

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 11  
SWITCH-CASE**



**Disusun oleh:**

**ALIN KARISA HIZANNAH**

**109082500010**

**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

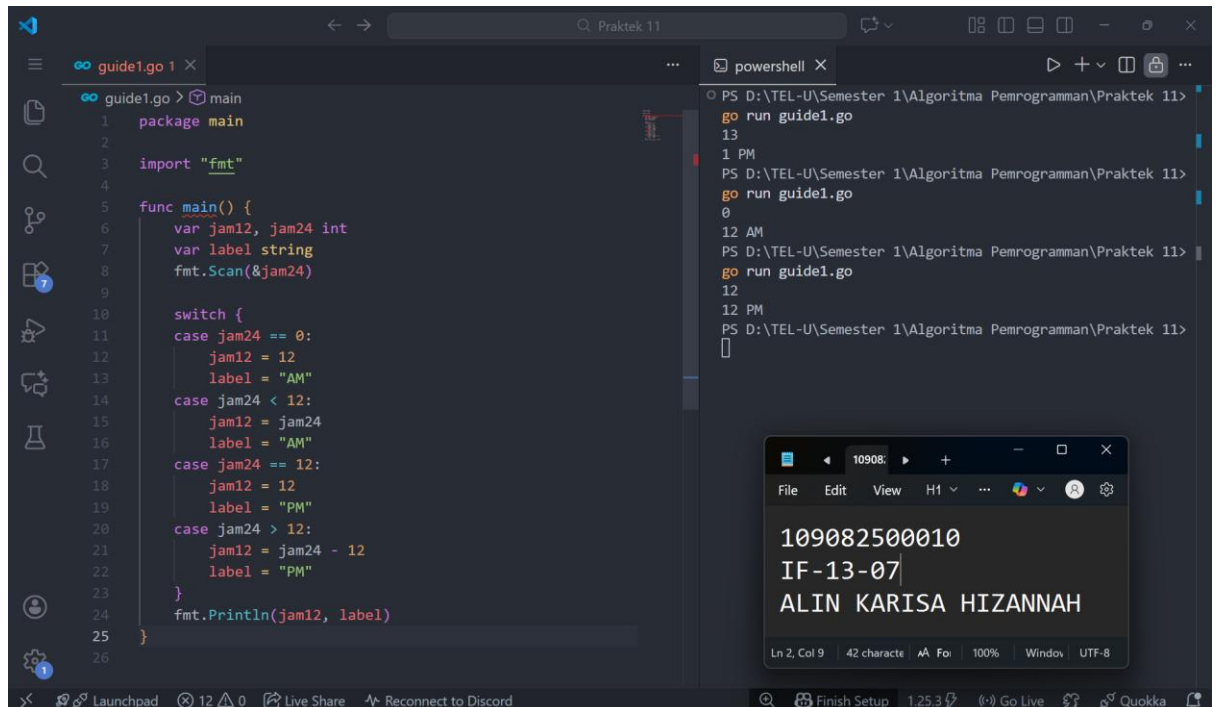
        label = "PM"

    }

    fmt.Println(jam12, label)

}
```

## Screenshoot program



The screenshot shows a Go program in VS Code and its execution in PowerShell. The Go program, named `guide1.go`, is a simple application that takes a 24-hour time input and converts it to a 12-hour format with AM/PM labels. The PowerShell window shows the execution of the program, where the input `109082500010` is processed to output `IF-13-07` and `ALIN KARISA HIZANNAH`.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var jam12, jam24 int
7     var label string
8     fmt.Scan(&jam24)
9
10    switch {
11    case jam24 == 0:
12        jam12 = 12
13        label = "AM"
14    case jam24 < 12:
15        jam12 = jam24
16        label = "AM"
17    case jam24 == 12:
18        jam12 = 12
19        label = "PM"
20    case jam24 > 12:
21        jam12 = jam24 - 12
22        label = "PM"
23    }
24    fmt.Println(jam12, label)
25 }
```

```
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run guide1.go
13
1 PM
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run guide1.go
0
12 AM
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run guide1.go
12
12 PM
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11>
```

109082500010  
IF-13-07  
ALIN KARISA HIZANNAH

## Deskripsi program

Program ini digunakan untuk mengubah format waktu dari 24 jam menjadi format 12 jam lengkap dengan penanda AM/PM. Pengguna hanya perlu memasukkan satu angka berupa jam dalam format 24 jam, kemudian program menentukan jam versi 12 jam berdasarkan kondisi yang dicek di dalam struktur switch.

