

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 11
SWITCH CASE**



Disusun oleh:

Rafi Ramadhan

109082500140

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("Masukan jam: ")

    fmt.Scan(&jam)

    switch {

    case jam == 0:
        fmt.Println("12 AM")

    case jam == 12:
        fmt.Println("12 PM")

    case jam > 0 && jam < 12:
        fmt.Println(jam, "AM")

    case jam > 12 && jam < 24:
        fmt.Println(jam-12, "PM")

    default:
        fmt.Println("Waktu tidak valid")
    }
}
```

Screenshoot program

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var jam int
7     fmt.Print("Masukan jam: ")
8     fmt.Scan(&jam)
9
10    switch {
11        case jam == 0:
12            fmt.Println("12 AM")
13        case jam == 12:
14            fmt.Println("12 PM")
15        case jam > 0 && jam < 12:
16            fmt.Println(jam, "AM")
17        case jam > 12 && jam < 24:
18            fmt.Println(jam-12, "PM")
19        default:
20            fmt.Println("Waktu tidak valid")
21    }
22 }

```

PS C:\modul3> go run guide1.go
Masukan jam: 15
3 PM

Deskripsi program

Kode ini adalah program konversi waktu dari format 24 jam ke format 12 jam dengan AM/PM. Program menerima input angka jam (0-23) dan mengubahnya menjadi sistem 12 jam yang umum digunakan, di mana:

- 00:00 menjadi 12 AM
- 12:00 menjadi 12 PM
- 01:00-11:00 menjadi 1-11 AM
- 13:00-23:00 menjadi 1-11 PM

Program akan menampilkan pesan "Waktu tidak valid" jika input di luar rentang 0-23.

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", "pinguicula", "sarracenia",
        "darlingtonia":

            fmt.Println("Termasuk Tanaman Karnivora")

            fmt.Println("Bukan Asli Indonesia")

        default:

            fmt.Println("Tidak termasuk Tanaman Karnivora")

    }

}

```

Screenshoot program

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

- Code Editor:** A dark-themed code editor window titled "guide2.go". It contains Go code for identifying carnivorous plants based on user input.
- Terminal:** A terminal window titled "powershell" showing command-line interactions. It includes three runs of the program with different inputs: "nepenthes" (asli Indonesia), "venus" (bukan asli Indonesia), and "ketopak" (not a carnivorous plant).
- File Explorer:** A sidebar showing the project structure with files like "guide2.go", "main.go", and "modul3.go".
- Output Window:** A window titled "Nama Raf" displaying student information: Nama : Rafi Ramadhan, NIM : 109082500140, Kelas : S1IF-13-02.

Deskripsi program

Kode ini adalah program identifikasi tanaman karnivora yang mengklasifikasikan tanaman berdasarkan input pengguna. Program akan mengelompokkan tanaman menjadi:

- Tanaman karnivora asli Indonesia (nepenthes, drosera)
- Tanaman karnivora bukan asli Indonesia (venus, pinguicula, sarracenia, darlingtonia)
- Bukan tanaman karnivora (untuk input lainnya)

Program berguna untuk membantu mengenali jenis-jenis tanaman karnivora dan asal daerahnya.

3. Guided 3 Source Code

```
package main
```

```
import "fmt"

func main() {
    var jenis string
    var durasi int

    fmt.Print("Jenis kendaraan (Motor/Mobil/Truk): ")
    fmt.Scan(&jenis)

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

    switch jenis {
    case "Motor":
        if durasi <= 2 {
            fmt.Println("Rp 7000")
        } else {
            fmt.Println("Rp 9000")
        }
    case "Mobil":
        if durasi <= 2 {
            fmt.Println("Rp 15000")
        } else {
            fmt.Println("Rp 20000")
        }
    case "Truk":
        if durasi <= 2 {
            fmt.Println("Rp 25000")
        } else {
            fmt.Println("Rp 35000")
        }
    default:
```

