

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11

SWITCH-CASE



Disusun oleh:

NADIFA AZKHIA SYARIF

109082530002

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 jam24 int

    fmt.Println("Masukkan jam (0-23): ")
    fmt.Scan(&jam24)

    switch {
    case jam24 == 0:
        fmt.Println("hasil konverensi: ")
        fmt.Println("12 AM")

    case jam24 == 12:
        fmt.Println("hasil konverensi: ")
        fmt.Println("12 PM")

    case jam24 > 0 && jam24 < 12:
        fmt.Println("hasil konverensi: ")
        fmt.Println(jam24, "AM")
```

```

        case jam24 > 12 && jam24 < 24:

            fmt.Println("hasil konverensi: ")

            fmt.Println(jam24-12, "PM")

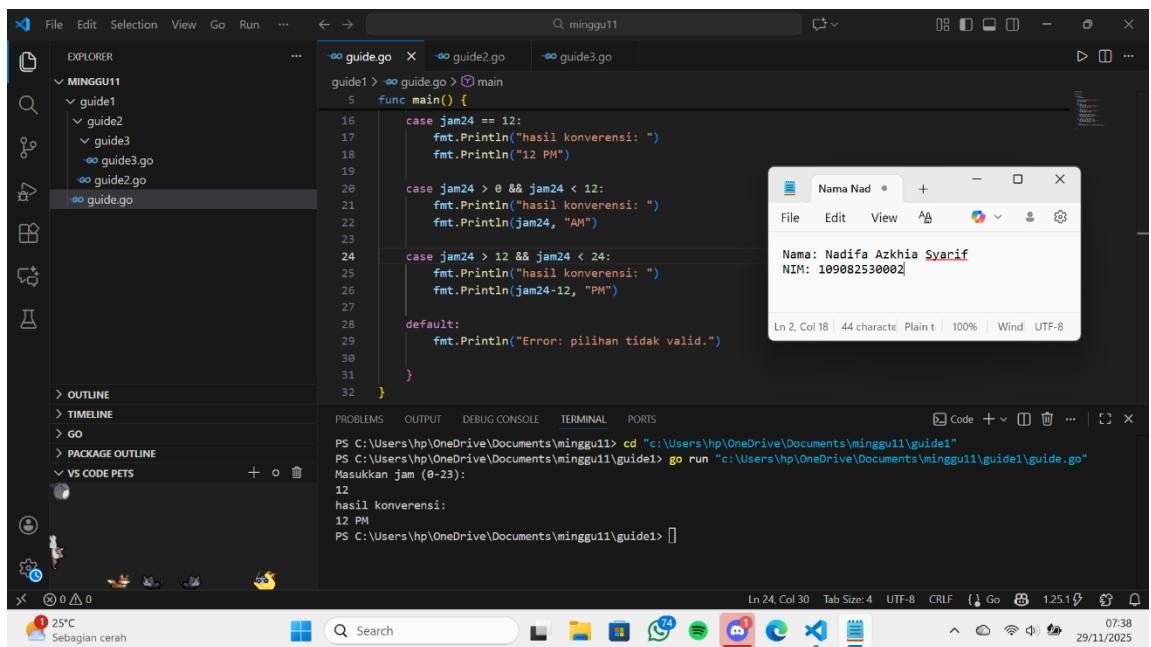
        default:

            fmt.Println("Error: pilihan tidak valid.")


    }
}

```

Screenshot program



Deskripsi program

Program di atas berfungsi untuk mengubah jam dari format 24 jam menjadi format 12 jam (AM/PM). Program mulai dengan membuat variabel jam24 untuk menampung input dari pengguna. Setelah itu, pengguna diminta memasukkan angka jam antara 0 sampai 23. Angka yang dimasukkan kemudian dicek menggunakan struktur switch yang berisi beberapa kondisi.

Jika memasukkan 0, program menampilkan 12 AM, karena jam 00 dalam format 24 jam artinya jam 12 malam. Jika memasukkan 12, program mengubahnya menjadi 12 PM, yaitu jam 12 siang. Untuk angka 1 sampai 11, program langsung menampilkan angka tersebut dengan tambahan AM, karena itu adalah jam pagi. Sementara itu, untuk angka 13 sampai 23, program mengubahnya ke format 12 jam dengan cara mengurangi 12, lalu menambahkan akhiran PM. Jika angka yang dimasukkan tidak berada dalam rentang 0–23, program menampilkan pesan error bahwa input tidak valid. Dengan begitu, program ini membantu melihat waktu dalam format AM/PM dengan benar.

2. Guided 2

Source Code

