

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 11
SWITCH-CASE**



Disusun oleh:

JEREMY CHRISTIAN SIHOMBING

109082500178

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

**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

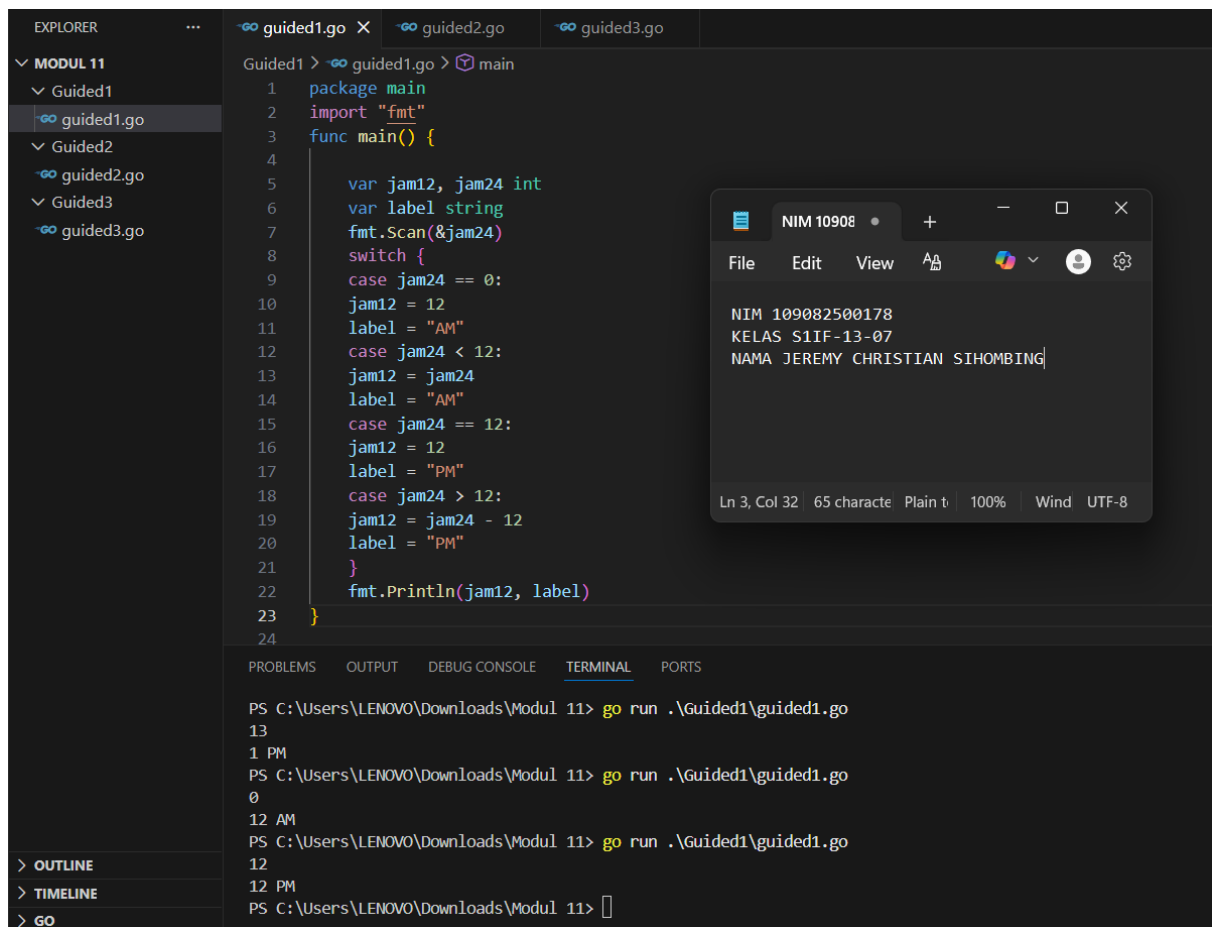
        label = "PM"

    }

    fmt.Println(jam12, label)

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk mengonversi jam dari format 24 jam menjadi format 12 jam. Pengguna cukup memasukkan satu angka yang mewakili jam dalam format 24 jam. Setelah menerima input, program menggunakan struktur switch untuk mengecek kondisi jam tersebut. Jika nilai jam yang dimasukkan adalah 0, program mengubahnya menjadi 12 AM karena 00:00 dalam format 24 jam sama dengan 12 AM pada format 12 jam. Jika nilai jam berada antara 1 sampai 11, program menampilkan jam apa adanya dan menambahkan label AM karena masih berada pada periode pagi. Untuk input bernilai 12, program langsung menampilkannya sebagai 12 PM. Sementara itu, jika jam lebih besar dari 12, program akan mengurangnya dengan 12 untuk mendapatkan format 12 jam, kemudian menandainya dengan PM.

