

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 11
SWITCH CASE**



Disusun oleh:
BENING PUTRI NARESWARI SUKARNO
109082500211
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

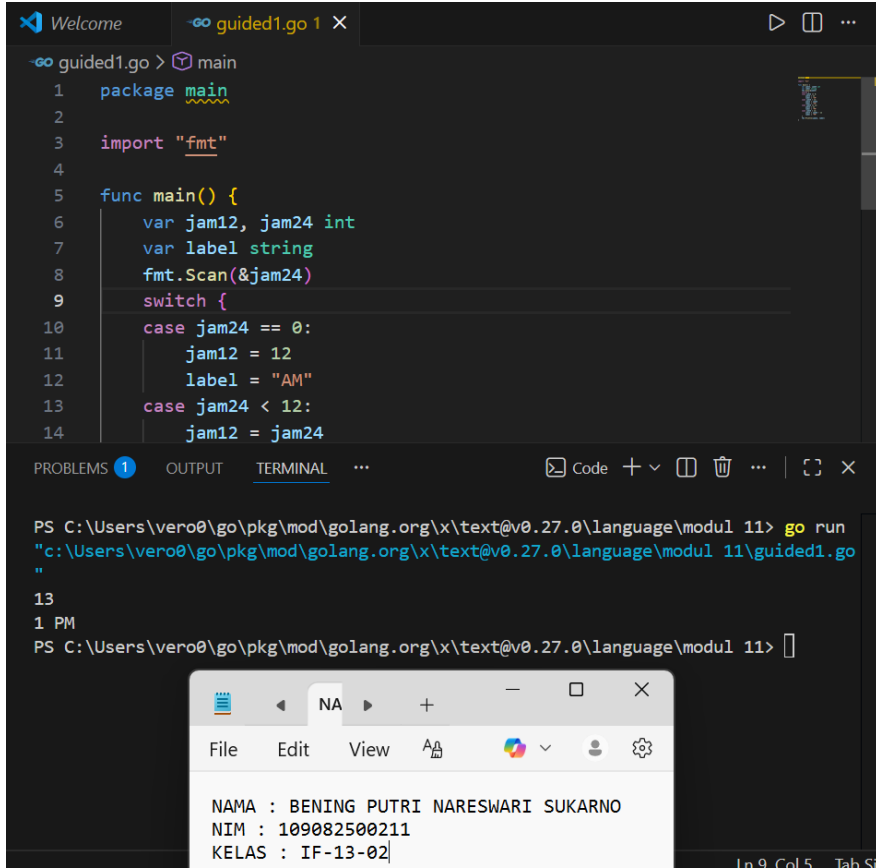
        label = "PM"

    }

    fmt.Println(jam12, label)
```

```
}
```

Screenshoot program



```
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     switch {
10    case jam24 == 0:
11        jam12 = 12
12        label = "AM"
13    case jam24 < 12:
14        jam12 = jam24
```