Jika jam yang dimasukkan adalah 0, program mengubahnya menjadi 12 AM. Jika kurang dari 12, jam tetap sama tetapi diberi label AM. Jika jam bernilai 12, maka hasilnya menjadi 12 PM. Dan jika lebih dari 12, nilai jam dikurangi 12 lalu diberi label PM. Setelah proses konversi selesai, program menampilkan hasil jam dalam format 12 jam beserta label AM atau PM yang sesuai.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var tanaman string

    fmt.Print("Masukkan nama tanaman: ")

    fmt.Scanln(&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**

The screenshot shows a Go program in a file named `guide2.go` and its execution in a PowerShell terminal. The Go program uses a `switch` statement to categorize plants based on their names. The terminal shows three runs of the program with inputs `nepenthes`, `venus`, and `karedok`, resulting in different classification messages.

```
guide2.go 1 x
1 package main
2
3 import "fmt"
4
5 func main() {
6     var tanaman string
7
8     fmt.Print("Masukkan nama tanaman: ")
9     fmt.Scanln(&tanaman)
10
11     switch tanaman {
12     case "nepenthes":
13         fmt.Println("Termasuk Tanaman Karnivora")
14         fmt.Println("Asli Indonesia")
15     case "venus":
16         fmt.Println("Termasuk Tanaman Karnivora")
17         fmt.Println("Bukan Asli Indonesia")
18     default:
19         fmt.Println("Tidak Termasuk Tanaman Karnivora")
20     }
21 }
22
```

```
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11>
go run guide2.go
Masukkan nama tanaman: nepenthes
Termasuk Tanaman Karnivora
Asli Indonesia
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11>
go run guide2.go
Masukkan nama tanaman: venus
Termasuk Tanaman Karnivora
Bukan Asli Indonesia
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11>
go run guide2.go
Masukkan nama tanaman: karedok
Tidak Termasuk Tanaman Karnivora
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11>
```

### Deskripsi program

Program ini digunakan untuk mengelompokkan jenis tanaman berdasarkan nama yang dimasukkan oleh pengguna. Ketika pengguna mengetik sebuah nama tanaman, program akan mencocokkannya dengan beberapa pilihan yang tersedia di dalam struktur `switch`. Jika nama yang dimasukkan adalah "nepenthes", program menampilkan bahwa tanaman tersebut termasuk tanaman karnivora dan merupakan tanaman asli Indonesia.

Jika pengguna memasukkan "venus", program tetap mengidentifikasinya sebagai tanaman karnivora, tetapi menegaskan bahwa tanaman tersebut bukan berasal dari Indonesia. Namun, jika nama tanaman yang diberikan tidak cocok dengan dua pilihan tersebut, program langsung menampilkan bahwa tanaman tersebut bukan tanaman karnivora. Dengan demikian, program memberikan informasi sederhana namun jelas berdasarkan nama tanaman yang diinput.

### 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)

    }

```

## Screenshoot program

The screenshot shows a Go program in a VS Code editor and its execution output in a terminal window.

**Code in VS Code:**

```

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

```

**Terminal Output:**

```

File Edit View A 100% Wind UTF-8
109082500010
IF-13-07
ALIN KARISA HIZANNAH
Ln 3, Col 21 42 character A4 For 100% Wind UTF-8

```

```
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run guide3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000

PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run guide3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000

PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run guide3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir: Rp 9000

PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run guide3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000

PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run guide3.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 D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11>
```

109082500010  
IF-13-07  
ALIN KARISA HIZANNAH  
Ln 2, Col 9 | 42 character | 100% | UTF-8

## Deskripsi program

Program ini berfungsi untuk menghitung tarif parkir berdasarkan jenis kendaraan dan lama durasi parkir yang dimasukkan oleh pengguna. Pertama, pengguna diminta mengetikkan jenis kendaraan—apakah Motor, Mobil, atau Truk—lalu memasukkan berapa jam kendaraan tersebut diparkir. Setelah itu, program mengevaluasi kombinasi keduanya menggunakan struktur switch untuk menentukan tarif yang sesuai.

Setiap jenis kendaraan memiliki tarif berbeda tergantung durasi. Motor dikenakan biaya lebih murah, sedangkan truk memiliki tarif paling tinggi. Untuk durasi 1–2 jam, tarifnya lebih rendah, dan jika durasinya lebih dari 2 jam, tarifnya meningkat. Jika pengguna memasukkan jenis kendaraan yang tidak dikenali atau durasi tidak valid, program akan menampilkan pesan kesalahan. Di akhir proses, tarif parkir yang diperoleh akan ditampilkan dalam format “Rp ...”.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {

    var ph float64

    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 image consists of two screenshots of a Visual Studio Code editor window. The top screenshot shows the source code of a Go program named `soal1.go`. The code defines a `main` function that reads a pH value from the command line and uses a `switch` statement to validate it. The bottom screenshot shows the same editor with a PowerShell terminal open, displaying the execution of the program for three different inputs: 8.6, 9, and 16. A small text box in the bottom right of both screenshots contains the text: 109082500010, IF-13-07, ALIN KARISA HIZANNAH.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float64
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
```

PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 11> go run soal1.go 8.6  
Air layak minum  
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 11> go run soal1.go 9  
Air tidak layak minum  
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 11> go run soal1.go 16  
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.  
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 11>

109082500010  
IF-13-07  
ALIN KARISA HIZANNAH

## Deskripsi program

Program ini digunakan untuk mengevaluasi kelayakan air minum berdasarkan nilai pH yang dimasukkan pengguna. Setelah pengguna memasukkan angka pH, program langsung memeriksa apakah angka tersebut berada dalam rentang yang valid, yaitu antara 0 hingga 14. Jika nilainya berada di luar batas tersebut, program akan menampilkan pesan bahwa nilai pH tidak valid.

Jika nilai pH masih dalam rentang yang benar, program lanjut mengecek apakah pH tersebut berada pada kisaran 6.5 sampai 8.6, yaitu standar umum air yang aman untuk diminum. Bila pH berada dalam rentang tersebut, program menyatakan air layak minum. Namun, jika berada di luar rentang aman, program menandai bahwa air tersebut tidak layak minum. Dengan demikian, program memberikan penilaian sederhana namun cukup akurat berdasarkan input nilai pH.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var kendaraan string

    var durasi int

    var tarif int

    fmt.Scan(&kendaraan, &durasi)


    if durasi < 1 {

        durasi = 1

    }


    switch kendaraan {

    case "motor":

        tarif = 2000

    case "mobil":

        tarif = 5000

    }
```

```

        case "truk":

