

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 11
SWITCH-CASE



Disusun oleh:

Ichsan Maulana Muhammad

109082500093

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 Go development environment with the following details:

- File Explorer:** Shows various Go files in the project structure, including `MODUL 3`, `MODUL 4`, `MODUL 5 DAN 6`, `MODUL 7`, `MODUL 9`, `MODUL 10`, `GUIDED1.go`, `GUIDED2.go`, `GUIDED3.go`, `TUGAS1.go`, `TUGAS3.go`, `tugas2.go`, and `ALPRO TEORI.go`.
- Code Editor:** Displays the content of `guided 11.go`. The code defines a package `main` with a `func main()` function. It reads a 24-hour time input from the user using `fmt.Scan(&jam24)` and then uses a switch statement to determine if it's AM or PM. Finally, it prints the converted time using `fmt.Println(jam12, label)`.
- Terminal:** Shows the command `go run "c:\Users\ASUS\yy\MODUL 11\guided 11.go"` being run three times, resulting in outputs "1 PM", "12 AM", and "12 PM".
- Output:** A preview pane showing the converted time: "NAMA: ICHSAN MAULANA M", "NIM: 109082500093", and "KELAS: IF-13-04".

Deskripsi program

1. Program diawali dengan package `main`, import `"fmt"`, serta fungsi `main()` sebagai fungsi utama untuk menjalankan program.
2. Mendeklarasikan variabel `jam24` dengan tipe data integer untuk menampung input jam dalam format 24 jam.
3. Program meminta pengguna memasukkan sebuah bilangan bulat dengan rentang 0 sampai 23 sebagai jam dalam format 24 jam, lalu membaca input tersebut menggunakan `fmt.Scan(&jam24)`.
4. Program melakukan pengecekan nilai `jam24` menggunakan struktur seleksi `if` untuk menentukan hasil konversi:
 - Jika nilai `jam24` adalah 0, maka hasil konversi menjadi 12 AM.
 - Jika nilai `jam24` adalah 12, maka hasil konversi menjadi 12 PM.
 - Jika nilai `jam24` kurang dari 12 (1–11), maka hasil konversi menjadi jam yang sama diikuti dengan AM.
 - Jika nilai `jam24` lebih dari 12 (13–23), maka hasil konversi menjadi nilai `jam24 - 12` diikuti dengan PM.
5. Program menyimpan hasil konversi tersebut ke dalam dua nilai, yaitu angka jam dan teks penunjuk waktu (AM/PM).
6. Program menampilkan hasil konversi waktu ke format 12 jam menggunakan `fmt.Println()`, misalnya seperti: 1 PM, 12 AM.
7. Output berupa teks yang menyatakan jam hasil konversi dari format 24 jam menjadi format 12 jam sesuai nilai yang dimasukkan pengguna

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

Screenshot program

The screenshot shows a code editor interface with multiple tabs open. The active tab contains the following Go 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.")
    }
}
```

The code defines a package named 'main' with a function 'main()' that reads a plant name from standard input. It then checks if the name is 'nepenthes' or 'drosera', in which case it prints 'Termasuk Tanaman Karnivora.' and 'Asli Indonesia.'. If the name is 'venus' or 'sarracenia', it prints 'Termasuk Tanaman Karnivora.' and 'Tidak Asli Indonesia.'. For any other name, it prints 'Tidak termasuk Tanaman Karnivora.'

Below the code editor, a terminal window shows the output of running the program:

```
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 11\guided 22.go"
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 11\guided 22.go"
venus
Termasuk Tanaman Karnivora.
Tidak Asli Indonesia.
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 11\guided 22.go"
karedok
Tidak termasuk Tanaman Karnivora.
PS C:\Users\ASUS\yy>
```

Deskripsi program

1. Program diawali dengan package main, import "fmt", serta fungsi main() sebagai fungsi utama untuk menjalankan program.
2. Mendeklarasikan variabel tanaman dengan tipe data string untuk menampung input nama tanaman dari pengguna.
3. Program meminta pengguna memasukkan nama sebuah tanaman, kemudian membaca input tersebut menggunakan fmt.Scan(&tanaman) atau fmt.Scanln(&tanaman).
4. Program melakukan pengecekan nama tanaman menggunakan struktur seleksi if untuk menentukan apakah tanaman tersebut termasuk tanaman karnivora atau tidak:
 - Jika nama tanaman adalah "nepenthes" atau "venus", maka tanaman termasuk tanaman karnivora.
 - Selain nama tersebut, maka tanaman tidak termasuk tanaman karnivora.
5. Jika tanaman termasuk tanaman karnivora, maka program melakukan pengecekan lebih lanjut untuk menentukan apakah tanaman tersebut asli Indonesia atau bukan asli Indonesia:
 - Jika nama tanaman adalah "nepenthes", maka statusnya Asli Indonesia.
 - Jika nama tanaman adalah "venus", maka statusnya Bukan Asli Indonesia.
6. Program menampilkan hasil berdasarkan pengecekan, Jika tanaman termasuk tanaman karnivora, program menampilkan dua baris teks:
 - "Termasuk Tanaman Karnivora"

- diikuti dengan "Asli Indonesia" atau "Bukan Asli Indonesia" sesuai data tanaman.
 - Jika tanaman tidak termasuk tanaman karnivora, program menampilkan pesan: "Tidak termasuk Tanaman Karnivora".
7. Output berupa informasi status tanaman sesuai input pengguna, yaitu apakah tanaman tersebut termasuk tanaman karnivora atau tidak, dan jika termasuk, apakah tanaman tersebut asli Indonesia atau bukan.

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)
}

```

Screenshot program

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

- File Explorer:** Shows various Go files in the project structure, including `guided 11.go`, `guided 22.go`, `guided 33.go`, and `GUIDED1.go`.
- Code Editor:** The file `guided 33.go` contains the Go code provided in the question. It includes a switch statement for vehicle types and duration, and a default case for invalid input.
- Terminal:** The terminal window shows the execution of the program. It prompts for vehicle type and duration, then prints the calculated tariff. The output is as follows:


```

NAMA: ICHSAN MAULANA M
NIM: 109082500093
KELAS: IF-13-04

Masukkan jenis kendaraan (Motor/Mobil/truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS C:\Users\ASUS\yy> go run "C:\Users\ASUS\yy\MODUL 11\guided 33.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
PS C:\Users\ASUS\yy>
```

Deskripsi program

1. Program diawali dengan package main, import "fmt", serta fungsi main() sebagai fungsi utama untuk menjalankan program.
2. Mendeklarasikan variabel jenis bertipe string untuk menyimpan jenis kendaraan, dan variabel durasi bertipe integer untuk menyimpan durasi parkir dalam jam.
3. Program meminta pengguna memasukkan jenis kendaraan (Motor/Mobil/Truk), kemudian membaca input tersebut menggunakan fmt.Scanln(&jenis).
4. Program meminta pengguna memasukkan durasi parkir dalam jam, kemudian membaca input tersebut menggunakan fmt.Scan(&durasi).
5. Program melakukan pengecekan nilai jenis menggunakan struktur seleksi if untuk menentukan tarif parkir berdasarkan tabel ketentuan tarif:
 - Jika kendaraan adalah Motor, tarif dikenakan sebesar Rp 7.000 untuk durasi 1–2 jam dan Rp 9.000 jika durasi lebih dari 2 jam.
 - Jika kendaraan adalah Mobil, tarif dikenakan sebesar Rp 15.000 untuk durasi 1–2 jam dan Rp 20.000 jika durasi lebih dari 2 jam.
 - Jika kendaraan adalah Truk, tarif dikenakan sebesar Rp 25.000 untuk durasi 1–2 jam dan Rp 35.000 jika durasi lebih dari 2 jam.
6. Jika input jenis kendaraan bukan Motor, Mobil, atau Truk, maka program tidak menghitung tarif parkir dan menampilkan pesan "Jenis kendaraan atau durasi parkir tidak valid".
7. Jika jenis kendaraan valid, program menentukan tarif parkir berdasarkan durasi parkir dengan pengecekan:
 - Jika durasi berada dalam rentang 1 sampai 2 jam, maka tarif digunakan sesuai ketentuan tarif awal.
 - Jika durasi lebih dari 2 jam, maka tarif diubah sesuai ketentuan tarif durasi lebih dari 2 jam.
8. Program menampilkan hasil akhir berupa string "Tarif Parkir: Rp ..." sesuai tarif yang telah dihitung.
9. Output berupa informasi tarif parkir berdasarkan jenis kendaraan dan durasi parkir, atau pesan kesalahan jika jenis kendaraan yang dimasukkan tidak valid.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var ph float64

    fmt.Scan(&ph)