```
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11> go run
"c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11\guided1.go"
13
1 PM
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11>
```

NAMA : BENING PUTRI NARESWARI SUKARNO
NIM : 109082500211
KELAS : IF-13-02

Deskripsi program

Program ini berfungsi untuk mengkonversi waktu dari format 24 jam ke format 12 jam.

Cara Kerja Program:

1. Program meminta Anda memasukkan satu angka, yaitu jam dalam format 24 jam (jam24) yang berupa bilangan bulat (dari 0 hingga 23).
2. Program menggunakan blok switch untuk menentukan nilai jam dalam format 12 jam (jam12) dan label waktu (AM atau PM) berdasarkan aturan berikut:
 - Pukul 00:00 (Tengah Malam): Menjadi 12 AM.
 - Jam 1 hingga 11: Tetap sama, dan berlabel AM.
 - Pukul 12:00 (Siang): Menjadi 12 PM.
 - Jam 13 hingga 23: Jam dikurangi 12, dan berlabel PM.
3. Program akan mencetak hasil konversi berupa jam (jam12) diikuti dengan label (AM atau PM).

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.")

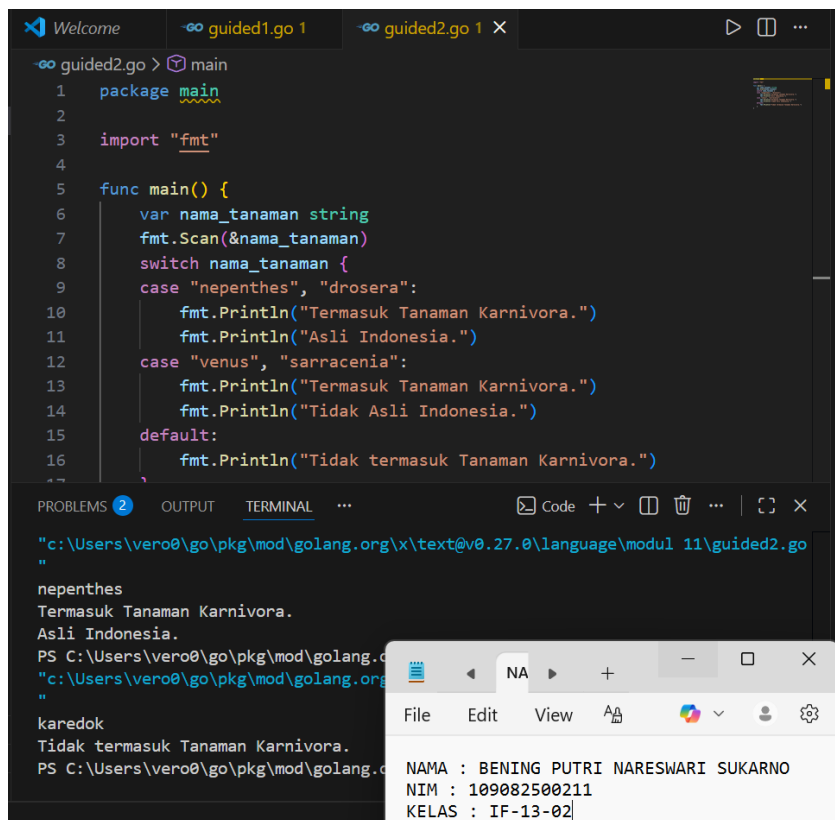
    default:

        fmt.Println("Tidak termasuk Tanaman Karnivora.")

    }

}
```

Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var nama_tanaman string
7     fmt.Scan(&nama_tanaman)
8     switch nama_tanaman {
9     case "nepenthes", "drosera":
10         fmt.Println("Termasuk Tanaman Karnivora.")
11         fmt.Println("Asli Indonesia.")
12     case "venus", "sarracenia":
13         fmt.Println("Termasuk Tanaman Karnivora.")
14         fmt.Println("Tidak Asli Indonesia.")
15     default:
16         fmt.Println("Tidak termasuk Tanaman Karnivora.")
17     }
18 }
```

PROBLEMS 2 OUTPUT TERMINAL ...

"c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11\guided2.go"

nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.

PS C:\Users\vero0\go\pkg\mod\golang.org>

"c:\Users\vero0\go\pkg\mod\golang.org"

karedok
Tidak termasuk Tanaman Karnivora.

PS C:\Users\vero0\go\pkg\mod\golang.org>

NA

File Edit View A A + - □ ×

NAMA : BENING PUTRI NARESWARI SUKARNO
NIM : 109082500211
KELAS : IF-13-02

Deskripsi program

Program ini berfungsi untuk mengklasifikasikan jenis tanaman yang dimasukkan pengguna, menentukan apakah tanaman tersebut Karnivora (pemakan serangga) atau bukan, dan sekaligus menentukan apakah jenis tersebut Asli Indonesia atau bukan.

Cara Kerja Program:

1. Program meminta Anda memasukkan nama satu jenis tanaman.
2. Program menggunakan blok switch yang memeriksa nama_tanaman terhadap beberapa kasus yang telah ditentukan:
 - Kasus 1 (nepenthes atau drosera): Jika tanaman cocok dengan salah satu nama ini, program mencetak bahwa tanaman itu Termasuk Tanaman Karnivora dan Asli Indonesia.
 - Kasus 2 (venus atau sarracenia): Jika tanaman cocok dengan salah satu nama ini, program mencetak bahwa tanaman itu Termasuk Tanaman Karnivora, tetapi Tidak Asli Indonesia.
 - Kasus default (Lainnya): Jika nama tanaman yang dimasukkan tidak cocok dengan semua kasus di atas (misalnya "karedok" atau "mawar"), program mencetak bahwa tanaman itu Tidak termasuk Tanaman Karnivora.
3. Output (Keluaran): Program menampilkan deskripsi hasil klasifikasi berdasarkan nama tanaman yang dimasukkan.

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 image displays a Go program in a code editor and its execution in a terminal window.

Go Program Code:

```
guided3.go > main
5 func main() {
10     fmt.Scan(&kendaraan)
11     fmt.Print("Masukkan durasi parkir (dalam jam): ")
12     fmt.Scan(&durasi)
13     switch {
14     case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
15         tarif = 7000
16     case kendaraan == "Motor" && durasi > 2:
17         tarif = 9000
18     case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
19         tarif = 15000
20     case kendaraan == "Mobil" && durasi > 2:
21         tarif = 20000
22     case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
23         tarif = 25000
24     case kendaraan == "Truk" && durasi > 2:
```

Terminal Output:

```
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11> go run
"c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11\guided3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 5
Tarif Parkir: Rp 9000
PS C:\Users\vero0\go\pkg\mod\golang.org
```

The terminal window also shows a file explorer overlay with the following details:

- File: NAMA
- Edit
- View
- Font: AA
- Color: 16-color palette
- User: User icon
- Settings: Gear icon
- Content: NAMA : BENING PUTRI NARESWARI SUKARNO
NIM : 109082500211
KELAS : TE-13-07

Deskripsi program

Program ini berfungsi untuk menghitung total tarif parkir berdasarkan jenis kendaraan dan durasi parkir dalam jam. Tarif yang dikenakan bersifat bertingkat (lebih dari 2 jam tarifnya berbeda).

Cara Kerja Program:

1. Program meminta dua input dari pengguna:
Jenis Kendaraan: Harus berupa teks yang sesuai ("Motor", "Mobil", atau "Truk").
Durasi Parkir: Berupa angka bulat dalam jam (misalnya 1, 2, 5).
2. Logika Penentuan Tarif: Program menggunakan blok switch dengan beberapa case yang menguji dua kondisi sekaligus (&&): jenis kendaraan DAN rentang durasi.
 - Durasi 1-2 Jam: Dikenakan tarif dasar.
 - Durasi > 2 Jam: Dikenakan tarif yang lebih mahal.
 - Kendaraan dan Durasi Tidak Cocok (default): Jika input tidak sesuai dengan skema (misalnya, jenis kendaraan bukan "Motor", "Mobil", atau "Truk"), tarif diatur ke 0 (atau menampilkan pesan error pada blok default).
3. Program akan mencetak total biaya parkir (Tarif Parkir: Rp X).

TUGAS

1. Tugas 1

Source code

```
package main

import (
    "fmt"
)

func main() {
    var ph float64
    var hasil string

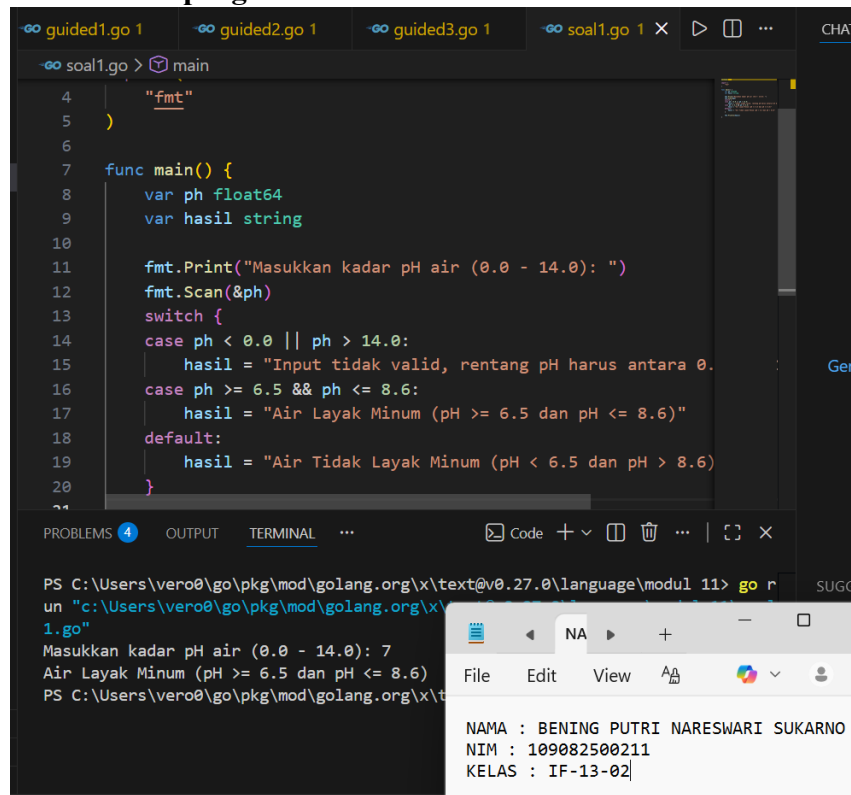
    fmt.Print("Masukkan kadar pH air (0.0 - 14.0): ")
    fmt.Scan(&ph)