```

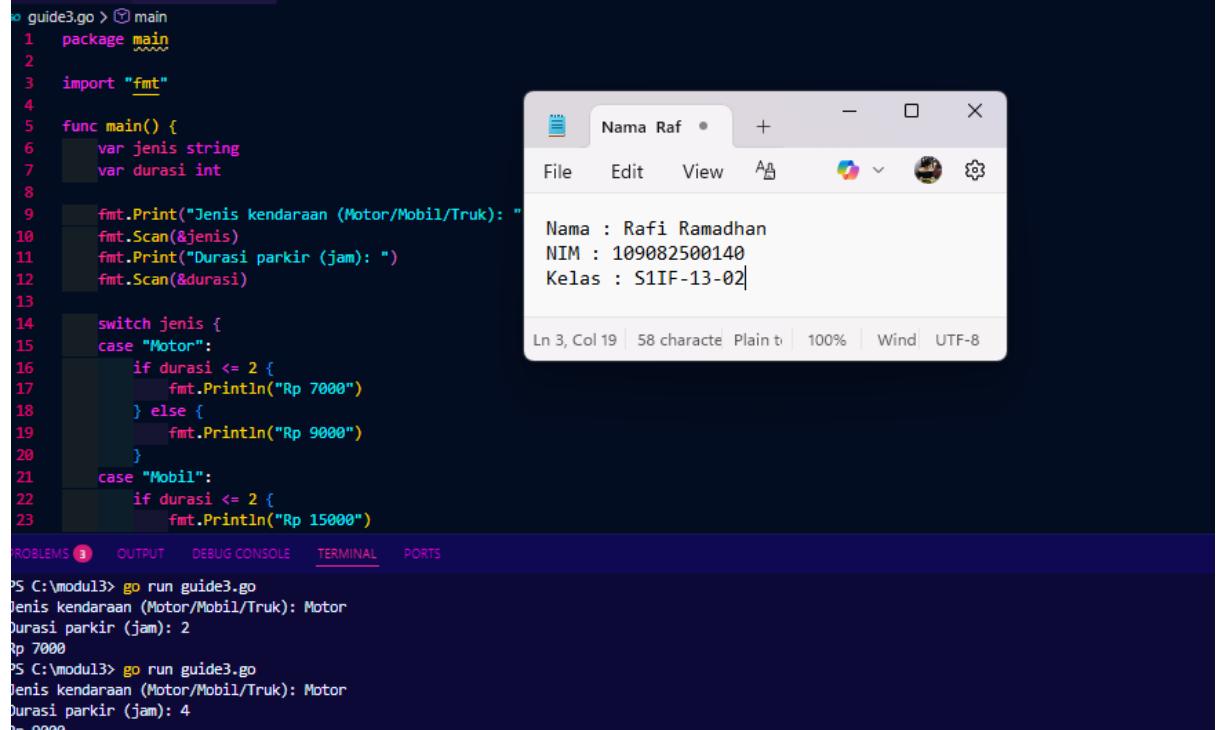
        fmt.Println("Jenis kendaraan atau durasi waktu
tidak valid")

    }

}

```

Screenshot program



```

guide3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var jenis string
7     var durasi int
8
9     fmt.Print("Jenis kendaraan (Motor/Mobil/Truk): ")
10    fmt.Scan(&jenis)
11    fmt.Print("Durasi parkir (jam): ")
12    fmt.Scan(&durasi)
13
14    switch jenis {
15        case "Motor":
16            if durasi <= 2 {
17                fmt.Println("Rp 7000")
18            } else {
19                fmt.Println("Rp 9000")
20            }
21        case "Mobil":
22            if durasi <= 2 {
23                fmt.Println("Rp 15000")

```

The screenshot shows a code editor with a Go file named `guide3.go`. The code defines a `main` package with a `main` function. It prompts the user for the type of vehicle and the duration of parking. Based on the vehicle type, it prints a flat rate. The code uses `fmt.Println` and `fmt.Scan` for input and output. Below the code editor is a terminal window showing the execution of the program and its output. The terminal shows two runs of the program, both resulting in `Rp 7000` for a 2-hour motorbike parking duration.

Deskripsi program

Kode ini adalah program perhitungan tarif parkir yang menghitung biaya parkir berdasarkan jenis kendaraan (Motor, Mobil, Truk) dan durasi parkir. Program menggunakan sistem tarif flat dengan dua kategori waktu:

- ≤ 2 jam: tarif normal
- > 2 jam: tarif lebih tinggi

Program akan menampilkan pesan error jika input jenis kendaraan tidak valid.

TUGAS

1. Tugas 1

Source code

```

package main

import "fmt"

```

```

func main() {
    var ph float64

    fmt.Print("masukan kadar pH: ")
    fmt.Scan(&ph)

