

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11

SWITCH-CASE



**Telkom
University
PURWOKERTO**

Disusun oleh:

BAGUS IRSYAD KAMAL

109082500215

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 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 components:

- Explorer:** Shows three files: 01.go, 02.go, and 03.go. 01.go is the active file.
- Code Editor:** Displays the content of 01.go, which contains a Go program that converts a 24-hour time input into a 12-hour format with AM/PM labels.
- Terminal:** Shows the command `go run contoh1/01.go` being run three times, resulting in outputs 13, 12 AM, and 12 PM respectively.
- powershell:** A separate window shows the results of the program execution: NAMA : BAGUS IRSYAD KAMAL and NIM : 109082500215.

Deskripsi program

Program ini digunakan untuk mengubah jam dalam format 24 jam (0–23) menjadi format 12 jam dengan label AM atau PM. Program menerima satu input, yaitu sebuah bilangan bulat yang mewakili jam dalam format 24 jam. Setelah membaca input, program menggunakan struktur switch untuk menentukan jam dalam format 12 jam. Ada beberapa kondisi yang dicek:

1. Jika jam24 bernilai 0, maka jam12 menjadi 12 dan label menjadi "AM". Ini karena jam 0 pada format 24 jam adalah pukul 12 malam.
2. Jika jam24 kurang dari 12, maka jam12 sama dengan jam24 dan labelnya "AM". Ini menunjukkan waktu pagi.
3. Jika jam24 sama dengan 12, maka jam12 tetap 12 tetapi label menjadi "PM". Ini menunjukkan pukul 12 siang.
4. Jika jam24 lebih dari 12, maka jam12 dihitung dengan $\text{jam24} - 12$ dan labelnya "PM". Ini menunjukkan waktu setelah siang hingga malam.

Setelah menentukan nilai jam12 dan label, program mencetak hasil konversi tersebut.

Tujuan program ini adalah membantu mengubah waktu dalam format 24 jam menjadi format 12 jam yang lebih umum digunakan dalam kehidupan sehari-hari.

Contoh

Input:

13

Jam 13 pada format 24 jam dikonversi menjadi 1 PM.

Output:

1 PM

Input:

0

Jam 0 pada format 24 jam dikonversi menjadi 12 AM.

Output:

12 AM

Input:

12

Jam 12 pada format 24 jam dikonversi menjadi 12 PM.

Output:

12 PM

Input:

7

Jam 7 pada format 24 jam dikonversi menjadi 7 AM.

Output:

7 AM

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.")

}

}

fmt.Println("vokal")

}else if huruf && !(vKecil || vBesar){

    fmt.Println("konsonan")

}else{

    fmt.Println("bukan huruf")

