

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 11
SWITCH-CASE**



Disusun oleh:

Anindya Rahadita Yumnaa

109082500138

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 jam24 int

    fmt.Print("Masukkan jam dalam format 24 jam: ")

    fmt.Scan(&jam24)

    var period string

    var jam12 int

    if jam24 == 0 {

        jam12 = 12

        period = "AM"

    } else if jam24 == 12 {

        jam12 = 12

        period = "PM"

    } else if jam24 > 12 {

        jam12 = jam24 - 12

        period = "PM"

    } else {

        jam12 = jam24

    }
```

```

        period = "AM"

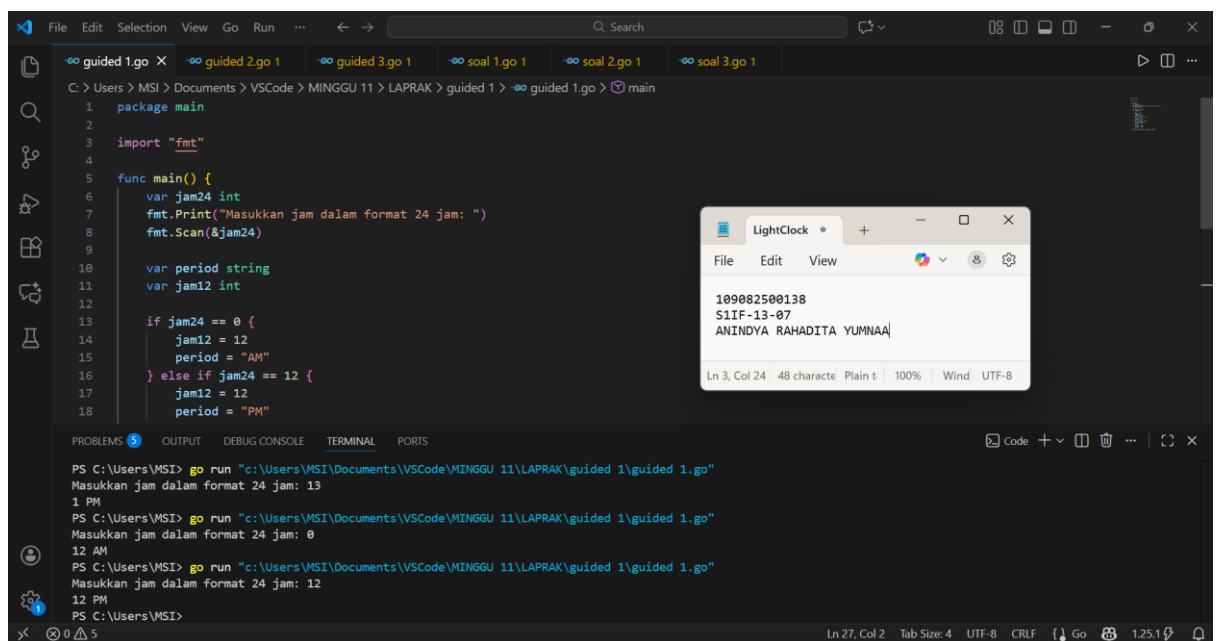
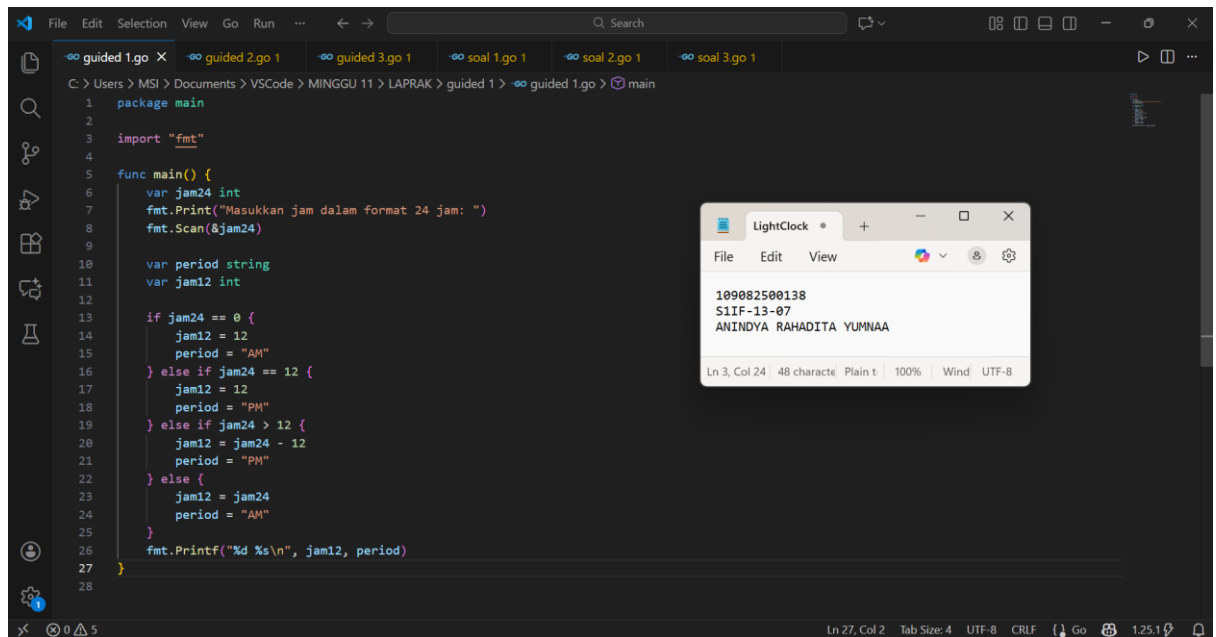
    }