            tarif = 8000

        }

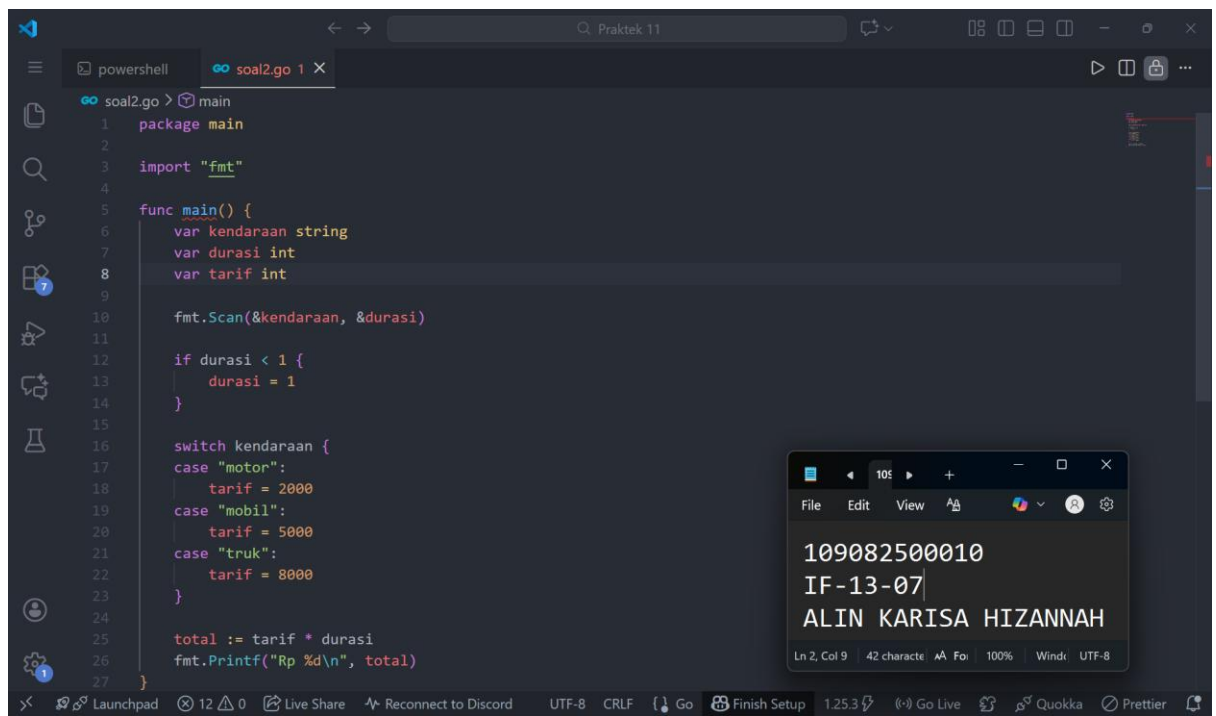
        total := tarif * durasi

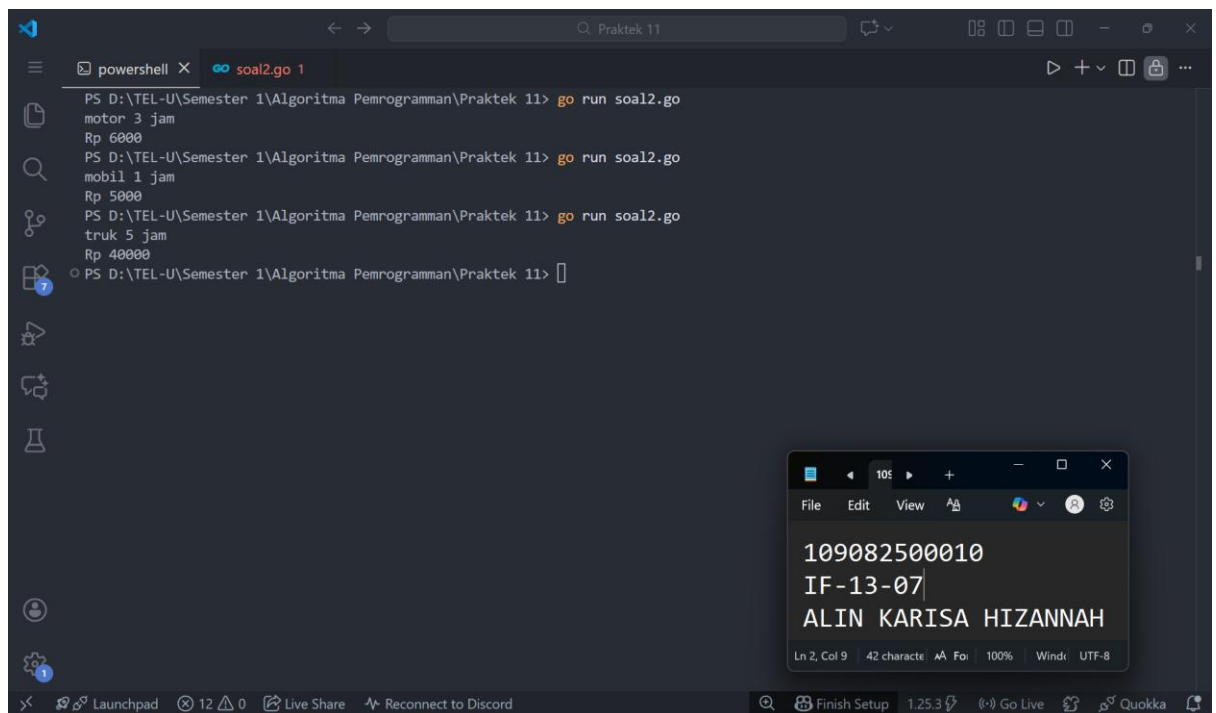
        fmt.Printf("Rp %d\n", total)