Setelah proses pengecekan selesai, program mencetak hasil konversi berupa jam dalam format 12 jam beserta label AM atau PM sesuai kondisi yang terjadi.

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":

        fmt.Println("Termasuk Tanaman Karnivora")

        fmt.Println("Asli Indonesia")

    case "venus":

        fmt.Println("Termasuk Tanaman Karnivora")

        fmt.Println("Bukan Asli Indonesia")

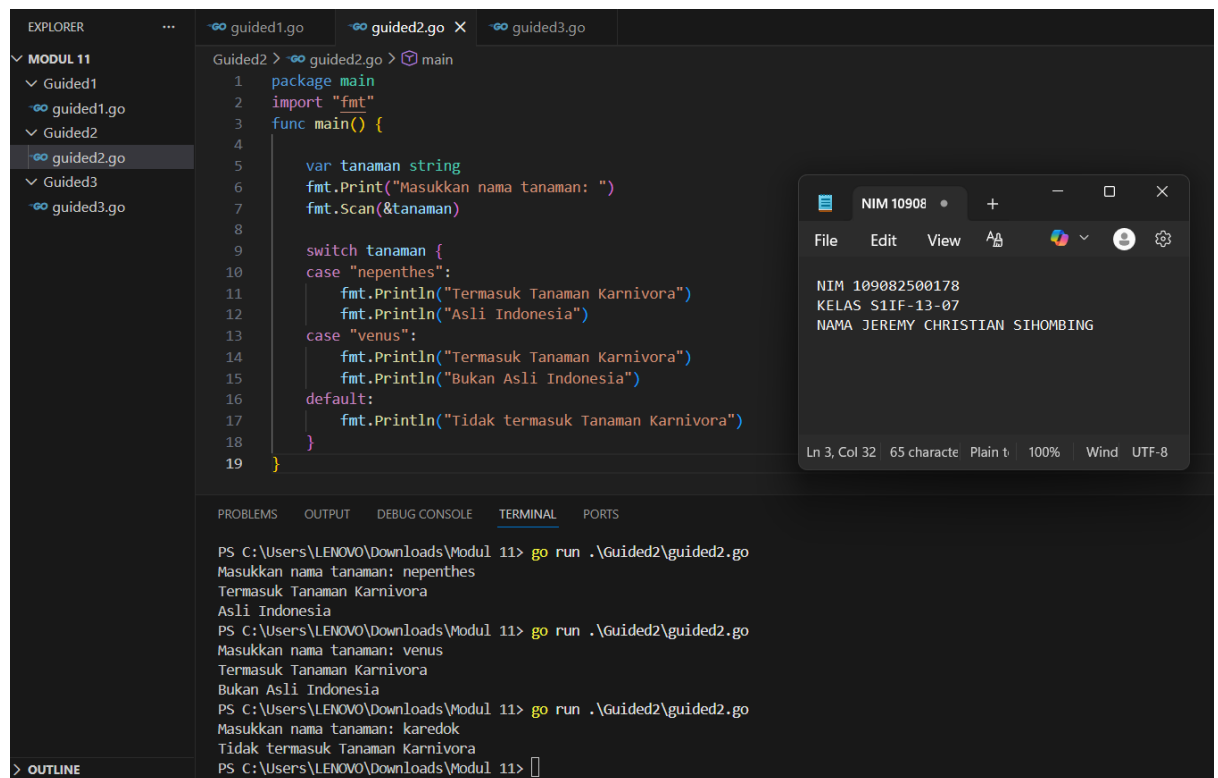
    default:

        fmt.Println("Tidak termasuk Tanaman Karnivora")

    }

}
```

Screenshoot program



```
EXPLORER
...
MODUL 11
  Guided1
  guided1.go
  Guided2
  guided2.go
  Guided3
  guided3.go