}

}

```

Screenshot program

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

- File Explorer:** Shows files 01.go, 02.go, and 03.go under a folder named "SWITCHCASE".
- Code Editor:** The active file is 02.go, containing a Go program that reads a string from standard input and prints it to standard output based on a switch statement.
- Terminal:** The terminal window shows the command `go run contoh2/02.go` being run twice. The first run prints "Terasik Tanaman Karnivora." for the input "nepenthes", and the second run prints "Asli Indonesia." for the input "venus".
- Output Window:** A separate window titled "powershell" shows the output of the program: "NAMA : BAGUS IRSYAD KAMAL" and "NIM : 109082500215".

Deskripsi program

Program ini digunakan untuk menentukan apakah sebuah nama tanaman termasuk tanaman karnivora atau bukan, serta apakah tanaman tersebut berasal dari Indonesia atau bukan. Program menerima satu input dari pengguna berupa sebuah teks (string) yang berisi nama tanaman.

Setelah membaca input, program menggunakan struktur switch untuk membandingkan nama tanaman tersebut dengan beberapa kategori:

1. Jika nama tanaman adalah "nepenthes" atau "drosera", maka program menampilkan bahwa tanaman tersebut termasuk tanaman karnivora dan berasal dari Indonesia.
2. Jika nama tanaman adalah "venus" atau "sarracenia", maka program menampilkan bahwa tanaman tersebut termasuk tanaman karnivora tetapi bukan asli Indonesia.
3. Jika nama tanaman tidak cocok dengan semua pilihan di atas, program menampilkan bahwa tanaman tersebut tidak termasuk tanaman karnivora.

Tujuan program ini adalah membantu mengidentifikasi jenis tanaman berdasarkan nama yang dimasukkan oleh pengguna.

Contoh

Input:

nepenthes

Output:

Termasuk Tanaman Karnivora.

Asli Indonesia.

Input:

venus

Output:

Termasuk Tanaman Karnivora.

Tidak Asli Indonesia.

Input:

karedok

Output:

Tidak termasuk Tanaman Karnivora.

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:

- Code Editor:** An IDE window displaying a file named `03.go` containing a Go program that uses a switch statement to calculate parking fees based on vehicle type and duration.
- Terminal:** A terminal window showing the command `go run contoh3/03.go` being run, followed by user input for vehicle type ("Motor") and duration (2 hours), and the resulting fee (Rp. 7000).
- Output Window:** A separate window showing the same terminal session and the calculated fee.

Deskripsi program

Program ini digunakan untuk menghitung tarif parkir berdasarkan dua input, yaitu:

1. Jenis kendaraan (Motor, Mobil, atau Truk)
2. Durasi parkir dalam jam

Setelah membaca kedua input tersebut, program menggunakan struktur switch untuk menentukan tarif parkir sesuai aturan berikut:

1. Jika kendaraan adalah "Motor" dan durasi parkir 1–2 jam, tarifnya adalah 7000. Jika durasi lebih dari 2 jam, tarifnya menjadi 9000.
2. Jika kendaraan adalah "Mobil" dan durasi parkir 1–2 jam, tarifnya adalah 15000. Jika durasi lebih dari 2 jam, tarifnya menjadi 20000.
3. Jika kendaraan adalah "Truk" dan durasi parkir 1–2 jam, tarifnya adalah 25000. Jika durasi lebih dari 2 jam, tarifnya menjadi 35000.

Jika pengguna memasukkan jenis kendaraan yang tidak termasuk ketiga pilihan di atas, atau durasi parkir kurang dari 1 jam, maka program menampilkan pesan: "Jenis kendaraan atau durasi parkir tidak valid"

Tujuan program ini adalah membantu menentukan tarif parkir dengan tepat berdasarkan jenis kendaraan dan durasi waktu parkir.

Contoh

Input:

Motor

2

Output:

Tarif Parkir: Rp 7000

Input:

Mobil

4

Output:

Tarif Parkir: Rp 20000

Input:

Truk

1

Output:

Tarif Parkir: Rp 25000

Input:

Sepeda

2

Output:

Jenis kendaraan atau durasi parkir tidak valid

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var ph float64

    fmt.Println("Masukkan kadar pH air: ")
    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 a Go development environment with the following components:

- Explorer:** Shows a project structure under "SWITCHCASE" with files: 01.go, contoh1, 02.go, contoh2, 03.go, tugas01, 01.go, tugas02, 02.go.
- Code Editor:** Displays the content of 01.go, which contains a Go program for pH level validation. The code includes imports for fmt and math, a main function that reads a float64 pH value from standard input, and a switch statement to determine if the pH is valid (between 0 and 14), suitable for drinking (between 6.5 and 8.6), or not suitable (otherwise).
- Terminal:** Shows command-line interactions:
 - PS C:\Users\HP VICTUS\Desktop\switchcase> go run tugas01/01.go
Masukkan kadar pH air: 8.6
Air layak minum
 - PS C:\Users\HP VICTUS\Desktop\switchcase> go run tugas01/01.go
Masukkan kadar pH air: 9
Air tidak layak minum
 - PS C:\Users\HP VICTUS\Desktop\switchcase> go run tugas01/01.go
Masukkan kadar pH air: 14
Air tidak layak minum
- Powershell:** Shows user information:
NAMA : BAGUS IRSYAD KAMAL
NIM : 109082500215

Deskripsi program

Program ini digunakan untuk menentukan apakah air dengan kadar pH tertentu layak untuk diminum atau tidak. Program menerima satu input berupa bilangan float yang mewakili kadar pH air.

Setelah membaca input, program melakukan beberapa pengecekan menggunakan struktur if:

1. Jika nilai pH kurang dari 0 atau lebih dari 14, program menampilkan pesan bahwa nilai pH tidak valid. Hal ini karena pH hanya berada pada rentang 0 sampai 14.
2. Jika nilai pH berada pada rentang 6.5 sampai 8.6, program menampilkan "Air layak minum". Ini merupakan kisaran aman untuk air minum.
3. Jika nilai pH berada di dalam rentang 0 sampai 14 tetapi berada di luar kisaran 6.5 sampai 8.6, program menampilkan "Air tidak layak minum".

Tujuan program ini adalah membantu menentukan kelayakan air minum berdasarkan kadar pH yang diinput pengguna.

Contoh

Input:

8.6

Output:

Air layak minum

Input:

9

Output:

Air tidak layak minum

Input:

16

Output:

Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi int
    var tarifPerJam int

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

    fmt.Scan(&kendaraan)

    fmt.Print("Masukkan durasi parkir (jam) : ")
    fmt.Scan(&durasi)

    if durasi < 1 {
        durasi = 1
    }

    switch kendaraan {
        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("Rp", total)

}