    }

```

## Screenshoot program





```
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run soal2.go
motor 3 jam
Rp 6000
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run soal2.go
mobil 1 jam
Rp 5000
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run soal2.go
truk 5 jam
Rp 40000
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11>
```

```
109082500010
IF-13-07
ALIN KARISA HIZANNAH
```

### Deskripsi program

Program ini menghitung total biaya parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Pengguna hanya perlu mengetikkan jenis kendaraan (motor, mobil, atau truk) beserta lama parkir dalam jam. Jika durasi yang dimasukkan kurang dari satu jam, program otomatis menetapkan durasi minimum menjadi satu jam agar perhitungan tetap masuk akal.

Setelah itu, program menentukan tarif per jam sesuai dengan jenis kendaraan. Motor dikenakan tarif 2000 rupiah per jam, mobil dikenakan tarif 5000 rupiah per jam, dan truk dikenakan tarif 8000 rupiah per jam. Total biaya parkir dihitung dengan mengalikan tarif tersebut dengan durasi parkir. Hasil perhitungan kemudian ditampilkan dalam format "Rp ...", sehingga pengguna dapat mengetahui jumlah biaya yang harus dibayar.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"
```

```
func main() {  
  
    var input int  
  
    fmt.Scan(&input)  
  
    fmt.Print("Kategori: ")  
  
    switch {  
  
    case input%10 == 0:  
  
        fmt.Println("Bilangan Kelipatan 10")  
  
        hasil := input / 10  
  
        fmt.Printf("Hasil pembagian antara %d / 10 =  
%d\n", input, hasil)  
  
    case input%5 == 0 && input != 5:  
  
        fmt.Println("Bilangan Kelipatan 5")  
  
        hasil := input * input  
  
        fmt.Printf("Hasil kuadrat dari %d^2 = %d\n",  
input, hasil)  
  
    case input%2 == 0:  
  
        fmt.Println("Bilangan Genap")  
  
        hasil := input * (input + 1)  
  
        fmt.Printf("Hasil perkalian dengan bilangan  
berikutnya %d * %d = %d\n",  
            input, input+1, hasil)  
  
    case input%2 == 1:  
  
        fmt.Println("Bilangan Ganjil")  
  
    }
```

```

        hasil := input + (input + 1)

        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n",

        input, input+1, hasil)

    }

}

```

## Screenshoot program

The screenshot shows a Go IDE with a file named `soal3.go`. The program uses a `switch` statement to handle different input cases. The output window shows the results of the program execution for the input `109082500010`.

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var input int
7     fmt.Scan(&input)
8     fmt.Print("Kategori: ")
9
10    switch {
11    case input%10 == 0:
12        fmt.Println("Bilangan Kelipatan 10")
13        hasil := input / 10
14        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", input, hasil)
15
16    case input%5 == 0 && input != 5:
17        fmt.Println("Bilangan Kelipatan 5")
18        hasil := input * input
19        fmt.Printf("Hasil kuadrat dari %d^2 = %d\n", input, hasil)
20
21    case input%2 == 0:
22        fmt.Println("Bilangan Genap")
23        hasil := input * (input + 1)
24        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n",
25            input, input+1, hasil)
26
27    case input%2 == 1:
28        fmt.Println("Bilangan Ganjil")
29        hasil := input + (input + 1)
30        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n",
31            input, input+1, hasil)
32    }
33 }