Guided2 > guided2.go > main
1 package main
2 import "fmt"
3 func main() {
4
5     var tanaman string
6     fmt.Print("Masukkan nama tanaman: ")
7     fmt.Scan(&tanaman)
8
9     switch tanaman {
10    case "nepenthes":
11        fmt.Println("Termasuk Tanaman Karnivora")
12        fmt.Println("Asli Indonesia")
13    case "venus":
14        fmt.Println("Termasuk Tanaman Karnivora")
15        fmt.Println("Bukan Asli Indonesia")
16    default:
17        fmt.Println("Tidak termasuk Tanaman Karnivora")
18    }
19 }
```

```
File Edit View A A 100% Wind UTF-8
NIM 109082500178
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING
Ln 3, Col 32 65 character Plain t 100% Wind UTF-8
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Guided2\guided2.go
Masukkan nama tanaman: nepenthes
Termasuk Tanaman Karnivora
Asli Indonesia
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Guided2\guided2.go
Masukkan nama tanaman: venus
Termasuk Tanaman Karnivora
Bukan Asli Indonesia
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Guided2\guided2.go
Masukkan nama tanaman: karedok
Tidak termasuk Tanaman Karnivora
PS C:\Users\LENOVO\Downloads\Modul 11>
```

Deskripsi program

Program ini digunakan untuk mengidentifikasi apakah suatu tanaman termasuk tanaman karnivora dan apakah tanaman tersebut berasal dari Indonesia atau tidak. Ketika program dijalankan, pengguna diminta memasukkan nama tanaman dalam bentuk teks. Setelah input diterima, program akan memprosesnya menggunakan struktur switch untuk mencocokkan nama tanaman dengan beberapa kategori yang sudah ditentukan. Jika pengguna memasukkan kata nepenthes, program akan menampilkan bahwa tanaman tersebut merupakan tanaman karnivora sekaligus asli Indonesia. Jika input berupa venus, program juga mengelompokkan tanaman tersebut sebagai tanaman karnivora, tetapi bukan asli Indonesia. Apabila nama tanaman yang dimasukkan tidak termasuk dalam kedua kategori tersebut, program menampilkan bahwa tanaman tersebut bukan tanaman karnivora. Melalui pengecekan ini, program memberikan informasi sederhana berdasarkan nama tanaman yang dimasukkan oleh pengguna.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var kendaraan string

    var durasi int

    fmt.Print("Masukkan jenis kendaraan  
(Motor/Mobil/Truk): ")

    fmt.Scan(&kendaraan)

    fmt.Print("Masukkan durasi parkir (dalam jam): ")

    fmt.Scan(&durasi)

    switch kendaraan {

    case "Motor":

        if durasi <= 2 {

            fmt.Print("Tarif Parkir: Rp 7000")

        } else if durasi > 2 {

            fmt.Print("Tarif Parkir: Rp 9000")

        }

    case "Mobil":

        if durasi <= 2 {

            fmt.Print("Tarif Parkir: Rp 15000")

        } else if durasi > 2 {

            fmt.Print("Tarif Parkir: Rp 20000")

        }

    case "Truk":
```

```
        if durasi <= 2 {  
            fmt.Print("Tarif Parkir: Rp 25000")  
        } else if durasi > 2 {  
            fmt.Print("Tarif Parkir: Rp 35000")  
        }  
        default:  
            fmt.Println("Jenis kendaraan atau durasi parkir  
tidak valid")  
            fmt.Println("Tarif Parkir: Rp 0")  
        }  
    }  
}
```

Screenshoot program

The screenshot displays a Go IDE with a project named 'MODUL 11' containing three files: 'guided1.go', 'guided2.go', and 'guided3.go'. The 'guided3.go' file is open, showing a Go program that calculates parking fees based on vehicle type and duration. The program uses a switch statement for vehicle types (Motor, Mobil, Truk) and nested if-else statements for duration-based pricing. A default case handles invalid inputs.

```
1 package main
2 import "fmt"
3 func main() {
4     var kendaraan string
5     var durasi int
6     fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
7     fmt.Scan(&kendaraan)
8     fmt.Print("Masukkan durasi parkir (dalam jam): ")
9     fmt.Scan(&durasi)
10    switch kendaraan {
11    case "Motor":
12        if durasi <= 2 {
13            fmt.Print("Tarif Parkir: Rp 7000")
14        } else if durasi > 2 {
15            fmt.Print("Tarif Parkir: Rp 9000")
16        }
17    case "Mobil":
18        if durasi <= 2 {
19            fmt.Print("Tarif Parkir: Rp 15000")
20        } else if durasi > 2 {
21            fmt.Print("Tarif Parkir: Rp 20000")
22        }
23    case "Truk":
24        if durasi <= 2 {
25            fmt.Print("Tarif Parkir: Rp 25000")
26        } else if durasi > 2 {
27            fmt.Print("Tarif Parkir: Rp 35000")
28        }
29    default:
30        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
31        fmt.Print("Tarif Parkir: Rp 0")
32    }
33 }
```