    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")
    case ph > 14:
        fmt.Println("Input tidak valid")
    }
}

```

Screenshot program

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

- Code Editor:** Displays the source code for `soal1.go`. The code defines a `main()` function that prints a prompt, reads input, and uses a `switch` statement to determine if the water is drinkable based on pH levels.
- Terminal:** Shows the command `go run soal1.go` being run twice. The first run with pH 7.5 outputs "Air layak minum". The second run with pH 10 outputs "Air tidak layak minum".
- Output Window:** A modal window titled "Nama Raf" displays the user's information: Nama : Rafi Ramadhan, NIM : 109082500140, Kelas : S1IF-13-02.

Deskripsi program

Kode ini buat ngecek layak atau nggaknya air minum berdasarkan pH. Cara kerjanya:

- Input angka pH

- Kalo pH antara 6.5-8.6 → "Air layak minum"
- Kalo pH di bawah 6.5 atau di atas 8.6 → "Air tidak layak minum"
- Kalo pH lebih dari 14 → "Input tidak valid"

Jadi simpelnya, program ini nentuin air bisa diminum atau enggak berdasarkan tingkat keasaman atau kebasanya.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var jam int
    var tarif int
    var total int

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

    fmt.Print("Masukan Durasi Parkir (dalam jam) : ")
    fmt.Scan(&jam)

    if jam < 1 {
        jam = 1
    }

    switch kendaraan {
    case "motor":
        tarif = 2000
    case "mobil":
        tarif = 5000
    case "truk":
        tarif = 8000
    default:
        fmt.Println("Jenis Kendaraan Tidak Valid")
    }
}
```

```

        return
    }

    total = tarif * jam

    fmt.Printf("Total Tarif Parkir: Rp %d\n", total)
}

```

Screenshoot program

The screenshot shows a code editor with a dark theme. The code is as follows:

```

soal2.go > main
1 package main
2
3 import "fmt"
4 func main() {
5     var kendaraan string
6     var jam int
7     var tarif int
8     var total int
9
10    fmt.Print("Masukan jenis kendaraan (motor/mobil/truk) : ")
11    fmt.Scan(&kendaraan)
12    fmt.Print("Masukan Durasi Parkir (dalam jam) : ")
13    fmt.Scan(&jam)
14
15    if jam < 1 {
16        jam = 1
17    }
18
19    switch kendaraan {
20        case "motor":
21            tarif = 2000
22        case "mobil":
23            tarif = 5000
24        case "truk":
25    }
26
27    total = tarif * jam
28
29    fmt.Printf("Total Tarif Parkir: Rp %d\n", total)
30}

```

A floating terminal window titled "Nama Raf" displays the following information:

- Nama : Rafi Ramadhan
- NIM : 109082500140
- Kelas : S1IF-13-02

At the bottom of the terminal window, status bar text includes "Ln 3, Col 19 | 58 character Plain text 100% Wind UTF-8".

Below the code editor, a navigation bar shows tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS.

The terminal window below shows the execution of the program:

```

PS C:\modul3> go run soal2.go
Masukan jenis kendaraan (motor/mobil/truk) : motor
Masukan Durasi Parkir (dalam jam) : 2
Total Tarif Parkir: Rp 4000
PS C:\modul3> go run soal2.go
Masukan jenis kendaraan (motor/mobil/truk) : mobil
Masukan Durasi Parkir (dalam jam) : 1
Total Tarif Parkir: Rp 5000

```

Deskripsi program

Kode ini buat hitung tarif parkir berdasarkan jenis kendaraan dan durasi. Cara kerjanya:

- Input jenis kendaraan (motor/mobil/truk) dan jam parkir
- Kalo jam parkir kurang dari 1, dianggap 1 jam (minimum charge)
- Motor bayar Rp 2.000/jam, mobil Rp 5.000/jam, truk Rp 8.000/jam
- Total bayar = tarif × jam parkir

Kalo jenis kendaraan ga valid, program langsung berhenti. Simple banget buat sistem parkir dasar.