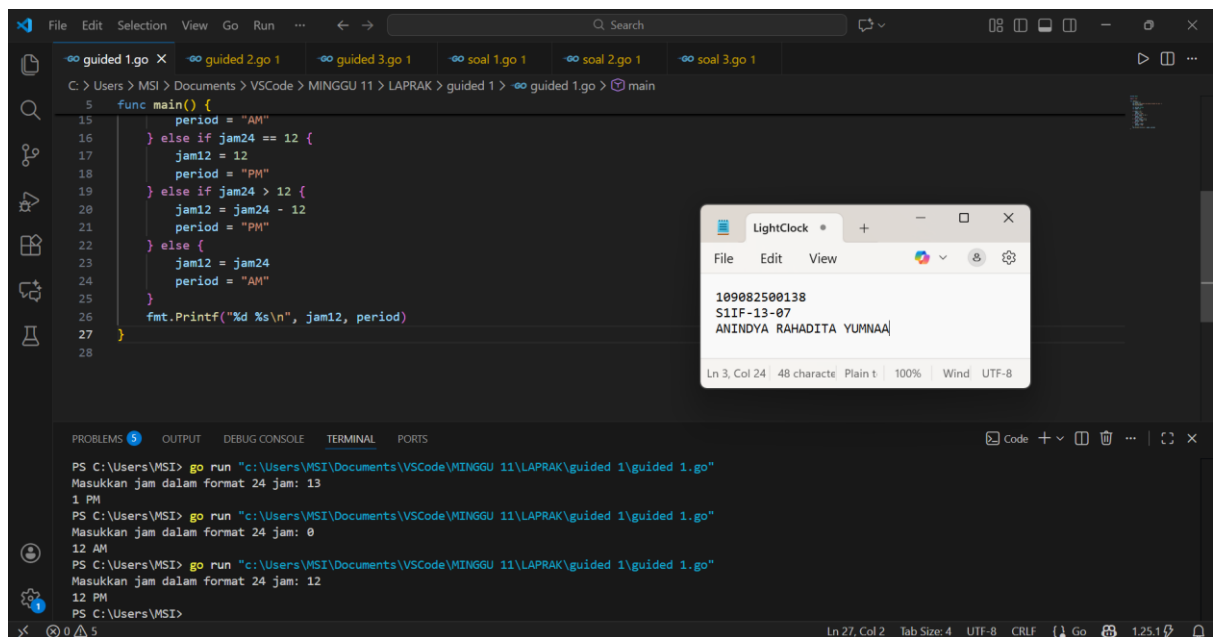
    fmt.Printf("%d %s\n", jam12, period)