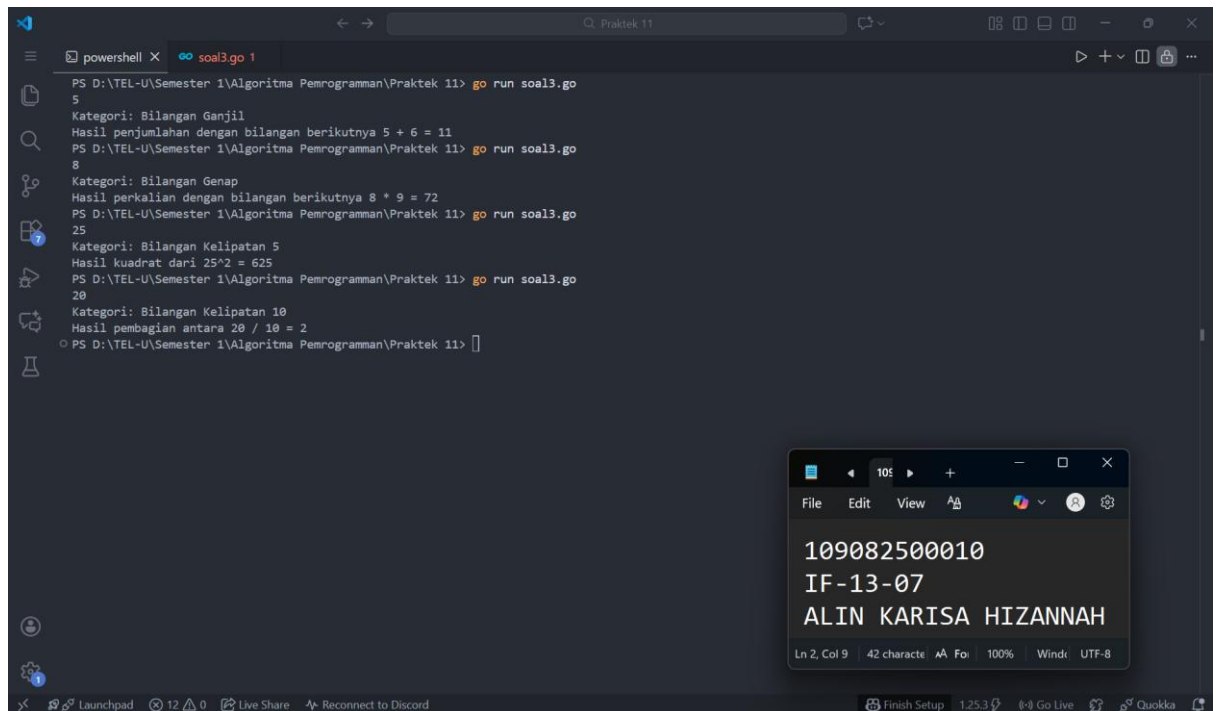
```

Output window content:

```

109082500010
IF-13-07
ALIN KARISA HIZANNAH
Ln 2, Col 9 | 42 character | 100% | Windi | UTF-8

```



The screenshot shows a VS Code interface with a terminal window and a Notepad window. The terminal window, titled 'powershell X', shows the execution of a Go program named 'soal3.go'. The program prompts for a number and then outputs a category and a result based on the number. The Notepad window shows the output of the program for the number 109082500010, which is categorized as 'IF-13-07' and 'ALIN KARISA HIZANNAH'.

```
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run soal3.go
5
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run soal3.go
8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run soal3.go
25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25^2 = 625
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11> go run soal3.go
20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS D:\TEL-U\Semester 1\Algoritma Pemrogramman\Praktek 11>
```

109082500010  
IF-13-07  
ALIN KARISA HIZANNAH

## Deskripsi program

Program ini bekerja dengan membaca sebuah bilangan bulat, lalu menentukan kategori bilangan tersebut berdasarkan aturan tertentu. Aturan kategori dibuat berurutan di dalam switch, sehingga ketika sebuah kondisi terpenuhi, program langsung menjalankan operasi yang sesuai tanpa mengecek kondisi lainnya. Dengan cara ini, setiap bilangan hanya akan masuk ke satu kategori.

Untuk bilangan ganjil, program menghitung penjumlahan antara bilangan tersebut dengan bilangan setelahnya. Untuk bilangan genap (yang bukan kelipatan 5 atau 10), program menghitung hasil perkalian bilangan itu dengan bilangan berikutnya. Jika bilangan tersebut merupakan kelipatan 5 tetapi bukan kelipatan 10, program menghitung kuadrat dari angka tersebut. Sedangkan jika bilangan tersebut kelipatan 10, program akan membaginya dengan 10.

Kategori dengan prioritas tertinggi adalah kelipatan 10, kemudian kelipatan 5, lalu bilangan genap, dan terakhir bilangan ganjil. Susunan prioritas ini penting agar sebuah bilangan yang termasuk lebih dari satu kategori tetap diproses sesuai ketentuan utama. Misalnya angka 20 adalah genap dan juga kelipatan 10, tetapi karena kondisi kelipatan

10 ditempatkan lebih dulu, maka angka 20 diproses sebagai bilangan kelipatan 10. Dengan begitu, hasil program selalu sesuai aturan yang sudah ditentukan.