```
package main

import "fmt"

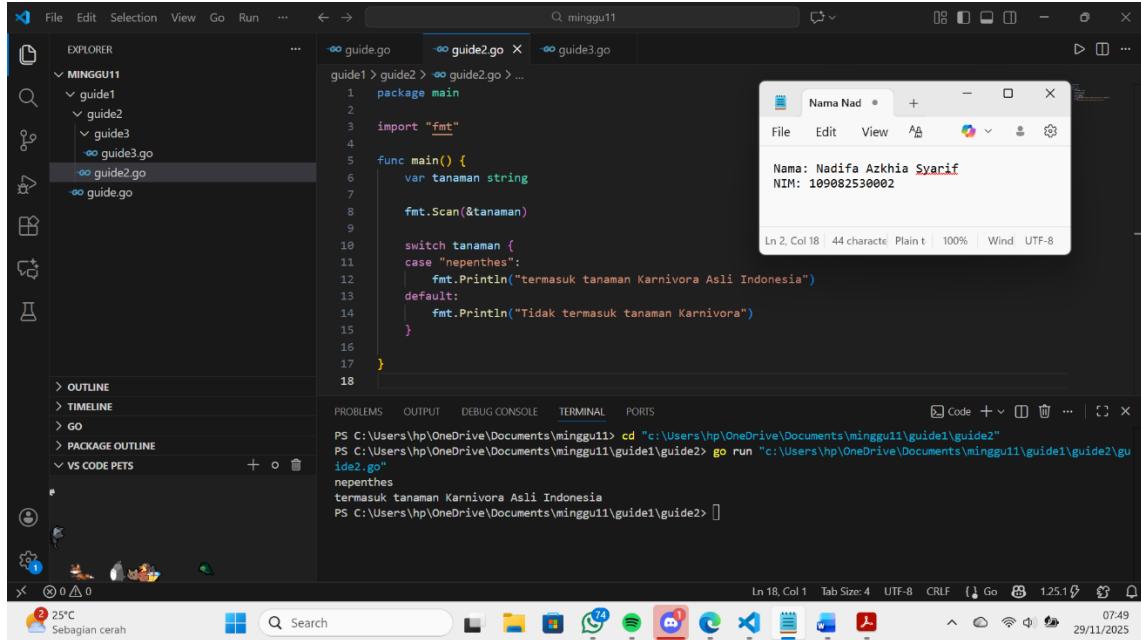
func main() {
    var tanaman string

    fmt.Scan(&tanaman)

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

```
}
```

Screenshot program



Deskripsi program

Program di atas digunakan untuk mengecek apakah suatu tanaman termasuk tanaman karnivora tertentu atau tidak. Program dimulai dengan membuat variabel tanaman yang akan menyimpan input berupa nama tanaman dari pengguna. Setelah itu, pengguna diminta mengetikkan nama tanaman dan program membaca input tersebut menggunakan `fmt.Scan`. Selanjutnya, program menggunakan struktur `switch` untuk membandingkan nilai input dengan kondisi yang tersedia. Jika pengguna memasukkan kata "nepenthes", program akan menampilkan pesan bahwa tanaman tersebut merupakan tanaman karnivora asli Indonesia. Namun, jika inputnya bukan "nepenthes", maka program langsung masuk ke bagian `default` dan menampilkan pesan bahwa tanaman tersebut tidak termasuk tanaman karnivora. Program ini sederhana, tetapi fungsinya jelas: mengidentifikasi apakah nama tanaman yang dimasukkan merupakan tanaman karnivora tertentu atau bukan.