```

Screenshot program

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

- EXPLORER:** Shows files: 01.go, contoh1, contoh2, contoh3, 02.go, tugas01, tugas02.
- CODE EDITOR:** Displays the Go code for a switch-case problem. The code includes a main function that prints for input, reads a duration, and calculates a total based on the input. It handles invalid vehicle types by returning an error message.
- TERMINAL:** Shows command-line interactions:
 - PS C:\Users\HP VICTUS\Desktop\switchcase> go run tugas02/02.go
 - Masukkan jenis kendaraan (motor/mobil/truk): motor
 - Masukkan durasi parkir (jam): 3
 - Rp 6000
 - PS C:\Users\HP VICTUS\Desktop\switchcase> go run tugas02/02.go
 - Masukkan jenis kendaraan (motor/mobil/truk): mobil
 - Masukkan durasi parkir (jam): 1
 - Rp 5000
 - PS C:\Users\HP VICTUS\Desktop\switchcase> go run tugas02/02.go
 - Masukkan jenis kendaraan (motor/mobil/truk): truk
- BROWSER:** A small window showing a login form with fields for NAMA and NIM.

Deskripsi program

Program ini digunakan untuk menghitung total biaya parkir berdasarkan dua input dari pengguna, yaitu:

1. Jenis kendaraan (motor, mobil, atau truk)
2. Durasi parkir dalam jam

Setiap jenis kendaraan memiliki tarif parkir per jam yang berbeda:

- Motor: Rp 2.000 per jam
- Mobil: Rp 5.000 per jam
- Truk: Rp 8.000 per jam

Setelah membaca input, program memeriksa durasi parkir. Jika durasi parkir kurang dari 1 jam, maka durasi otomatis dianggap 1 jam.

Program kemudian menggunakan struktur switch untuk menentukan tarif per jam:

- Jika pengguna memasukkan "motor", tarif per jam = 2000
- Jika pengguna memasukkan "mobil", tarif per jam = 5000
- Jika pengguna memasukkan "truk", tarif per jam = 8000
- Jika input bukan salah satu dari ketiga jenis tersebut, program menampilkan pesan "Jenis kendaraan tidak valid."

Setelah tarif per jam ditentukan, program mengalikan tarif tersebut dengan durasi untuk mendapatkan total biaya parkir. Total biaya parkir kemudian ditampilkan kepada pengguna.

Tujuan program ini adalah membantu menghitung total tarif parkir secara otomatis berdasarkan jenis kendaraan dan lamanya waktu parkir.