}

```

Screenshoot program





Deskripsi program

Program Go ini memiliki fungsi utama untuk **mengubah format waktu dari 24 jam menjadi format 12 jam yang dilengkapi dengan keterangan AM atau PM.**

Pada awalnya, program akan meminta pengguna untuk memasukkan angka jam dalam format 24 jam (mulai dari 0 hingga 23). Setelah menerima input, program akan memproses angka tersebut menggunakan serangkaian logika kondisional.

Inti dari program ini adalah proses konversi:

- Jika jam yang dimasukkan adalah **0 (tengah malam)**, program akan mengubahnya menjadi **12 AM**.
- Jika jam adalah **12 (tengah hari)**, program akan menampilkannya sebagai **12 PM**.
- Untuk jam yang **lebih dari 12 (misalnya jam 13 hingga 23)**, program akan mengurangnya dengan 12 dan memberikan keterangan **PM**. Contohnya, jam 17 akan dikonversi menjadi 5 PM.
- Sedangkan jam dari **1 hingga 11** akan tetap sama dan langsung diberikan keterangan **AM**.

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.")

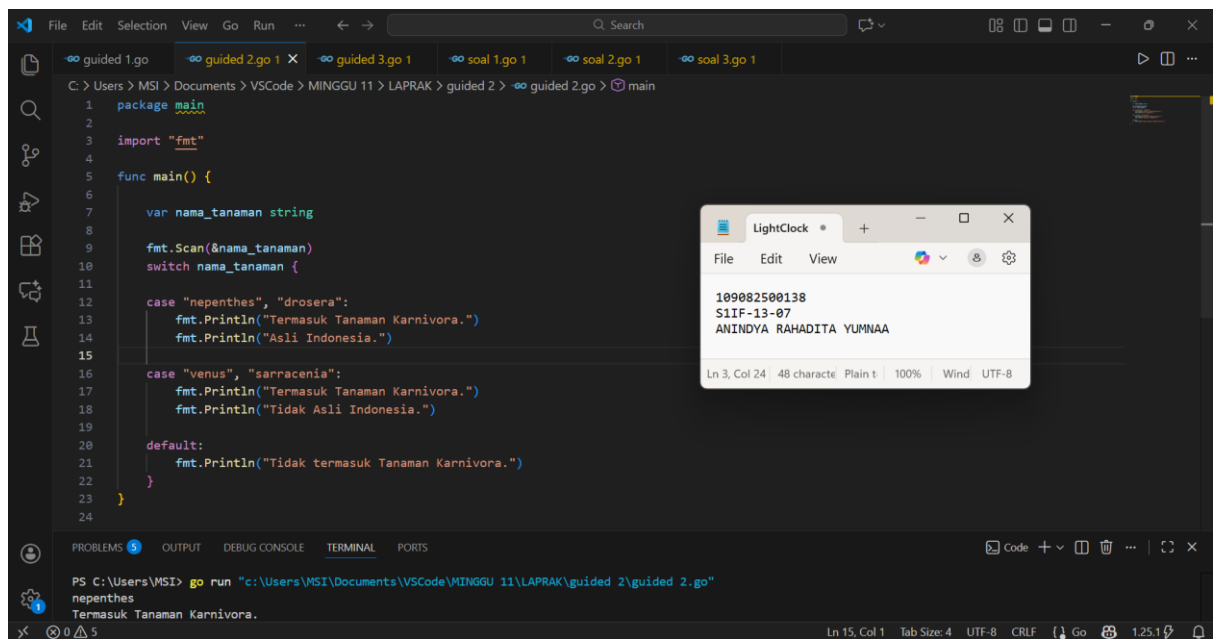
    fmt.Println("Tidak Asli Indonesia.")

default:

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

}
```