3. Tugas 3

Source code

```
package main

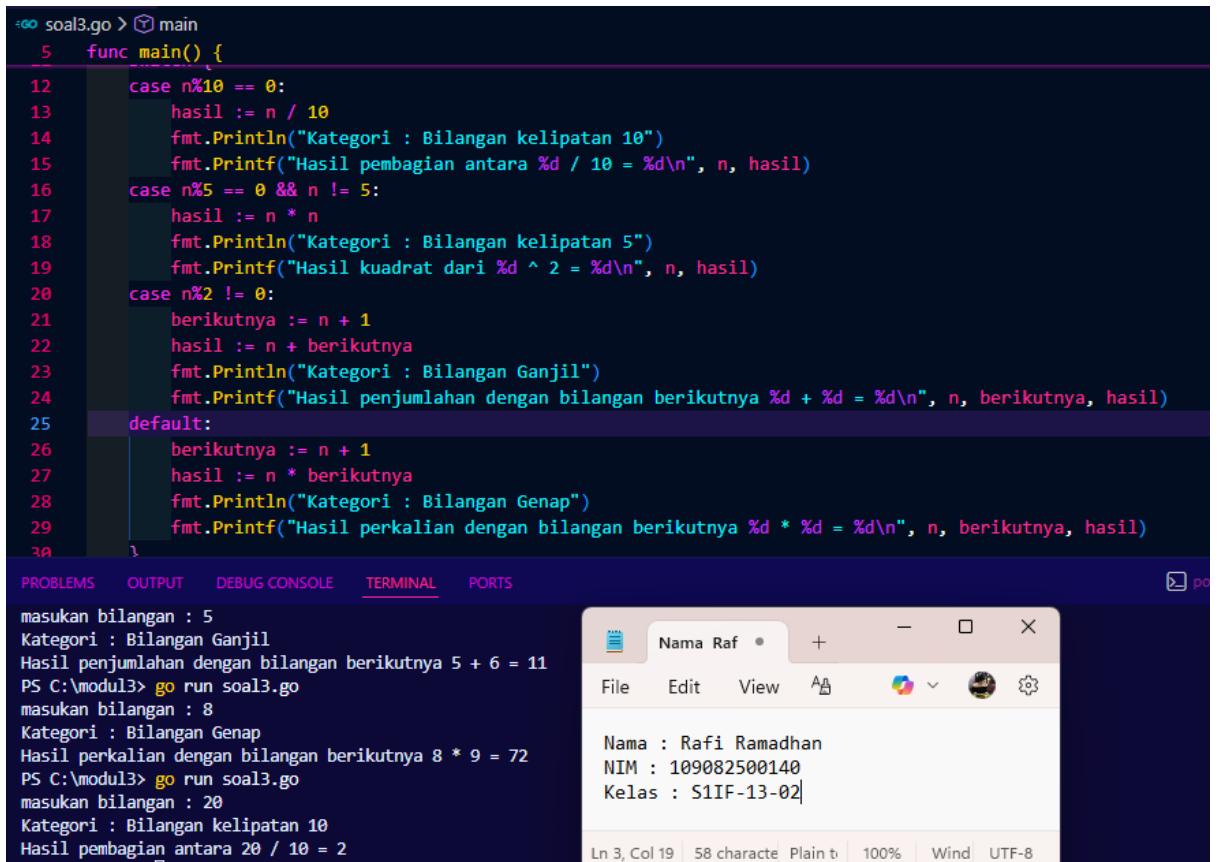
import "fmt"

func main() {
    var n int

    fmt.Print("masukan bilangan : ")
    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:
        berikutnya := n + 1
        hasil := n + berikutnya
        fmt.Println("Kategori : Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n", n, berikutnya, hasil)
    default:
        berikutnya := n + 1
        hasil := n * berikutnya
        fmt.Println("Kategori : Bilangan Genap")
        fmt.Printf("Hasil perkalian dengan bilangan
berikutnya %d * %d = %d\n", n, berikutnya, hasil)
    }
}
```

Screenshoot program



```
soal3.go > main
func main() {
    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:
        berikutnya := n + 1
        hasil := n + berikutnya
        fmt.Println("Kategori : Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, berikutnya, hasil)
    default:
        berikutnya := n + 1
        hasil := n * berikutnya
        fmt.Println("Kategori : Bilangan Genap")
        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", n, berikutnya, hasil)
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
masukan bilangan : 5
Kategori : Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS C:\modul3> go run soal3.go
masukan bilangan : 8
Kategori : Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\modul3> go run soal3.go
masukan bilangan : 20
Kategori : Bilangan kelipatan 10
Hasil pembagian antara 20 / 10 = 2
```

Nama Raf

File Edit View A A File Edit View A A Nama : Rafi Ramadhan
NIM : 109082500140
Kelas : S1IF-13-02

Deskripsi program

Kode ini merupakan program yang memproses sebuah bilangan integer berdasarkan karakteristiknya dan melakukan operasi matematika tertentu sesuai dengan kategori bilangan tersebut.

Program bekerja dengan menerima input sebuah bilangan, kemudian melalui struktur switch case menentukan kategori bilangan dan melakukan perhitungan:

- Kelipatan 10: bilangan dibagi 10
- Kelipatan 5 (kecuali 5): bilangan dikuadratkan
- Bilangan ganjil: dijumlahkan dengan bilangan berikutnya
- Bilangan genap: dikalikan dengan bilangan berikutnya

Program ini efektif untuk demonstrasi penggunaan conditional logic dengan multiple conditions dan menunjukkan bagaimana sebuah input dapat diproses dengan cara berbeda berdasarkan sifat-sifat matematikanya.