3. Guided 3

Source Code

```
package main

import "fmt"
```

```
func main() {  
  
    var jenis string  
  
    var durasi int  
  
    var tarif int  
  
  
    fmt.Print("Masukkan jenis kendaraan  
(Motor/Mobil/Truk) : ")  
  
    fmt.Scan(&jenis)  
  
  
    fmt.Print("Masukkan durasi parkir (jam) : ")  
  
    fmt.Scan(&durasi)  
  
  
    switch jenis {  
  
        case "Motor":  
  
            if durasi >= 1 && durasi <= 2 {  
  
                tarif = 7000  
  
            } else if durasi > 2 {  
  
                tarif = 9000  
  
            } else {  
  
                tarif = 0  
  
            }  
  
  
        case "Mobil":  
  
            if durasi >= 1 && durasi <= 2 {  
  
                tarif = 15000  
  
            } else if durasi > 2 {
```

```
        tarif = 20000

    } else {

        tarif = 0

    }

case "Truk":

    if durasi >= 1 && durasi <= 2 {

        tarif = 25000

    } else if durasi > 2 {

        tarif = 35000

    } else {

        tarif = 0

    }

default:

    fmt.Println("Jenis kendaraan tidak valid.")

    return

}

fmt.Println("Tarif Parkir: Rp", tarif)

}
```

Screenshoot program

The screenshot shows a Visual Studio Code (VS Code) interface running on a Windows operating system. The main editor area displays a Go program named `guide3.go`. The code defines a function `main()` that takes a string input for vehicle type ('Motor', 'Mobil', or 'Truk') and a duration in minutes. It calculates a parking fee based on the duration: 25000 for 1-2 minutes, 35000 for over 2 minutes, and 0 for other cases. It then prints the result. Below the code, the terminal window shows the command to run the program and the resulting output. A floating terminal window titled "Nama Nad" shows the user's name and NIM. The status bar at the bottom indicates the current line (Ln 47), column (Col 1), tab size (4), encoding (UTF-8), and file save status.

```
File Edit Selection View Go Run ... ← → minggu11
EXPLORER
MINGGU11
guide1
guide2
guide3
guide3.go
guide2.go
guide.go

guide1 > guide2 > guide3 > guide3.go > main
5 func main() {
34
35     case "Truk":
36         if durasi >= 1 && durasi <= 2 {
37             tarif = 25000
38         } else if durasi > 2 {
39             tarif = 35000
40         } else {
41             tarif = 0
42         }
43
44     default:
45         fmt.Println("Jenis kendaraan tidak valid.")
46         return
47     }
48
49     fmt.Println("Tarif Parkir: Rp", tarif)
50 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\hp\OneDrive\Documents\minggu11> cd "c:\Users\hp\OneDrive\Documents\minggu11\guide1\guide2\guide3"
PS C:\Users\hp\OneDrive\Documents\minggu11\guide1\guide2\guide3> go run "c:\Users\hp\OneDrive\Documents\minggu11\guide1\guide2\guide3\guide3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (jam): 2
Tarif Parkir: Rp 7000
PS C:\Users\hp\OneDrive\Documents\minggu11\guide1\guide2\guide3>

File Edit View Aa File Edit View Aa
Nama Nad + - X
Nama: Nadifa Azkhia Syarif
NIM: 109082530002
Ln 2, Col 18 44 character Plain t 100% Wind UTF-8
Code + - x ... x
Ln 47, Col 1 Tab Size: 4 UTF-8 CRLF ⌂ Go ⌂ 1.25.1 ⌂ 07:53
Search
0 2
29/11/2023
```

Deskripsi program

Program di atas digunakan untuk menghitung tarif parkir berdasarkan jenis kendaraan dan durasi parkir. Program meminta pengguna memasukkan jenis kendaraan (Motor, Mobil, atau Truk) serta lama parkir dalam jam. Nilai ini kemudian diproses menggunakan struktur switch untuk menentukan ketentuan tarif sesuai jenis kendaraan. Setiap jenis kendaraan memiliki aturan tarif yang berbeda. Jika durasi parkir 1–2 jam, tarif mengikuti tarif dasar. Jika lebih dari 2 jam, tarifnya lebih tinggi. Motor memiliki tarif 7000 untuk 1–2 jam dan 9000 untuk lebih dari 2 jam. Mobil dikenai tarif 15000 untuk 1–2 jam dan 20000 jika lebih dari 2 jam. Sementara itu, Truk memiliki tarif 25000 untuk 1–2 jam dan 35000 untuk lebih dari 2 jam. Jika jenis kendaraan tidak sesuai pilihan, program akan menampilkan pesan error. Terakhir, program menampilkan tarif parkir yang harus dibayar. Program ini membantu menentukan biaya parkir secara otomatis berdasarkan jenis kendaraan dan durasi.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main()
```