The terminal output shows the program's execution with various inputs:

```
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Guided3\guided3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Guided3\guided3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Guided3\guided3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir: Rp 9000
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Guided3\guided3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Guided3\guided3.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
```

On the right, a text editor window titled 'NIM 10908' contains the following text:

```
NIM 109082500178
KELAS SIIF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING
```

The text editor's status bar indicates the cursor is at line 3, column 32, with 65 characters, in plain text, 100% zoom, and UTF-8 encoding.

Deskripsi program

Program ini digunakan untuk menentukan tarif parkir berdasarkan jenis kendaraan dan lama waktu parkir. Saat dijalankan, program terlebih dahulu meminta pengguna memasukkan jenis kendaraan, yaitu Motor, Mobil, atau Truk. Setelah itu, pengguna diminta mengisi durasi parkir dalam satuan jam.

Setelah menerima kedua input tersebut, program menggunakan struktur switch untuk membedakan perhitungan tarif sesuai kategori kendaraan. Untuk kendaraan Motor, tarif yang dikenakan adalah Rp 7.000 jika durasi parkir maksimal 2 jam, dan Rp 9.000 jika melebihi 2 jam. Untuk Mobil, tarifnya Rp 15.000 untuk durasi hingga 2 jam, dan Rp 20.000 jika lebih dari 2 jam. Sementara itu, kendaraan Truk dikenakan tarif Rp 25.000 untuk durasi 2 jam atau kurang, dan Rp 35.000 apabila durasi parkirnya lebih lama. Jika pengguna memasukkan jenis kendaraan yang tidak sesuai dengan ketiga kategori tersebut, program akan menampilkan pesan bahwa input tidak valid dan menetapkan tarif sebesar Rp 0. Hasil akhir berupa tarif parkir ditampilkan sesuai kondisi yang terpenuhi.

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")

    }

}
```

Screenshoot program

The screenshot shows a Go program in a VS Code editor. The Explorer on the left shows a project structure for 'MODUL 11' with files 'guided1.go', 'guided2.go', 'guided3.go', and 'soal1.go'. The main editor displays the code for 'soal1.go'.

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

The bottom panel shows the 'TERMINAL' output:

```
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal1\soal1.go
Masukkan kadar pH: 8.6
Air layak minum
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal1\soal1.go
Masukkan kadar pH: 9
Air tidak layak minum
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal1\soal1.go
Masukkan kadar pH: 16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.
PS C:\Users\LENOVO\Downloads\Modul 11>
```

Overlaid on the bottom right is a window titled 'NIM 10908' containing the following text:

```
NIM 109082500178
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING
```

The status bar at the bottom indicates 'Ln 3, Col 32', '65 character', 'Plain t', '100%', 'Wind', and 'UTF-8'.

Deskripsi program

Program ini meminta input pH dari pengguna. Pada bagian pengecekan, digunakan struktur switch tanpa ekspresi, sehingga setiap case berisi kondisi logis. Cara ini adalah teknik yang umum dipakai di Golang untuk menangani percabangan berdasarkan rentang nilai. Kasus pertama mengecek apakah pH berada di luar batas 0 sampai 14. Jika iya, program menampilkan bahwa pH tidak valid. Jika pH valid, program masuk ke kasus berikutnya untuk menentukan kelayakan air minum. Air dinilai layak diminum jika nilai pH berada dalam rentang 6.5 hingga 8.6. Jika tidak memenuhi kedua kondisi tersebut, blok default akan menampilkan bahwa air tidak layak minum.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var kendaraan string

    var jam, tarif, Motor, Mobil, Truk int

    fmt.Print("Masukkan jenis kendaraan dan durasi parkir dalam jam: ")

    fmt.Scan(&kendaraan, &jam)

    switch kendaraan {

    case "Motor":

        if jam >= 1 {

            tarif = 2000

            Motor = tarif * jam

            fmt.Print("Rp.", Motor)

        }

    case "Mobil":

        if jam >= 1 {

            tarif = 5000

            Mobil = tarif * jam

            fmt.Print("Rp.", Mobil)

        }

    case "Truk":
```

```

        if jam >= 1 {

            tarif = 8000

            Truk = tarif * jam

            fmt.Print("Rp.", Truk)

        }

    }

}