Screenshoot program

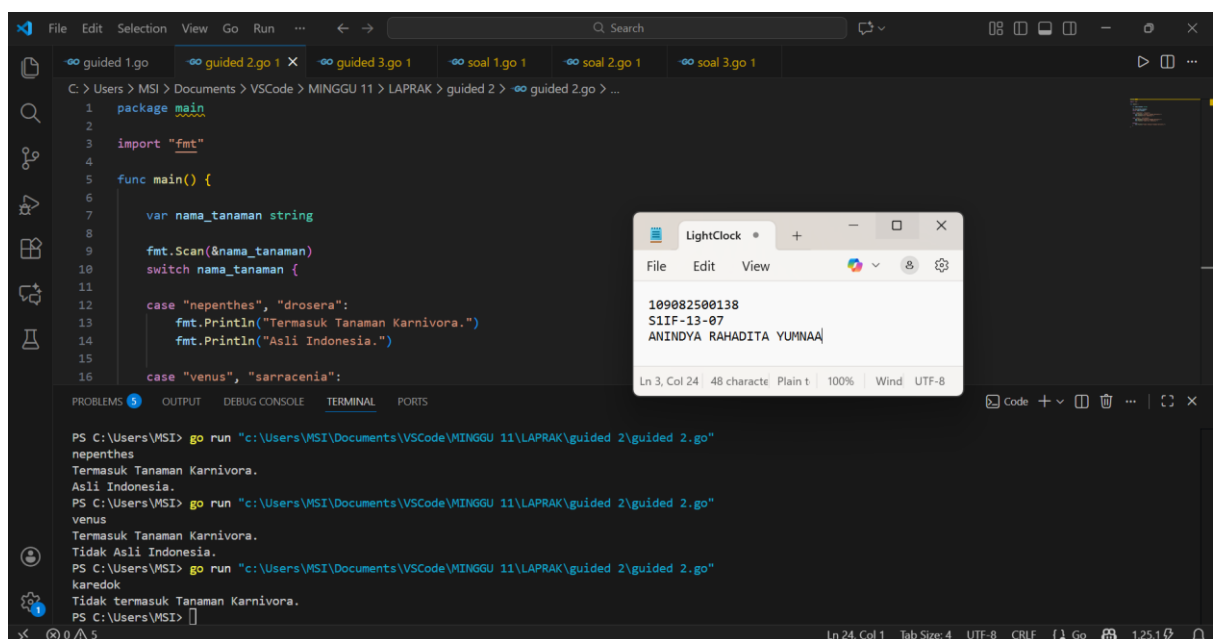


The screenshot shows the Visual Studio Code editor with a Go file named `guided 2.go`. The code defines a `main` function that takes a string `nama_tanaman` and uses a `switch` statement to classify it. The terminal output shows the program running and printing the classification for the input `nepenthes`.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var nama_tanaman string
8
9     fmt.Scan(&nama_tanaman)
10    switch nama_tanaman {
11
12        case "nepenthes", "drosera":
13            fmt.Println("Termasuk Tanaman Karnivora.")
14            fmt.Println("Asli Indonesia.")
15
16        case "venus", "sarracenia":
17            fmt.Println("Termasuk Tanaman Karnivora.")
18            fmt.Println("Tidak Asli Indonesia.")
19
20        default:
21            fmt.Println("Tidak termasuk Tanaman Karnivora.")
22        }
23    }
24 }
```

Terminal Output:

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 2\guided 2.go"
nepenthes
Termasuk Tanaman Karnivora.
```



This screenshot shows the same Go program as the first image, but the terminal output displays the results of multiple test cases. The program correctly identifies `nepenthes` and `venus` as carnivorous plants from Indonesia, while `drosera` is identified as a carnivorous plant but not from Indonesia, and `karedok` is not a carnivorous plant.