    switch {
        case ph >= 0 && ph <= 14:
            switch {
                case ph >= 6.5 && ph <= 8.6:
                    fmt.Println("Air layak minum")

                case ph < 6.5 || ph > 8.6:
                    fmt.Println("Air tidak layak minum")
            }
        default:
            fmt.Println("Input tidak valid. Nilai pH harus antara
0 dan 14.")
    }
}
```

Screenshot program

The screenshot shows a code editor interface with multiple tabs open. The active tab contains a Go program named `tugas 11.go`. The code defines a package `main` with a `main()` function. This function reads a float64 value from the user using `fmt.Scan(&ph)`. It then checks if the pH value is between 0 and 14. If it is, it prints "Air layak minum". If it's less than 0 or greater than 14, it prints "Input tidak valid. Nilai pH harus antara 0 dan 14.". The code editor also shows a terminal window where the program is run and its output is displayed.

```
package main

import "fmt"

func main() {
    var ph float64
    fmt.Scan(&ph)

    switch {
    case ph >= 0 && ph <= 14:
        switch {
            case ph >= 6.5 && ph <= 8.5:
                fmt.Println("Air layak minum")
            case ph < 6.5 || ph > 8.5:
                fmt.Println("Air tidak layak minum")
        }
    default:
        fmt.Println("Input tidak valid. Nilai pH harus antara 0 dan 14.")
    }
}
```

```
NAMA: ICHSAN MAULANA M
NIM: 109082500093
KELAS: IF-13-04
```

Deskripsi program

1. Program diawali dengan `package main`, `import "fmt"`, serta fungsi `main()` sebagai fungsi utama untuk menjalankan program.
2. Program mendeklarasikan variabel `ph` bertipe `float64` untuk menyimpan nilai pH yang diinput pengguna.
3. Program meminta pengguna memasukkan nilai pH, kemudian membaca input tersebut menggunakan `fmt.Scan(&ph)`.
4. Program melakukan pengecekan validitas nilai pH menggunakan struktur seleksi `if`:
 - Jika $\text{ph} < 0$ atau $\text{ph} > 14$, maka program menampilkan pesan "Input tidak valid, rentang pH 0 - 14".
5. Jika nilai pH valid, program melanjutkan pengecekan kategori pH:
 - Jika $\text{ph} \geq 6.5$ dan $\text{ph} \leq 8.5$, maka program menampilkan "Air layak minum".
 - Jika $\text{ph} < 6.5$ atau $\text{ph} > 8.5$, program menampilkan "Air tidak layak minum".
6. Program menampilkan output sesuai kondisi yang terpenuhi berdasarkan nilai pH yang dimasukkan pengguna.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var jenis string
    var durasi, tarifperjam int

    fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk) : ")
    fmt.Scan(&jenis)

    fmt.Print("Masukkan durasi parkir (dalam 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 the Visual Studio Code (VS Code) interface. The left side features the Explorer sidebar with a tree view of files and folders. The main area is the Code Editor, which contains the Go code for calculating parking fees. Below the editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing command-line input and output. The right side of the interface includes a floating terminal window titled 'Terminal' displaying student information: NAMA: ICHSAN MAULANA M, NIM: 109082500093, KELAS: IF-13-04. At the bottom, there are status bars for the current file (Ln 3, Col 16), character count (56 character), and other settings.

```

package main

import "fmt"

func main() {
    var jenis string
    var durasi, tarifperjam int

    fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk): ")
    fmt.Scan(&jenis)
    fmt.Print("Masukkan durasi parkir (dalam jam): ")
    fmt.Scan(&durasi)

    if durasi < 1 {
        durasi = 1
    }

    switch jenis {
    case "motor":
        tarifperjam = 2000
    case "mobil":
        tarifperjam = 5000
    case "truk":
        tarifperjam = 8000
    default:
        break
    }
}

PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL_11\tugas_22.go"
Masukkan jenis kendaraan (motor/mobil/truk): motor
Masukkan durasi parkir (dalam jam): 3
Total biaya parkir: Rp 6000
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL_11\tugas_22.go"
Masukkan jenis kendaraan (motor/mobil/truk): mobil
Masukkan durasi parkir (dalam jam): 1
Total biaya parkir: Rp 5000
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL_11\tugas_22.go"
Masukkan jenis kendaraan (motor/mobil/truk): truk
Masukkan durasi parkir (dalam jam): 5
Total biaya parkir: Rp 40000
PS C:\Users\ASUS\yy>

```

Deskripsi program

1. Program diawali dengan *package main, import "fmt"*, serta fungsi main() sebagai fungsi utama untuk menjalankan program.
2. Program mendeklarasikan variabel jenis bertipe string untuk menyimpan jenis kendaraan, serta variabel durasi bertipe int untuk menyimpan durasi parkir dalam jam.
3. Program meminta pengguna memasukkan jenis kendaraan (motor/mobil/truk), kemudian membaca input menggunakan fmt.Scan(&jenis).
4. Program meminta pengguna memasukkan durasi parkir, kemudian membaca nilai tersebut menggunakan fmt.Scan(&durasi).
5. Jika durasi kurang dari 1, program menetapkan durasi menjadi 1 sesuai ketentuan minimum.
6. Program menentukan tarif per jam berdasarkan jenis kendaraan menggunakan struktur seleksi if:
 - Jika kendaraan motor, tarif per jam adalah 2000.
 - Jika kendaraan mobil, tarif per jam adalah 5000.
 - Jika kendaraan truk, tarif per jam adalah 8000.
 - Jika input jenis kendaraan tidak sesuai, program menampilkan "Jenis kendaraan tidak valid" dan tidak menghitung tarif.
7. Jika jenis kendaraan valid, program menghitung total biaya parkir dengan mengalikan tarif per jam dengan durasi parkir.
8. Program menampilkan hasil akhir berupa "Total biaya parkir: Rp ..." berdasarkan perhitungan tarif.
9. Output berupa total biaya parkir atau pesan kesalahan jika jenis kendaraan tidak valid.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)

    switch {
    case n%2 != 0:
```

```

        fmt.Printf("- Bilangan Ganjil (hasil %d + %d =
%d)\n", n, n+1, n+(n+1))

    case n%2 == 0:

        fmt.Printf("- Bilangan Genap (hasil %d * %d = %d)\n",
n, n+1, n*(n+1))

    }

    switch {

    case n%5 == 0:

        fmt.Printf("- Bilangan Kelipatan 5 (hasil %d^2 =
%d)\n", n, n*n)

    }

    switch {

    case n%10 == 0:

        fmt.Printf("- Bilangan Kelipatan 10 (hasil %d / 10 =
%d)\n", n, n/10)

    }

}

```

Screenshot program

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

- EXPLORER**: Shows project structure with modules: MODUL 6, MODUL 7, MODUL 9, MODUL 10, MODUL 11, and several guided files.
- CODE EDITOR**: Displays the source code for `tugas 33.go`. The code handles user input for a number and prints results for odd numbers, even numbers, multiples of 5, and multiples of 10.
- TERMINAL**: Shows command-line interactions. It runs the program and provides output for various test cases, including odd and even numbers, and multiples of 5 and 10.
- OUTPUT**: Shows the program's output, which includes the calculated results for each case based on the user input.

Deskripsi program

1. Program diawali dengan `package main`, `import "fmt"`, dan fungsi `main()` sebagai fungsi utama.

2. Program mendeklarasikan variabel n bertipe int untuk menyimpan bilangan yang diinput pengguna.
3. Program meminta pengguna memasukkan satu bilangan bulat, kemudian membaca input menggunakan `fmt.Scan(&n)`.
4. Program mengelompokkan bilangan berdasarkan ketentuan kategori:
 - Bilangan Ganjil: program menghitung hasil penjumlahan $n + (n+1)$.
 - Bilangan Genap: program menghitung hasil perkalian $n \times (n+1)$.
 - Bilangan Kelipatan 5: program menghitung hasil kuadrat $n \times n$.
 - Bilangan Kelipatan 10: program menghitung hasil pembagian $n / 10$.
5. Penentuan kategori dilakukan menggunakan seleksi bertingkat if:
 - Jika $n \% 10 == 0$, termasuk Kelipatan 10.
 - Else jika $n \% 5 == 0$, termasuk Kelipatan 5.
 - Else jika $n \% 2 == 0$, termasuk Bilangan Genap.
 - Else termasuk Bilangan Ganjil.
6. Program menampilkan hasil dalam format teks seperti contoh, misalnya:
 - Kategori: "Bilangan Ganjil" diikuti dengan hasil perhitungan, Hasil penjumlahan dengan bilangan berikutnya $n + (n+1) = hasil$.
7. Output disesuaikan dengan kategori dan perhitungan berdasarkan bilangan yang diinput pengguna.