```

Screenshoot program

The screenshot displays a Go IDE with the following components:

- EXPLORER:** Shows a project structure with 'MODUL 11' containing files like 'guided1.go', 'guided2.go', 'guided3.go', 'Soal1', 'Soal2', and 'soal2.go'.
- EDITOR:** Displays the source code for 'soal2.go'. The code defines a 'main' package with a 'main' function. It uses a switch statement to handle different vehicle types ('Motor', 'Mobil', 'Truk') based on the input 'jam' (duration). For each vehicle, it calculates the 'tarif' (rate) and the total cost ('Motor', 'Mobil', 'Truk') based on the input 'jam'.
- TERMINAL:** Shows the execution output of the program. It prompts the user to 'Masukkan jenis kendaraan dan durasi parkir dalam jam:' and shows the results for 'Motor 3 jam' (Rp.6000), 'Mobil 1 jam' (Rp.5000), and 'Truk 5 jam' (Rp.40000).

```

Soal2 > go run .\soal2.go
package main
import "fmt"
func main() {

    var kendaraan string
    var jam, tarif, Motor, Mobil, Truk int
    fmt.Print("Masukkan jenis kendaraan dan durasi parkir dalam jam: ")
    fmt.Scan(&kendaraan, &jam)

    switch kendaraan {
    case "Motor":
        if jam >= 1 {
            tarif = 2000
            Motor = tarif * jam
            fmt.Print("Rp.", Motor)
        }
    case "Mobil":
        if jam >= 1 {
            tarif = 5000
            Mobil = tarif * jam
            fmt.Print("Rp.", Mobil)
        }
    case "Truk":
        if jam >= 1 {
            tarif = 8000
            Truk = tarif * jam
            fmt.Print("Rp.", Truk)
        }
    }
}

```

PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal2\soal2.go
Masukkan jenis kendaraan dan durasi parkir dalam jam: Motor 3 jam
Rp.6000
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal2\soal2.go
Masukkan jenis kendaraan dan durasi parkir dalam jam: Mobil 1 jam
Rp.5000
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal2\soal2.go
Masukkan jenis kendaraan dan durasi parkir dalam jam: Truk 5 jam
Rp.40000
PS C:\Users\LENOVO\Downloads\Modul 11>

Deskripsi program

Program ini berfungsi untuk menghitung total biaya parkir berdasarkan jenis kendaraan dan lama parkir yang dimasukkan oleh pengguna. Di awal program, pengguna diminta menginput jenis kendaraan dan jumlah jam parkir. Jenis kendaraan yang dapat dimasukkan adalah Motor, Mobil, atau Truk. Setelah input diterima, program menggunakan struktur switch-case untuk menentukan tarif parkir sesuai dengan kendaraan yang dipilih. Motor memiliki tarif 2.000 rupiah per jam, Mobil 5.000 rupiah per jam, dan Truk 8.000 rupiah per jam.

Jika durasi parkir yang dimasukkan minimal satu jam, program akan mengalikan tarif yang sesuai dengan jumlah jam parkir untuk mendapatkan total biaya. Hasil perhitungan kemudian langsung ditampilkan dalam bentuk "Rp." diikuti total biayanya. Dengan alur tersebut, program dapat menghitung biaya parkir secara sederhana sesuai dengan jenis kendaraan dan durasi parkir yang diberikan oleh pengguna.

3. Tugas 3

Source code

```
package main
import "fmt"
func main() {

    var bilangan int
    fmt.Print("Masukkan sebuah bilangan bulat: ")
    fmt.Scan(&bilangan)

    switch {
    case bilangan % 10 == 0:
        fmt.Println("Kategori: Bilangan kelipatan 10")
        bagi := bilangan / 10
        fmt.Println("Hasil pembagian antara", bilangan,
"/", "10", "=", bagi)
    case bilangan % 5 == 0 && bilangan != 5:
        fmt.Println("Kategori: Bilangan kelipatan 5")
        pangkat := bilangan * bilangan
        fmt.Println("Hasil kuadrat dari",
bilangan,"^2", "=", pangkat )
    case bilangan % 2 == 1:
        fmt.Println("Kategori: Bilangan Ganjil")
        tambah := bilangan + 1
        total := bilangan + tambah
        fmt.Println("Hasil penjumlahan dengan bilangan
berikutnya", bilangan, "+", tambah, "=", total)
    case bilangan % 2 == 0:
        fmt.Println("Kategori: Bilangan Genap")
        kali := bilangan + 1
        total := bilangan * kali
        fmt.Println("Hasil perkalian dengan bilangan
berikutnya", bilangan, "*", kali, "=", total)
    }
}
```