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 2\guided 2.go"
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 2\guided 2.go"
venus
Termasuk Tanaman Karnivora.
Tidak Asli Indonesia.
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 2\guided 2.go"
karedok
Tidak termasuk Tanaman Karnivora.
PS C:\Users\MSI>
```

Deskripsi program

Program Go ini berfungsi untuk **mengklasifikasikan nama-nama tanaman** yang dimasukkan oleh pengguna untuk mengetahui apakah tanaman tersebut termasuk **karnivora** dan apakah ia **berasal dari Indonesia**. Saat program dijalankan, ia akan menunggu input dari pengguna, yaitu nama tanaman. Setelah nama diterima, program akan membandingkannya dengan daftar yang sudah ditentukan menggunakan mekanisme `switch`. Ada dua kategori utama tanaman karnivora yang dikenali oleh program: pertama, jika nama yang dimasukkan adalah **"nepenthes"** atau **"drosera"**, program akan mengidentifikasi mereka sebagai **Tanaman Karnivora yang Asli Indonesia**. Kedua, jika nama yang dimasukkan adalah **"venus"** (seperti Venus flytrap) atau **"sarracenia"**, program akan tetap menggolongkannya sebagai **Tanaman**

Karnivora, tetapi dengan keterangan bahwa ia **Tidak Asli Indonesia**. Jika nama tanaman yang dimasukkan tidak ditemukan dalam daftar nama-nama karnivora tersebut, program akan secara otomatis menampilkan pesan bahwa tanaman tersebut **Tidak termasuk Tanaman Karnivora**.

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\n(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")

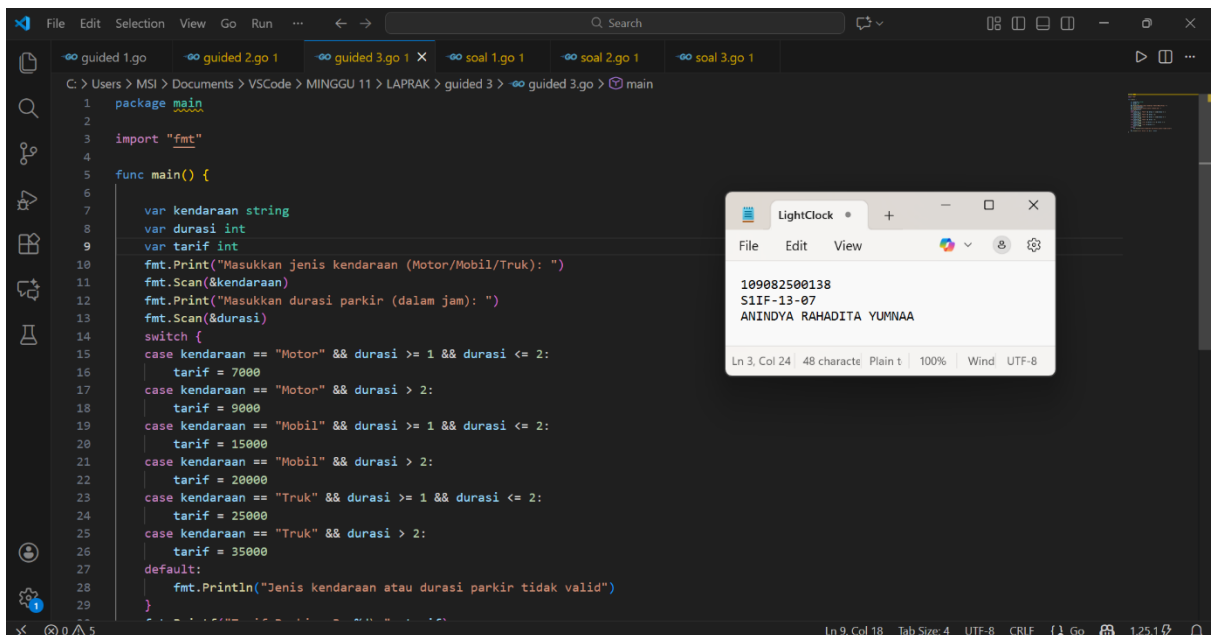
    }

    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)

}

```

Screenshoot program



The screenshot shows a Go program in VS Code and its execution output in a terminal window. The program is a parking fee calculator that takes vehicle type and duration as input and outputs the corresponding fee.