    switch {
    case ph < 0.0 || ph > 14.0:
        hasil = "Input tidak valid, rentang pH harus antara 0.0 dan 14.0."
    case ph >= 6.5 && ph <= 8.6:
        hasil = "Air Layak Minum (pH >= 6.5 dan pH <= 8.6)"
    default:
        hasil = "Air Tidak Layak Minum (pH < 6.5 dan pH > 8.6)"
    }

    fmt.Println(hasil)
```

```
}
```

Screenshoot program



```
4     "fmt"
5 )
6
7 func main() {
8     var ph float64
9     var hasil string
10
11     fmt.Print("Masukkan kadar pH air (0.0 - 14.0): ")
12     fmt.Scan(&ph)
13     switch {
14     case ph < 0.0 || ph > 14.0:
15         hasil = "Input tidak valid, rentang pH harus antara 0."
16     case ph >= 6.5 && ph <= 8.6:
17         hasil = "Air Layak Minum (pH >= 6.5 dan pH <= 8.6)"
18     default:
19         hasil = "Air Tidak Layak Minum (pH < 6.5 dan pH > 8.6)"
20     }
21 }
```

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11> go r
un "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11>
1.go"
Masukkan kadar pH air (0.0 - 14.0): 7
Air Layak Minum (pH >= 6.5 dan pH <= 8.6)
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11>

Deskripsi program

Program ini berfungsi untuk menentukan apakah air layak minum atau tidak berdasarkan nilai kadar pH yang dimasukkan oleh pengguna. Program ini mengikuti standar kesehatan tertentu untuk menilai air.

Cara Kerja Program:

1. Input (Masukan): Program meminta Anda memasukkan satu angka desimal (float64) yang mewakili kadar pH air (nilai idealnya antara 0.0 hingga 14.0).
2. Logika Penilaian (switch case): Program menggunakan blok switch tanpa variabel eksplisit (switch {}) yang berfungsi seperti serangkaian if-else untuk menguji kondisi pH:
 - Kasus 1 (Input Tidak Valid): Jika nilai pH di bawah 0.0 atau di atas 14.0, maka input dianggap tidak sah/tidak valid.
 - Kasus 2 (Air Layak Minum): Jika nilai pH berada dalam rentang 6.5 hingga 8.6 (inklusif), maka air diklasifikasikan sebagai Air Layak Minum.
 - Kasus default (Air Tidak Layak Minum): Jika nilai pH berada dalam rentang valid (0.0 - 14.0) tetapi di luar rentang layak minum (misalnya pH 4.0 atau pH 10.0), maka air diklasifikasikan sebagai Air Tidak Layak Minum.
3. Output (Keluaran): Program akan mencetak hasil klasifikasi air (Layak Minum, Tidak Layak Minum, atau Input Tidak Valid).

2. Tugas 2

Source code

```
package main

import (
    "fmt"
)

func main() {
    var jenis string
    var durasi float64
    var tarif int

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

    fmt.Print("Durasi parkir (jam, cth 3.5): ")
    fmt.Scan(&durasi)

    switch jenis {
    case "motor":
        tarif = 2000
    case "mobil":
        tarif = 5000
    case "truk":
        tarif = 8000
    default:
```

