

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11

SWITCH-CASE



Disusun oleh:

FARID HERDIYANTO VITASANDI

109082500123

S1IF-13-02

Asisten Praktikum

Adithana dharma putra

Alfin Ilham Berlianto

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 jam int

    fmt.Print("Masukkan jam (0 - 23): ")

    fmt.Scanln(&jam)

    switch{
        case jam == 0:
            fmt.Println("12 AM")
        case jam == 12:
            fmt.Println("12 PM")
        case jam > 0 && jam < 12:
            fmt.Printf("%d AM\n", jam)
        case jam > 12 && jam < 24:
            fmt.Printf("%d PM\n", jam-12)
        default:
            fmt.Printf("Input tidak valid")
    }
}
```

Screenshot program

The screenshot shows a Go development environment in Visual Studio Code. The left sidebar displays a project structure under 'PRAKTIKUM-11' with files like main.go, pH_air.go, main.go, and main.go. The main editor shows a Go program that converts a 24-hour input into a 12-hour format (AM/PM). A floating terminal window titled 'NAMA FAR' shows the execution of the program and its output:

```
NAMA: FARID HERDIYANTO VITASANDI  
KELAS: IF-02  
NIM: 109082500123
```

The terminal also shows the command `go run main.go` and the user input `Masukkan jam (0 - 23): 13`, resulting in the output `1 PM`.

Deskripsi program

Program di atas meminta User untuk memasukkan jam dalam format 24 jam, kemudian menggunakan struktur *switch* untuk mengonversinya ke format 12 jam. Jika nilai yang dimasukkan adalah 0, program akan menampilkan “12 AM”, sedangkan jika nilainya 12, program akan menampilkan “12 PM”. Untuk rentang 1 hingga 11, program menampilkan nilai jam dengan akhiran “AM”, dan untuk rentang 13 hingga 23, program mengurangi nilai jam dengan 12 sebelum menampilkannya dengan “PM”. Apabila nilai yang dimasukkan berada di luar rentang 0 hingga 23, program akan memberikan keluaran berupa pesan bahwa input tidak valid.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var tanaman string

    fmt.Print("Masukkan nama tanaman: ")

    fmt.Scan(&tanaman)

    switch tanaman{
        case "nepenthes", "drosera":
            fmt.Println("Termasuk tanaman karnivora")
            fmt.Println("Asli Indonesia")
        case "venus", "sarracenia":
            fmt.Println("Termasuk tanaman karnivora")
            fmt.Println("Bukan Asli Indonesia")
        default:
            fmt.Println("Tidak termasuk tanaman karnivora")
    }
}
```

Screenshot program

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure under "PRAKTIKUM-11" with files like main.go, parkir.go, phair.go, and pola_aritmatika.go.
- Code Editor:** Displays a Go code snippet for identifying carnivorous plants. It includes a switch statement for "nepenthes" and "drosera" (both labeled as "Asli Indonesia"), and another for "venus" and "sarracenia" (both labeled as "Tersusuk tanaman karnivora"). A default case handles other inputs.
- Terminal:** Shows the command "go run main.go" being run, followed by the output: "Masukkan nama tanaman: karedok", "Tidak termasuk tanaman karnivora".
- Output Panel:** Shows the output "NAMA: FARID HERDIYANTO VITASANDI", "KELAS: IF-02", and "NIM: 109082500123".
- Status Bar:** Shows file path ("D:\Documents\cooyeah\alpro1\praktikum-11\master"), character count (80 characters), and encoding (UTF-8).

Deskripsi program

Program di atas dibuat untuk membaca input berupa nama tanaman yang dimasukkan User, kemudian memeriksanya menggunakan struktur *switch* untuk menentukan kategorinya. Jika User memasukkan “nepenthes” atau “drosera”, program akan menampilkan bahwa tanaman tersebut termasuk tanaman karnivora dan berasal dari Indonesia. Apabila input berupa “venus” atau “sarracenia”, program tetap mengelompokkan tanaman tersebut sebagai tanaman karnivora, tetapi menegaskan bahwa keduanya bukan tanaman asli Indonesia. Jika nama tanaman yang dimasukkan tidak termasuk dalam daftar tersebut, program akan menampilkan bahwa tanaman tersebut bukan bagian dari kelompok tanaman karnivora.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var lama_parkir int

    fmt.Println("Masukkan jenis kendaraan (Mobil/Motor/Truk) :")
    fmt.Scan(&kendaraan)

    fmt.Println("Masukkan durasi waktu parkir (jam) : ")
    fmt.Scan(&lama_parkir)

    switch kendaraan{
        case "mobil":
            if lama_parkir <= 2{
                fmt.Println("Rp.15.000")
            } else {
                fmt.Println("Rp.20.000")
            }
        case "motor":
            if lama_parkir <= 2{
                fmt.Println("Rp.7.000")
            }
    }
}
```