Screenshoot program

The screenshot displays a Go IDE with a project named 'MODUL 11'. The Explorer panel on the left shows a directory structure with files 'guided1.go', 'guided2.go', 'guided3.go', and a subdirectory 'Soal3' containing 'soal1.go', 'soal2.go', and 'soal3.go'. The 'soal3.go' file is selected and its code is visible in the editor. The code defines a `main` function that takes an integer `bilangan` and performs various operations based on its value using a `switch` statement. The operations include checking for multiples of 10, 5, and 2, calculating squares, and performing addition and multiplication. The output of the program is shown in the terminal panel at the bottom, which includes the input of the number 5 and the resulting calculations and category assignments.

```
1 package main
2 import "fmt"
3 func main() {
4
5     var bilangan int
6     fmt.Print("Masukkan sebuah bilangan bulat: ")
7     fmt.Scan(&bilangan)
8
9     switch {
10    case bilangan % 10 == 0:
11        fmt.Println("Kategori: Bilangan kelipatan 10")
12        bagi := bilangan / 10
13        fmt.Println("Hasil pembagian antara", bilangan, "/", "10", "=", bagi)
14    case bilangan % 5 == 0 && bilangan != 5:
15        fmt.Println("Kategori: Bilangan kelipatan 5")
16        pangkat := bilangan * bilangan
17        fmt.Println("Hasil kuadrat dari", bilangan, "^2", "=", pangkat )
18    case bilangan % 2 == 1:
19        fmt.Println("Kategori: Bilangan Ganjil")
20        tambah := bilangan + 1
21        total := bilangan + tambah
22        fmt.Println("Hasil penjumlahan dengan bilangan berikutnya", bilangan, "+", tambah, "=", total)
23    case bilangan % 2 == 0:
24        fmt.Println("Kategori: Bilangan Genap")
25        kali := bilangan + 1
26        total := bilangan * kali
27        fmt.Println("Hasil perkalian dengan bilangan berikutnya", bilangan, "*", kali, "=", total)
28    }
29 }
```

Masukkan sebuah bilangan bulat: 5
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal3\soal3.go
Masukkan sebuah bilangan bulat: 8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal3\soal3.go
Masukkan sebuah bilangan bulat: 25
Kategori: Bilangan kelipatan 5
Hasil kuadrat dari 25 ^2 = 625
PS C:\Users\LENOVO\Downloads\Modul 11> go run .\Soal3\soal3.go
Masukkan sebuah bilangan bulat: 20
Kategori: Bilangan kelipatan 10
Hasil pembagian antara 20 / 10 = 2

Deskripsi program

Program ini mulai dengan meminta pengguna memasukkan satu bilangan bulat. Setelah itu, program akan mengecek bilangan tersebut termasuk kategori apa lewat beberapa kondisi di dalam switch. Kalau bilangan itu habis dibagi 10, program bilang kalau itu *kelipatan 10*, lalu menghitung hasil pembagian bilangan tersebut dengan 10 dan menampilkannya. Kalau bukan kelipatan 10 tapi habis dibagi 5 (dan bukan angka 5), program menganggapnya *kelipatan 5*. Di bagian ini program menghitung kuadrat dari bilangan tersebut lalu menampilkan hasilnya. Jika bukan dua kategori tadi, program mengecek apakah bilangannya ganjil. Kalau ganjil, program menjumlahkan bilangan itu dengan bilangan setelahnya (bilangan + 1), lalu menampilkan hasil penjumlahannya. Terakhir, kalau semua kondisi sebelumnya tidak cocok, berarti bilangan itu genap. Untuk bilangan genap, program mengalikan bilangan itu dengan bilangan setelahnya (bilangan × (bilangan + 1) dan menampilkan hasilnya.

Intinya, program menentukan dulu bilangan itu masuk kategori apa, baru menghitung operasi yang cocok dengan kategori tersebut.