2.go"
karedok
Tidak termasuk tanaman Karnivora
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11>
```

The status bar at the bottom indicates the terminal is in UTF-8 mode.

Deskripsi program

Program ini sebenarnya sederhana. Program ini meminta pengguna untuk memasukkan nama sebuah tanaman, lalu berdasarkan nama itu program menentukan apakah tanaman tersebut termasuk tanaman karnivora atau bukan. Pertama, input disimpan dalam variabel `tanaman`. Setelah itu, program menggunakan `switch` untuk membandingkan nama yang dimasukkan.

Kalau pengguna mengetik "`nepenthes`", program akan langsung menampilkan bahwa itu adalah tanaman karnivora yang asli dari Indonesia. Jika yang dimasukkan adalah "`venus`", maka program memberi informasi bahwa tanaman tersebut memang karnivora, tetapi bukan berasal dari Indonesia. Dan kalau nama tanaman yang ditulis tidak cocok dengan dua pilihan itu, program memberikan pesan bahwa tanaman tersebut bukan termasuk tanaman karnivora.

Secara keseluruhan, program ini hanya bertugas mengecek nama tanaman dan memberikan keterangan singkat berdasarkan nama yang cocok.

3. Guided 3 Source Code

```
package main

import "fmt"

func main() {
    var jenis
    string
    var durasi,
    tarif int
    fmt.Println("jenis kendaraan
(motor/mobil/truk) :
")
    fmt.Scanln(&jenis)
    fmt.Println("durasi parkir (dalam
jam) : ")
    fmt.Scanln(&durasi)
    switch jenis {
        case "motor":
            if
durasi >= 1 &&
durasi <= 2 {
                tarif =
7000
            } else
if durasi > 2 {
                tarif =
9000
            }
        case "mobil":
```

```
        if
durasi >= 1 &&
durasi <= 2 {
    tarif =
15000
} else
if durasi > 2 {
    tarif =
20000
}
case "truk":
    if
durasi >= 1 &&
durasi <= 2 {

    tarif = 25000
} else
if durasi > 2 {

    tarif = 30000
}
default:

fmt.Println("J
enis kendaraan atau
durasi parkir tidak
valid")
return
}
fmt.Println("T
arif Parkir: Rp",
tarif)
}
```

Screenshot Program

```
Guide3 > Guide3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var jenis string
7     var durasi, tarif int
8
9     fmt.Print("jenis kendaraan (motor/mobil/truk): ")
er\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide3\Guide3.go"
jenis kendaraan (motor/mobil/truk): motor
durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:\Users\us
er\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide3\Guide3.go"
jenis kendaraan (motor/mobil/truk): mobil
durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:\Users\us
er\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide3\Guide3.go"
jenis kendaraan (motor/mobil/truk): truk
durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:\Users\us
er\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Guide3\Guide3.go"
jenis kendaraan (motor/mobil/truk): sepeda
durasi parkir (dalam jam): 2
Jenis kendaraan atau durasi parkir tidak valid
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> 
```

Deskripsi program

Program ini dibuat untuk menghitung tarif parkir berdasarkan jenis kendaraan dan lama durasinya. Ketika program dijalankan, pengguna diminta memasukkan jenis kendaraan terlebih dahulu—apakah **motor**, **mobil**, atau **truk**. Setelah itu, pengguna juga memasukkan durasi parkir dalam satuan jam. Kedua input tersebut kemudian digunakan sebagai dasar perhitungan tarif.

Program menggunakan `switch` untuk memisahkan aturan tarif berdasarkan jenis kendaraannya.

- Jika kendaraan yang dipilih adalah **motor**, tarifnya Rp7.000 untuk durasi 1–2 jam, dan akan menjadi Rp9.000 jika durasinya lebih dari 2 jam.
- Untuk **mobil**, tarifnya Rp15.000 untuk 1–2 jam, dan Rp20.000 jika lebih dari 2 jam.
- Sedangkan untuk **truk**, tarifnya Rp25.000 untuk 1–2 jam, dan Rp30.000 jika lewat dari 2 jam.

Jika pengguna memasukkan jenis kendaraan yang tidak sesuai pilihan, program langsung menampilkan pesan bahwa input tidak valid dan proses akan dihentikan.

Setelah semua kondisi terpenuhi dan tarif berhasil dihitung, program menampilkan hasil akhirnya berupa besarnya tarif parkir sesuai jenis kendaraan dan durasi parkir yang dimasukkan. Program ini membantu mempermudah menentukan biaya parkir berdasarkan aturan tarif yang sudah ditetapkan.

Tugas

1. Tugas 1 Source code