```
        }else{
            fmt.Println("Rp.9.000")
        }

    case "truk":
        if lama_parkir <= 2{
            fmt.Println("Rp.25.000")
        }else{
            fmt.Println("Rp.35.000")
        }
    default:
        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
    }
}
```

Screenshot program

The screenshot shows a code editor interface with a Go file named `main.go` open. The code defines a `main()` function that prompts the user for the type of vehicle and parking duration, then calculates and prints the fee based on a switch statement. The terminal window shows the execution of the program and its output for a car parked for 4 hours.

```
File Edit Selection View ... ← → praktikum-11
EXPLORER PRAKTIKUM-11
main.go
func main() {
    // ...
    //! seat guided 3
    var kendaraan string
    var lama_parkir int
    fmt.Println("Masukkan jenis kendaraan (Mobil/Motor/Truk): ")
    fmt.Scan(&kendaraan)
    fmt.Println("Masukkan durasi waktu parkir (jam): ")
    fmt.Scan(&lama_parkir)

    switch kendaraan{
        case "mobil":
            if lama_parkir <= 2{
                fmt.Println("Rp.15.000")
            } else {
                fmt.Println("Rp.20.000")
            }
        case "motor":
            if lama_parkir <= 2{
                fmt.Println("Rp.7.000")
            } else{
                fmt.Println("Rp.9.000")
            }
        case "truk":
            if lama_parkir <= 2{
                fmt.Println("Rp.25.000")
            } else{
                fmt.Println("Rp.35.000")
            }
        default:
            fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
    }
}

NAMA: FARID HERDIYANTO VITASANDI
KELAS: IF-02
NIM: 109082500123|
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PAGES

```
-[Documents]\codeyear\alpro1\praktikum-11\master
↳ go run main.go
Masukkan jenis kendaraan (Mobil/Motor/Truk): mobil
Masukkan durasi waktu parkir (jam): 4
Rp.20.000
-[Documents]\codeyear\alpro1\praktikum-11\master
```

Deskripsi program

Kode di atas meminta User untuk memasukkan jenis kendaraan serta durasi parkir, kemudian memproses keduanya menggunakan struktur *switch* untuk menentukan biaya yang harus dibayar. Jika pengguna memilih “mobil”, program akan menampilkan biaya Rp15.000 untuk durasi parkir hingga dua jam, dan Rp20.000 jika durasinya lebih lama. Untuk jenis “motor”, biaya yang dikenakan adalah Rp7.000 apabila waktu parkir tidak lebih dari dua jam, dan Rp9.000 jika melewati batas tersebut. Pada kendaraan “truk”, tarifnya ditetapkan sebesar Rp25.000 untuk durasi hingga dua jam dan Rp35.000 apabila durasinya lebih panjang. Jika input jenis kendaraan tidak sesuai dengan daftar yang tersedia, program akan menampilkan pesan bahwa jenis kendaraan atau durasi parkir tidak valid.

TUGAS

1. Soal 1

Source Code

```
package main

import "fmt"

func main() {
    var kadar_pH float64

    fmt.Println("Masukkan kadar pH air (0-14): ")

    fmt.Scan(&kadar_pH)

    if kadar_pH < 0 || kadar_pH > 14{
        fmt.Println("Nilai pH tidak valid. Nilai pH harus
antara 0 - 14.")

        return
    }

    switch {
    case kadar_pH >= 6.5 && kadar_pH <= 8.6:
        fmt.Println("Air layak minum")
    default:
        fmt.Println("Air tidak layak minum")
    }
}
```

Screenshot program

The screenshot shows a Go development environment in a dark-themed IDE. The left sidebar displays a project structure under 'PRAKTIKUM-11' with files like 'main.go', 'ph_air.go', and 'pole_aritmatika.go'. The main editor area contains the following Go code:

```
File Edit Selection View ... ← → 🔍 praktikum-11
EXPLORER PRAKTIKUM-11
    parkir
        main.go
    ph_air
        main.go
    pole_aritmatika
        main.go
    main.go
    main.go ph_air
    main.go parkir
    main.go pole_aritmatika

ph_air > main > main
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var kadar_ph float64
8
9     fmt.Println("Masukkan kadar pH air (0-14): ")
10    fmt.Scan(&kadar_ph)
11
12    if kadar_ph < 0 || kadar_ph > 14{
13        fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 - 14.")
14        return
15    }
16
17    switch {
18        case kadar_ph >= 6.5 && kadar_ph <= 8.6:
19            fmt.Println("Air layak minum")
20        default:
21            fmt.Println("Air tidak layak minum")
22    }
23 }
```