```
var ph float64

fmt.Print("Masukkan kadar 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 a Microsoft Windows desktop environment. In the center is a terminal window titled 'MINGGU11' with the command 'cd "c:\Users\hp\OneDrive\Documents\minggu11\guide1\soal1" go run "c:\Users\hp\OneDrive\Documents\minggu11\guide1\soal1\soal1.go"'. The output of the terminal is: 'Masukkan kadar pH: 9 Air tidak layak minum'. Below the terminal is a code editor window in VS Code. The file 'soal1.go' contains Go code for determining water quality based on pH input. The code includes imports for 'fmt', a 'main' function with a float64 variable 'ph', and a switch statement for pH values. A floating window titled 'Nama Nad' displays the name 'Nadifa Azkhia Syarif' and NIM '109082530002'. The VS Code interface also shows tabs for 'guide.go', 'soal.go', 'guide2.go', and 'guide3.go'.

Deskripsi program

Program di atas adalah yang digunakan untuk menentukan apakah kadar pH suatu air layak diminum atau tidak. Program dimulai dengan membuat variabel ph untuk menyimpan input berupa angka desimal. Setelah itu, diminta untuk memasukkan nilai pH, lalu program membaca input tersebut menggunakan fmt.Scan. Untuk menentukan hasilnya, program menggunakan struktur switch tanpa ekspresi, sehingga setiap case berisi kondisi logis. Jika nilai pH yang dimasukkan berada di luar rentang 0 sampai 14, program langsung menampilkan bahwa nilai pH tidak valid. Jika nilai pH berada pada rentang 6.5 hingga 8.6, program menampilkan bahwa air tersebut layak minum. Selain kondisi tersebut, program otomatis masuk ke bagian default dan menampilkan bahwa air tidak layak minum. Dengan cara ini, program dapat mengelompokkan nilai pH dan menentukan kategori air dengan jelas berdasarkan aturan yang sudah diberikan.

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

}

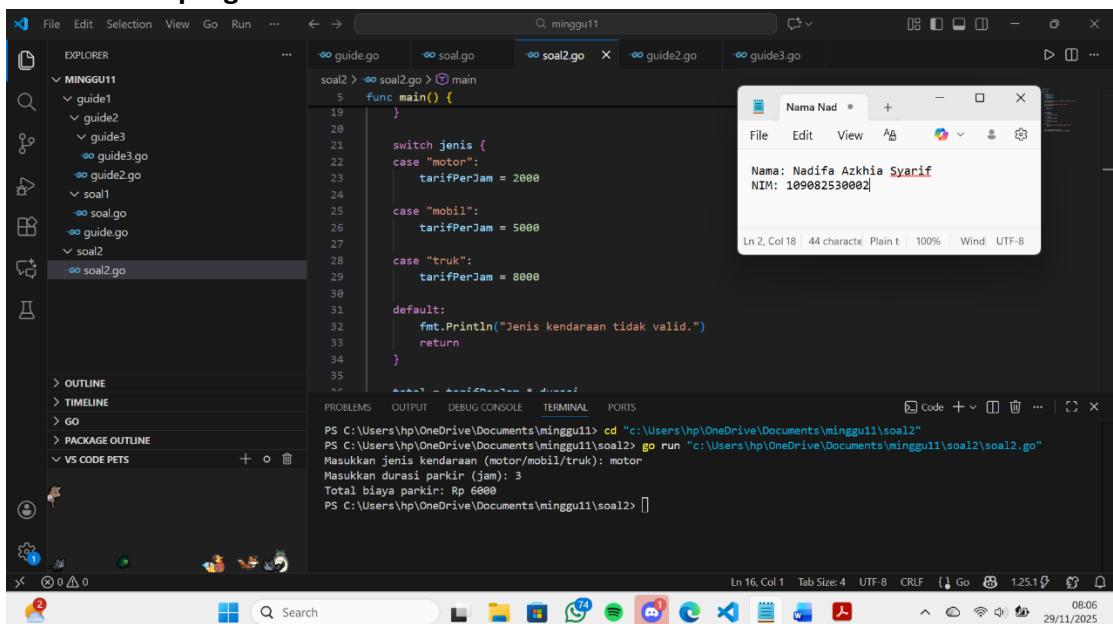
total = tarifPerJam * durasi

fmt.Println("Total biaya parkir: Rp", total)