Contoh

Input:

motor

3

Output:

Rp 6000

Input:

mobil

1

Output:

Rp 5000

Input:

truk

5

Output:

Rp 40000

Tugas 3

Source code

```
package main

import "fmt"

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

    switch {
    case n%2 != 0:
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n",
                    n, n+1, n+(n+1))

    case n%2 == 0:
        if n%10 == 0 {
            fmt.Println("Kategori: Bilangan Kelipatan
10")
            fmt.Printf("Hasil pembagian antara %d / 10
= %d\n", n, n/10)
        } else if n%5 == 0 {
            fmt.Println("Kategori: Bilangan Kelipatan
5")
            fmt.Printf("Hasil kuadrat dari %d ^2 =
%d\n", n, n*n)
        } else {
            fmt.Println("Kategori: Bilangan Genap")
            fmt.Printf("Hasil perkalian dengan bilangan
berikutnya %d * %d = %d\n",
                        n, n+1, n*(n+1))
        }
    }

    default:
```

```

        fmt.Println("Input tidak valid")
    }
}

```

Screenshot program

The screenshot shows a Go development environment in VS Code. The Explorer sidebar on the left lists several Go files under a 'SWITCHCASE' folder, including '01.go contoh1', '02.go contoh2', '03.go contoh3', '01.go tugas01', '02.go tugas02', and '03.go tugas03'. The main editor area displays the '03.go' file, which contains a Go program that reads an integer from the user and classifies it based on its divisibility by 10 or 5. The terminal below shows the execution of the program and its output for two different inputs: 5 and 25.

```

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

    switch {
    case n%2 != 0:
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n",
            n, n+1, n+(n+1))

    case n%2 == 0:
        if n%10 == 0 {
            fmt.Println("Kategori: Bilangan Kelipatan 10")
            fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, n/10)
        } else if n%5 == 0 {
            fmt.Println("Kategori: Bilangan Kelipatan 5")
            fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", n, n*n)
        } else {
            fmt.Println("Kategori: Bilangan Genap")
            fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n",
                n, n+1, n*(n+1))
        }
    default:
        fmt.Println("Input tidak valid")
    }
}

Hasil pembagian antara 20 / 10 = 2
PS C:\Users\VP VICTUS\Desktop\switchcase> go run tugas03/03.go
Masukkan angka: 5
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS C:\Users\VP VICTUS\Desktop\switchcase> go run tugas03/03.go
Masukkan angka: 25
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 25 + 26 = 51
PS C:\Users\VP VICTUS\Desktop\switchcase>

```

Deskripsi program

Program ini digunakan untuk mengidentifikasi kategori sebuah bilangan bulat, kemudian melakukan operasi matematika tertentu sesuai kategori tersebut. Program membaca satu input berupa bilangan bulat dari pengguna.

Setelah membaca input, program menentukan kategori bilangan menggunakan struktur switch:

1. Bilangan Kelipatan 10

Jika bilangan habis dibagi 10, program menampilkan kategori "Bilangan Kelipatan 10", lalu menghitung hasil pembagian antara bilangan dengan angka 10.

2. Bilangan Kelipatan 5

Jika bilangan tidak termasuk kelipatan 10 tetapi habis dibagi 5, program menampilkan kategori "Bilangan Kelipatan 5", lalu menghitung kuadrat dari bilangan tersebut.

3. Bilangan Genap

Jika bilangan tidak termasuk kelipatan 10 atau 5 tetapi merupakan bilangan genap, program menampilkan kategori "Bilangan Genap", lalu menghitung perkalian antara bilangan dengan bilangan berikutnya.

4. Bilangan Ganjil

Jika bilangan tidak masuk kategori apa pun di atas, berarti bilangan tersebut ganjil.

Program menampilkan kategori "Bilangan Ganjil", lalu menghitung penjumlahan antara bilangan dengan bilangan berikutnya.

Tujuan program ini adalah untuk mengelompokkan bilangan ke dalam kategori tertentu, kemudian menampilkan hasil operasi matematika sesuai aturan kategori tersebut.

Contoh

Input:

5

Output:

Kategori: Bilangan Ganjil

Hasil penjumlahan dengan bilangan berikutnya $5 + 6 = 11$

Input:

8

Output:

Kategori: Bilangan Genap

Hasil perkalian dengan bilangan berikutnya $8 * 9 = 72$

Input:

25

Output:

Kategori: Bilangan Kelipatan 5

Hasil kuadrat dari $25^2 = 625$

Input:

20

Output:

Kategori: Bilangan Kelipatan 10

Hasil pembagian antara $20 / 10 = 2$