The terminal below shows the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
- Documents\coolyeah\alpro1\praktikum-11\pH_air\master
  ↵ go run main.go
  Masukkan kadar pH air (0-14): 16
  Nilai pH tidak valid. Nilai pH harus antara 0 - 14.
<- Documents\coolyeah\alpro1\praktikum-11\pH_air\master
  ↵ 
```

A modal window titled 'NAMA FAF' displays student information:

NAMA:	FARID HERDIYANTO VITASANDI
KELAS:	IF-02
NIM:	109082500123

The status bar at the bottom indicates the terminal has 23 lines, 2 characters, and is in UTF-8 encoding.

Deskripsi program

Program di atas adalah program yang dibuat untuk melakukan pengecekan awal untuk memastikan bahwa nilai pH suatu air berada dalam rentang 0 hingga 14. Jika berada di luar batas tersebut, program akan langsung menghentikan proses dan menampilkan pesan bahwa input tidak valid. Setelah lolos validasi, program akan menggunakan struktur *switch* berbasis kondisi untuk menentukan kelayakan air. Jika kadar pH berada pada rentang 6.5 hingga 8.6, program akan menyatakan bahwa air tersebut layak minum, sedangkan nilai pH di luar rentang tersebut dianggap tidak memenuhi syarat dan program akan menampilkan bahwa air tidak layak minum.

2. Soal 2

Source Code

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var jenis_kendaraan string
    var durasi_parkir, total_biaya int

    fmt.Println("Masukkan jenis kendaraan (Motor/Mobil/Truk) :")
    fmt.Scan(&jenis_kendaraan)

    jenis_kendaraan = strings.ToLower(jenis_kendaraan)

    fmt.Println("Durasi parkir (jam) : ")
    fmt.Scan(&durasi_parkir)

    if durasi_parkir < 0 {
        fmt.Println("Durasi parkir tidak valid")
        return
    }
}
```

```
if durasi_parkir < 1{  
    durasi_parkir = 1  
}  
  
switch {  
  
case "motor" == jenis_kendaraan:  
  
    total_biaya = 2000 * durasi_parkir  
  
    fmt.Printf("Total biaya parkir motor anda adalah:  
Rp.%d\n",total_biaya)  
  
case "mobil" == jenis_kendaraan:  
  
    total_biaya = 5000 * durasi_parkir  
  
    fmt.Printf("Total biaya parkir mobil anda adalah:  
Rp.%d\n",total_biaya)  
  
case "truk" == jenis_kendaraan:  
  
    total_biaya = 8000 * durasi_parkir  
  
    fmt.Printf("Total biaya parkir truk anda adalah:  
Rp.%d\n",total_biaya)  
  
default:  
  
    fmt.Println("Jenis kendaraan tidak dikenali")  
}  
}
```

Screenshot program

The screenshot shows a code editor interface with a Go file named 'main.go' open. The code defines a function 'main()' that prompts the user for the type of vehicle ('Motor/Mobil/Truk') and the duration of parking ('jam'). It then calculates the total fee based on the selected vehicle type and duration. The code uses standard input/output operations and a switch statement to handle different vehicle types. The output tab shows the execution of the program, where it asks for the vehicle type and duration, and then prints the total fee.

```
func main() {
    var jenis_kendaraan string
    var durasi_parkir, total_biaya int
    fmt.Println("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
    fmt.Scan(&jenis_kendaraan)
    jenis_kendaraan = strings.ToLower(jenis_kendaraan)
    fmt.Println("Durasi parkir (jam): ")
    fmt.Scan(&durasi_parkir)

    if durasi_parkir < 0 {
        fmt.Println("Durasi parkir tidak valid")
        return
    }

    if durasi_parkir < 1 {
        durasi_parkir = 1
    }

    switch {
    case "motor" == jenis_kendaraan:
        total_biaya = 2000 * durasi_parkir
        fmt.Printf("Total biaya parkir motor anda adalah: Rp.%d\n", total_biaya)
    case "mobil" == jenis_kendaraan:
        total_biaya = 5000 * durasi_parkir
        fmt.Printf("Total biaya parkir mobil anda adalah: Rp.%d\n", total_biaya)
    case "truk" == jenis_kendaraan:
        total_biaya = 8000 * durasi_parkir
        fmt.Printf("Total biaya parkir truk anda adalah: Rp.%d\n", total_biaya)
    default:
        fmt.Println("Jenis kendaraan tidak dikenali")
    }
}
```