```
package main

import "fmt"

func main() {
    var ph
    float64

    fmt.Println("Masukkan nilai 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")  
}  
}
```

Screenshot Program

The screenshot shows the VS Code interface with the following details:

- Explorer:** Shows a tree view of files under "MODUL 11". The "Soal1" folder is expanded, showing "Soal1.go".
- Editor:** Displays the content of "Soal1.go". The code uses a switch statement to check if the pH value is between 0 and 14. If it's not, it prints an invalid message. If it is, it checks if it's between 6.5 and 8.6, printing a minimum message if so.
- Terminal:** Shows the command "go run" being used to execute the program. It then prompts for a pH value (8.6), which is then printed as "Air layak minum".
- Bottom Status Bar:** Shows the file path "C:\Users\user\OneDrive\...Modul 11\Soal1\Soal1.go", line 18, column 45, tab size 4, and other system information.

Deskripsi program

Program ini dibuat untuk mengevaluasi kelayakan air berdasarkan nilai pH yang dimasukkan oleh pengguna. Ketika program dijalankan, pengguna diminta menginput angka pH, kemudian angka tersebut diproses untuk menentukan apakah air termasuk layak dikonsumsi atau tidak. Nilai pH sendiri merupakan ukuran tingkat keasaman atau kebasaan suatu larutan, dengan rentang yang sah berada di antara 0 hingga 14.

Setelah nilai pH diterima, program menggunakan struktur switch tanpa ekspresi—yang artinya setiap kondisi akan dievaluasi sebagai sebuah pernyataan logis. Kondisi pertama memeriksa apakah nilai pH berada di luar rentang 0 sampai 14. Jika iya, program langsung memberi tahu bahwa input tidak valid dan nilai tersebut tidak dapat digunakan untuk menentukan kualitas air.

Jika nilai pH berada dalam rentang yang benar, program lanjut mengecek apakah nilai tersebut berada pada kisaran 6.5 hingga 8.6. Rentang ini dianggap ideal karena menunjukkan bahwa air berada pada tingkat keasaman yang aman dan memenuhi standar umum kelayakan air minum. Bila nilai pH berada dalam interval tersebut, program menyatakan bahwa air tersebut layak minum.

Namun, jika nilai pH masih dalam batas valid tetapi berada di luar rentang ideal, program menyimpulkan bahwa air tersebut tidak layak dikonsumsi. Hal ini mencerminkan bahwa kualitas air terlalu asam atau terlalu basa sehingga berpotensi tidak aman.

Secara keseluruhan, program ini berfungsi sebagai alat sederhana untuk menilai kualitas air berdasarkan parameter pH, serta memberikan umpan balik cepat mengenai kelayakan air tersebut.

2. Tugas 2 Source code

```
package main

import "fmt"

func main() {
    var jenis string
    var durasi int
    var tarifPerJam int
    var total int
    fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk) : ")
    fmt.Scan(&jenis)
    fmt.Print("Masukkan durasi parkir (jam) : ")
    fmt.Scan(&durasi)

    if durasi < 1 {
        durasi = 1
    }

    switch jenis {
    case "motor":
        tarifPerJam = 2000

    case "mobil":
        tarifPerJam = 5000

    case "truk":
        tarifPerJam = 8000

    default:
        fmt.Println("Jenis kendaraan tidak valid.")
        return
    }
}
```

```

        total = tarifPerJam * durasi
        fmt.Println("Total biaya parkir: Rp", total)
    }

```

Screenshot Program

The screenshot shows a Go code editor interface with the following details:

- File Explorer:** Shows files like e1.go, Guide1.go, Guide2.go, Guide3.go, Soal1.go, Soal2.go, and Soal3.go.
- Code Editor:** Displays the content of Soal2.go:

```

func main() {
    tarifPerJam = 5000
    case "truk":
        tarifPerJam = 8000
    default:
        fmt.Println("Jenis kendaraan tidak valid.")
        return
}

```
- Terminal:** Shows the command `go run "c:/Users/user/OneDrive/Dokumen/CODINGAN MANGGALA/Modul 11/Soal2/Soal2.go"` being run, followed by user input for vehicle type ("motor") and duration (3), resulting in a total fee of Rp 6000.
- Status Bar:** Includes information like line number (Ln 38), column (Col 2), tab size (4), and date (01/12/2025).

Deskripsi program

Program ini digunakan untuk menghitung total biaya parkir berdasarkan jenis kendaraan dan lama durasi parkir. Pengguna terlebih dahulu memasukkan jenis kendaraan—motor, mobil, atau truk—serta durasi parkir dalam jam. Jika durasi yang dimasukkan kurang dari 1 jam, program otomatis mengatur durasinya menjadi 1 jam agar tetap masuk akal.

Melalui struktur switch, setiap jenis kendaraan memiliki tarif per jam yang berbeda: motor dikenakan Rp2.000 per jam, mobil Rp5.000 per jam, dan truk Rp8.000 per jam. Jika pengguna memasukkan jenis kendaraan yang tidak sesuai, program langsung menampilkan pesan bahwa input tidak valid.

Setelah tarif per jam ditentukan, program menghitung total biaya dengan mengalikan tarif dengan durasi, kemudian menampilkan hasilnya. Program ini membantu memberikan perhitungan tarif parkir secara sederhana dan otomatis berdasarkan kategori kendaraan.

3. Tugas 3 Source code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("M
asukan bilangan
bulat: ")

    fmt.Scan(&n)
    switch {
    case n%10 ==
0:
        hasil
        := n / 10
        fmt.Println(
"Kategori:
Bilangan
Kelipatan 10")
        fmt.Printf("

Hasil pembagian
antara %d / 10 =
%d\n", n, hasil)
    case n%5 ==
0 && n != 5:
        hasil
        := n * n
        fmt.Println(
"Kategori:
Bilangan
Kelipatan 5")
```

```
    fmt.Printf("

Hasil kuadrat

dari %d ^ 2 =
%d\n", n, hasil)

    case n%2 ==

0:

    hasil

:= n * (n + 1)

    fmt.Println(

"Kategori:

Bilangan Genap")

    fmt.Printf("

Hasil perkalian

dengan bilangan

berikutnya %d *

%d = %d\n", n,
n+1, hasil)

    default:

    hasil

:= n + (n + 1)

    fmt.Println(

"Kategori:

Bilangan Ganjil")

    fmt.Printf("

Hasil penjumlahan

dengan bilangan

berikutnya %d +
%d = %d\n", n,
n+1, hasil)

    }

}
```

Screenshoot program

The screenshot shows a Go code editor interface with several tabs open. The main tab displays a Go program named Soal3.go. The code uses a switch statement to categorize a number based on its remainder when divided by 2. It handles cases for even numbers, odd numbers, and a default case. The output window shows the execution of the program, where it prompts for a number, categorizes it, and performs arithmetic operations like squaring or multiplying by the next integer.

```
func main() {
    n := int(input("Masukan bilangan bulat: "))
    if n%2 == 0 {
        hasil := n * (n + 1)
        fmt.Println("Kategori: Bilangan Genap")
        fmt.Println("Hasil perkalian dengan bilangan berikutnya", hasil)
    } else {
        hasil := n * n
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Println("Hasil penjumlahan dengan bilangan berikutnya", hasil)
    }
}
```

Output window:

```
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:/Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Soal3\Soal3.go"
Masukan bilangan bulat: 5
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:/Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Soal3\tempCodeRunnerFile.go"
Masukan bilangan bulat: 8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:/Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Soal3\tempCodeRunnerFile.go"
Masukan bilangan bulat: 25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^ 2 = 625
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11> go run "c:/Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 11\Soal3\tempCodeRunnerFile.go"
Masukan bilangan bulat: 20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
```

Deskripsi program

Program ini berfungsi untuk mengelompokkan sebuah bilangan bulat ke dalam kategori tertentu, lalu melakukan operasi sesuai kategorinya. Setelah pengguna memasukkan sebuah angka, program mengeceknya menggunakan switch tanpa ekspresi.

Jika bilangan tersebut merupakan kelipatan 10, program membaginya dengan 10. Jika bilangan itu kelipatan 5 (selain angka 5), program menghitung kuadratnya. Jika angka tersebut genap namun bukan kelipatan 5 atau 10, program mengalikan angka itu dengan angka setelahnya. Dan jika tidak memenuhi semua kondisi sebelumnya, berarti angka tersebut ganjil, sehingga program menjumlahkannya dengan bilangan berikutnya.

Hasil perhitungan akan langsung ditampilkan sesuai kategori bilangan tersebut.