```
        tarif = 0

    }

    if tarif > 0 {

        if durasi < 1.0 {

            durasi = 1.0

        }

        jamDikenakanBiaya := int(durasi)

        if durasi > float64(jamDikenakanBiaya) {

            jamDikenakanBiaya++

        }

        totalBiaya := jamDikenakanBiaya * tarif

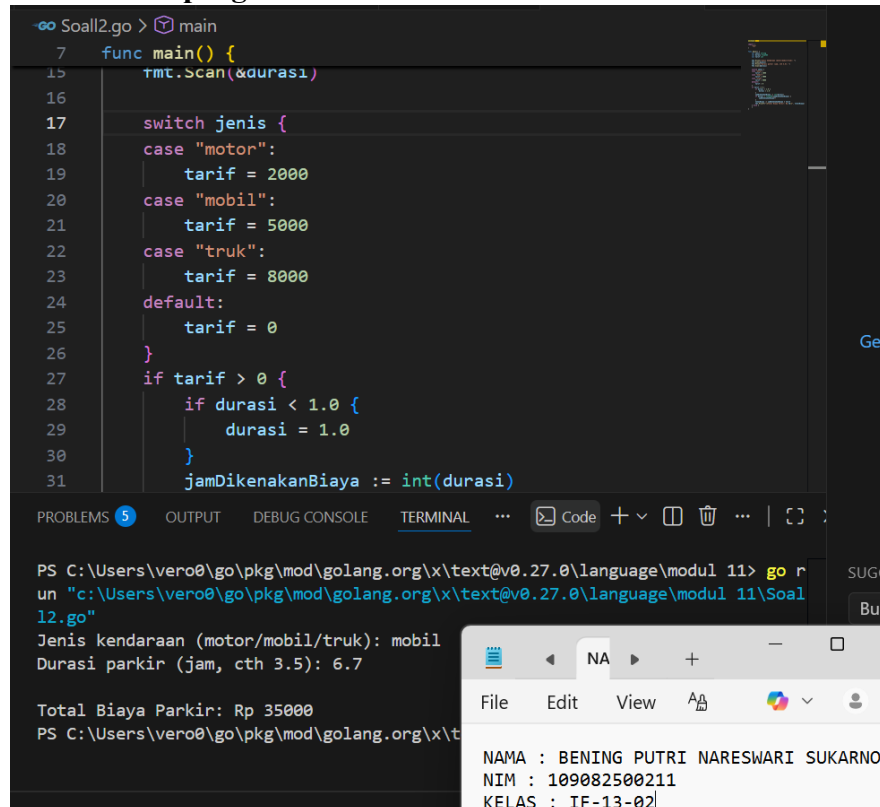
        fmt.Printf("\nTotal Biaya Parkir: Rp %d\n",
totalBiaya)

    } else {

    }

}
```

Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is named `Soal12.go` and is located in the `main` package. The code defines a `main` function that takes a `jenis` string and a `durasi` float as input. It uses a `switch` statement to determine the parking rate (`tarif`) based on the vehicle type: `motor` (2000), `mobil` (5000), and `truk` (8000). If the vehicle type is not specified, the rate is 0. The program also checks if the duration is less than 1.0 and rounds it up to 1.0. Finally, it calculates the total parking fee (`jamDikenakanBiaya`) by multiplying the rate by the duration and prints it.

```
7 func main() {
15     rmt.Scan(&durasi)
16
17     switch jenis {
18     case "motor":
19         tarif = 2000
20     case "mobil":
21         tarif = 5000
22     case "truk":
23         tarif = 8000
24     default:
25         tarif = 0
26     }
27     if tarif > 0 {
28         if durasi < 1.0 {
29             durasi = 1.0
30         }
31     }
32     jamDikenakanBiaya := int(durasi * tarif)
33     fmt.Println("Total Biaya Parkir: Rp", jamDikenakanBiaya)
34 }
```

The terminal output shows the program being run from the command prompt. The user enters `mobil` for the vehicle type and `3.5` for the duration. The program outputs the total parking fee as `Rp 35000`.