}

```

Screenshot program



Deskripsi program

Program di atas adalah yang digunakan untuk menghitung total biaya parkir berdasarkan jenis kendaraan dan lama durasi parkir. Program dimulai dengan membuat beberapa variabel, yaitu jenis untuk menyimpan jenis kendaraan, durasi untuk menyimpan lama parkir dalam jam, tarifPerJam untuk menampung tarif per jam, dan total untuk hasil perhitungan. Setelah itu, program meminta masukan berupa jenis kendaraan seperti motor, mobil, atau truk, serta durasi parkir dalam jam. Sebelum menentukan tarif, program mengecek apakah durasi yang dimasukkan kurang dari satu jam. Jika iya, durasi otomatis dianggap satu jam sesuai aturan. Selanjutnya, program menggunakan struktur switch untuk menentukan tarif berdasarkan jenis kendaraan. Motor dikenakan tarif 2000 per jam, mobil sebesar 5000 per jam, dan truk sebesar 8000 per jam. Jika jenis kendaraan tidak sesuai pilihan, program menampilkan pesan bahwa jenis tersebut tidak valid dan langsung berhenti. Setelah tarif per jam ditentukan, program mengalikan tarif tersebut dengan durasi parkir untuk mendapatkan total biaya. Hasil akhir kemudian ditampilkan. Dengan alur

ini, program dapat menghitung biaya parkir secara otomatis sesuai jenis kendaraan dan lama waktu parkir.

3. Tugas 3

Source code

```
package main

import "fmt"

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

    next := n + 1
    var hasil int

    switch {
    case n%10 == 0:
        fmt.Println("Kelipatan 10")
        hasil = n / 10
        fmt.Println("Hasil:", hasil)

    case n%5 == 0:
        fmt.Println("Kelipatan 5")
        hasil = n * n
        fmt.Println("Hasil:", hasil)

    case n%2 == 0:
        fmt.Println("Genap")
        hasil = n * next
        fmt.Println("Hasil:", hasil)

    default:
        fmt.Println("Ganjil")
        hasil = n + next
        fmt.Println("Hasil:", hasil)
    }
}
```

```
}
```

Screenshot program

```
func main() {
    hasil = n / 10
    fmt.Println("Hasil:", hasil)

    case n%5 == 0:
        fmt.Println("Kelipatan 5")
        hasil = n * n
        fmt.Println("Hasil:", hasil)

    case n%2 == 0:
        fmt.Println("Genap")
        hasil = n * next
        fmt.Println("Hasil:", hasil)

    default:
        fmt.Println("Ganjil")
        hasil = n + next
        fmt.Println("Hasil:", hasil)
}
```

PS C:\Users\hp\OneDrive\Documents\minggu11\soal3> go run "c:\Users\hp\OneDrive\Documents\minggu11\soal3\soal3.go"
Masukkan bilangan: 8
Genap
Hasil: 72
PS C:\Users\hp\OneDrive\Documents\minggu11\soal3>

Deskripsi program

Program ini meminta sebuah bilangan lalu memeriksa jenis bilangan tersebut menggunakan struktur switch. Pertama, program menyimpan nilai bilangan berikutnya pada variabel next. Kemudian, *switch* digunakan untuk mengecek kondisi secara berurutan: jika bilangan tersebut kelipatan 10, maka program menampilkan “Kelipatan 10” dan hasil dihitung dari pembagian bilangan dengan 10. Jika bukan, tetapi bilangan itu kelipatan 5, program menampilkan “Kelipatan 5” dan hasil dihitung dari bilangan dikali dirinya sendiri. Jika tidak termasuk keduanya tetapi bilangan genap, program menampilkan “Genap” dan hasil diperoleh dari bilangan dikalikan dengan nilai setelahnya. Jika semua kondisi di atas tidak terpenuhi, berarti bilangan tersebut ganjil dan hasil dihitung dari bilangan ditambah bilangan setelahnya. Setelah menentukan jenis bilangan dan menghitung hasilnya, program menampilkan nilai akhir tersebut.