Deskripsi program

Program di atas meminta pengguna untuk memasukkan jenis kendaraan serta durasi parkir, kemudian program akan mengonversi input jenis kendaraan menjadi huruf kecil untuk memastikan konsistensi dalam proses pengecekan. Program selanjutnya akan memvalidasi durasi parkir dengan memastikan bahwa nilainya tidak negatif; jika nilai tersebut kurang dari nol, program langsung menghentikan eksekusi dan menampilkan pesan bahwa input tidak valid. Apabila durasi parkir kurang dari satu jam, program menetapkan nilainya menjadi satu sebagai durasi minimum. Setelah melewati tahap validasi, program menggunakan struktur *switch* berbasis kondisi untuk menentukan tarif parkir berdasarkan jenis kendaraan. Untuk kendaraan motor, biaya dihitung sebesar 2.000 dikali durasi parkir, sedangkan mobil dikenakan tarif 5.000 per jam, dan truk dikenakan 8.000 per jam. Jika jenis kendaraan tidak sesuai dengan kategori yang tersedia, program akan menampilkan pesan bahwa jenis kendaraan tidak dikenali.

3. Soal 3

Source Code

```
package main

import "fmt"

func main() {
    var bilangan int

    fmt.Println("Masukkan bilangan: ")
    fmt.Scan(&bilangan)

    switch{
        case bilangan == 0:
            fmt.Println("Input tidak boleh nol")
        case bilangan%10 == 0:
            hasil := bilangan / 10
            fmt.Println("Kategori: Bilangan Kelipatan 10")
            fmt.Sprintf("Hasil pembagian antara %d dengan 10
adalah %d", bilangan, hasil)
        case bilangan%5 == 0:
            hasil := bilangan * bilangan
            fmt.Println("Kategori: Bilangan Kelipatan 5")
            fmt.Sprintf("Hasil kuadrat dari %d adalah %d",
bilangan, hasil)
    }
}
```

```
case bilangan%2 == 0:

    hasil := bilangan * (bilangan + 1)

    fmt.Println("Kategori: Bilangan Genap")

    fmt.Printf("Hasil perkalian antara %d dengan bilangan
berikutnya %d adalah %d", bilangan, bilangan+1, hasil)

case bilangan%2 != 0:

    hasil := bilangan + (bilangan + 1)

    fmt.Println("Kategori: Bilangan Ganjil")

    fmt.Printf("Hasil penjumlahan antara %d dengan
bilangan berikutnya %d adalah %d", bilangan, bilangan+1,
hasil)

}

}
```

Screenshot program

The screenshot shows a Go development environment with the following details:

- File Explorer:** Shows a project structure under "PRAKTIKUM-11" with files like "main.go", "pola_aritmatika.go", and "pola_aritmatika_test.go".
- Code Editor:** Displays the content of "main.go" which contains a function "main()" that reads an integer from the user and prints its category based on various conditions.
- Terminal:** Shows the command "go run main.go" and its output:

```
Input bilangan: 25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 adalah 625
```
- Status Bar:** Includes file paths ("C:\Users\coolyear\alpro1\praktikum-11\pola_aritmatika\main.go"), line numbers ("Ln 32, Col 2"), tab size ("Tab Size 4"), encoding ("UTF-8"), and other system information.

Deskripsi program

Program di atas dibuat untuk membaca sebuah bilangan dari inputan User lalu memprosesnya menggunakan struktur *switch* berbasis kondisi untuk menentukan kategori bilangan beserta operasi yang harus dilakukan. Jika bilangan yang dimasukkan bernilai 0, program langsung menampilkan pesan bahwa input tidak boleh nol. Apabila bilangan merupakan kelipatan 10, program membagi bilangan tersebut dengan 10 dan menampilkan hasilnya. Jika bilangan termasuk kelipatan 5 tetapi bukan kelipatan 10, program menghitung nilai kuadrat dari bilangan tersebut. Untuk bilangan genap yang tidak memenuhi kondisi sebelumnya, program mengalikan bilangan dengan bilangan setelahnya. Sementara itu, jika bilangan merupakan bilangan ganjil, program menjumlahkan bilangan tersebut dengan bilangan berikutnya.