```
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11> go run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11\Soal12.go"
Jenis kendaraan (motor/mobil/truk): mobil
Durasi parkir (jam, cth 3.5): 6.7

Total Biaya Parkir: Rp 35000
PS C:\Users\vero0\go\pkg\mod\golang.org\x\t
```

Deskripsi program

Program ini berfungsi untuk menghitung total biaya parkir berdasarkan jenis kendaraan dan durasi parkir

Cara Kerja:

1. Program meminta Anda memasukkan dua hal: jenis kendaraan (motor, mobil, atau truk) dan durasi parkir (dalam jam, bisa desimal seperti 3.5).
2. Program menggunakan switch untuk menentukan tarif per jam:
 - motor = Rp 2.000/jam
 - mobil = Rp 5.000/jam
 - truk = Rp 8.000/jam
 - Jika jenis kendaraan salah, tarif diatur menjadi Rp 0.
 - Perhitungan Durasi: Jika tarifnya valid (bukan Rp 0), program melakukan penyesuaian durasi:
 - Minimum 1 Jam: Jika Anda parkir kurang dari 1 jam (misalnya 0.5 jam), program akan menghitungnya sebagai 1 jam.
 - Pembulatan ke Atas: Durasi selalu dibulatkan ke jam penuh berikutnya (misalnya, 3.5 jam akan dibulatkan menjadi 4 jam).
3. Total biaya parkir dihitung dengan mengalikan durasi jam yang dibulatkan dengan tarif per jam, lalu dicetak ke terminal. Jika jenis kendaraan salah, tidak ada biaya yang dicetak.

3. Tugas 3

Source code

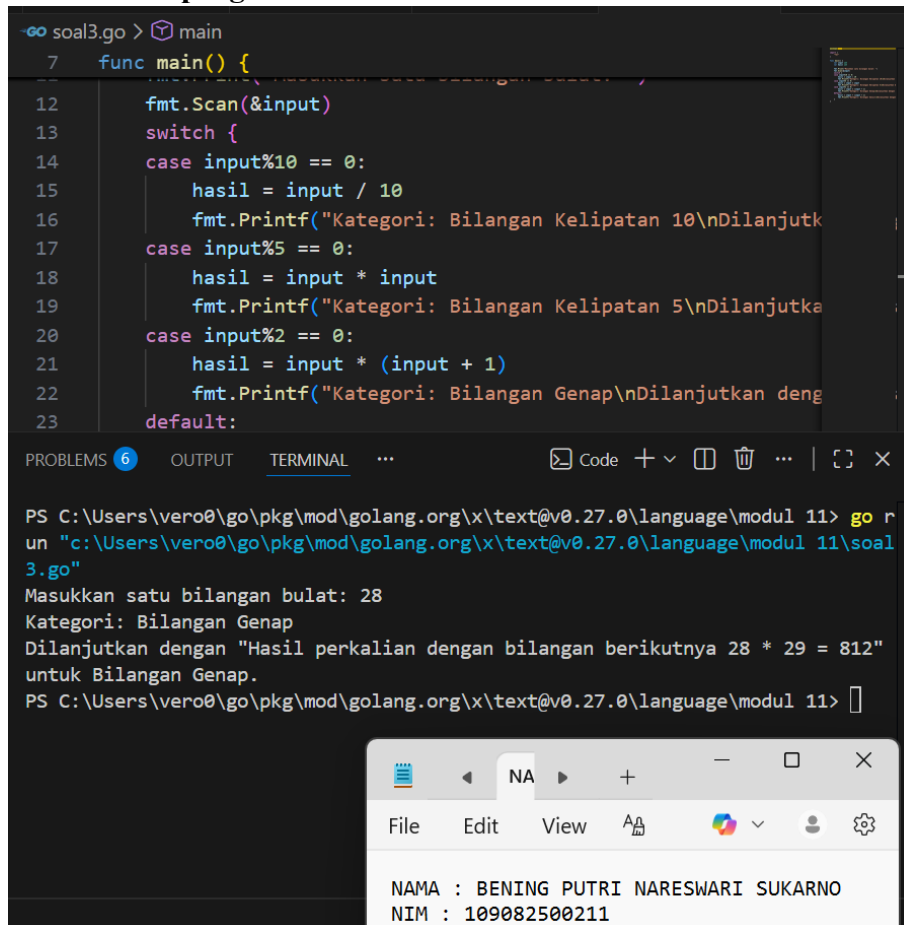
```
package main

import (
    "fmt"
)

func main() {
    var input int
    var hasil int

    fmt.Print("Masukkan satu bilangan bulat: ")
    fmt.Scan(&input)
    switch {
    case input%10 == 0:
        hasil = input / 10
        fmt.Printf("Kategori: Bilangan Kelipatan
10\nDilanjutkan dengan \"Hasil pembagian antara %d / 10 =
%d\" untuk Bilangan Kelipatan 10.\n", input, hasil)
    case input%5 == 0:
        hasil = input * input
        fmt.Printf("Kategori: Bilangan Kelipatan
5\nDilanjutkan dengan \"Hasil kuadrat dari %d ^ 2 = %d\"
untuk Bilangan Kelipatan 5.\n", input, hasil)
    case input%2 == 0:
        hasil = input * (input + 1)
        fmt.Printf("Kategori: Bilangan Genap\nDilanjutkan
dengan \"Hasil perkalian dengan bilangan berikutnya %d * %d =
%d\" untuk Bilangan Genap.\n", input, input+1, hasil)
    default:
        hasil = input + (input + 1)
        fmt.Printf("Kategori: Bilangan Ganjil\nDilanjutkan
dengan \"Hasil penjumlahan dengan bilangan berikutnya %d + %d
= %d\" untuk Bilangan Ganjil.\n", input, input+1, hasil)
    }
}
```

Screenshoot program



```
soal3.go > main
7 func main() {
12     fmt.Scan(&input)
13     switch {
14     case input%10 == 0:
15         hasil = input / 10
16         fmt.Printf("Kategori: Bilangan Kelipatan 10\nDilanjutk
17     case input%5 == 0:
18         hasil = input * input
19         fmt.Printf("Kategori: Bilangan Kelipatan 5\nDilanjutka
20     case input%2 == 0:
21         hasil = input * (input + 1)
22         fmt.Printf("Kategori: Bilangan Genap\nDilanjutkan deng
23     default:
```

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11> go r
un "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11\soal
3.go"

Masukkan satu bilangan bulat: 28
Kategori: Bilangan Genap
Dilanjutkan dengan "Hasil perkalian dengan bilangan berikutnya 28 * 29 = 812"
untuk Bilangan Genap.
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 11>

NAMA : BENING PUTRI NARESWARI SUKARNO
NIM : 109082500211

Deskripsi program

Program ini berfungsi untuk mengidentifikasi pola aritmatika (Kelipatan 10, kelipatan 5, genap, atau ganjil) dari sebuah bilangan bulat yang dimasukkan, lalu melakukan operasi matematika spesifik sesuai dengan pola tersebut.

Cara Kerja:

1. Program meminta Anda memasukkan satu bilangan bulat.
2. Program menggunakan switch case untuk memeriksa pola bilangan dengan urutan prioritas yang ketat (menggunakan operator modulo %):
 - Kelipatan 10: Jika habis dibagi 10, dibagi 10.
 - Kelipatan 5: Jika habis dibagi 5 (tetapi bukan kelipatan 10), dikuadratkan.
 - Genap: Jika habis dibagi 2 (tetapi bukan kelipatan 10 atau 5), dikalikan dengan bilangan berikutnya.
 - Ganjil (default): Jika tidak termasuk kategori di atas, dijumlahkan dengan bilangan berikutnya.
3. Program mencetak kategori bilangan dan hasil operasi yang sesuai.

Contoh Masukkan:

1. Pengguna memasukkan bilangan 28
2. Program akan mengidentifikasi bilangan tersebut sebagai Bilangan Genap (karena 28 habis dibagi 2 tetapi tidak habis dibagi 10 atau 5).
3. Program kemudian akan melakukan operasi perkalian antara 28 dengan bilangan berikutnya (29), dan mencetak Hasil perkalian dengan bilangan berikutnya $28 * 29 = 812$ untuk Bilangan Genap.