Go Program 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
10    fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
11    fmt.Scan(&kendaraan)
12    fmt.Print("Masukkan durasi parkir (dalam jam): ")
13    fmt.Scan(&durasi)
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 }

```

Terminal Output:

```

File Edit View
109082500138
SIIF-13-07
ANINDYA RAHADITA YUMNAA
Ln 3, Col 24 48 caracte Plain t 100% Wind UTF-8

```



```
func main() {  
    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:

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 3\guided 3.go"  
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor  
Masukkan durasi parkir (dalam jam): 2  
Tarif Parkir: Rp 7000  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 3\guided 3.go"  
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil  
Masukkan durasi parkir (dalam jam): 4  
Tarif Parkir: Rp 20000  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 3\guided 3.go"  
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk  
Masukkan durasi parkir (dalam jam): 1  
Tarif Parkir: Rp 25000  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 3\guided 3.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 C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 3\guided 3.go"  
Masukkan jenis kendaraan (Motor/Mobil/Truk):
```

Deskripsi program

Program Go ini bertugas menghitung berapa total biaya yang harus dibayar pengguna. Program akan bertanya dulu **jenis kendaraan** apa yang parkir (Motor, Mobil, atau Truk) dan **berapa lama** durasinya dalam jam.

Inti dari program ini ada pada bagaimana ia menetapkan tarif. Biaya parkir di sini tidak sama rata, program bergantung pada dua hal sekaligus: jenis kendaraan, dan apakah durasinya **pendek (1 sampai 2 jam)** atau **panjang (lebih dari 2 jam)**. Jadi, di balik layar, program membandingkan semua kemungkinan itu:

- Misalnya, jika parkir **Motor** selama **satu jam**, biayanya Rp 7.000. Tapi jika parkir **lebih dari 2 jam**, biayanya naik jadi Rp 9.000.
- Hal yang sama berlaku untuk **Mobil** dan **Truk**, di mana masing-masing memiliki dua tarif yang berbeda untuk durasi pendek dan durasi panjang.

Jika pengguna memasukkan jenis kendaraan yang tidak dikenal oleh sistem, program akan menampilkan pesan peringatan dan tidak melanjutkan perhitungan. Setelah semua kondisi dicocokkan, program akan menemukan tarif yang tepat dan langsung mencetaknya di layar.

TUGAS

1. Tugas 1

Source code

```
package main  
  
import "fmt"
```

```

func main() {

    var ph float64

    fmt.Print("Masukkan nilai pH: ")

    fmt.Scan(&ph)

    if ph < 0 || ph > 14 {

        fmt.Println("Nilai pH tidak valid. Nilai pH harus
antara 0 dan 14.")

    } else if ph >= 6.5 && ph <= 8.6 {

        fmt.Println("Air layak minum")

    } else {

        fmt.Println("Air tidak layak minum")

    }

}

```

Screenshoot program

The screenshot shows the VS Code interface with a Go file named `soal 1.go` open. The code defines a `main` function that prompts the user for a pH value and checks it against two conditions: validity (0 to 14) and suitability for drinking (6.5 to 8.6). The terminal output shows the program being run three times with different inputs: 8.6, 9, and 16. The first two runs show the program correctly identifying suitable and unsuitable drinking water. The third run shows the program correctly identifying an invalid pH value.

```

C: > Users > MSI > Documents > VSCode > MINGGU 11 > LAPRAK > soal 1 > soal 1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float64
7
8     fmt.Print("Masukkan nilai pH: ")
9     fmt.Scan(&ph)
10
11     if ph < 0 || ph > 14 {
12         fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
13     } else if ph >= 6.5 && ph <= 8.6 {
14         fmt.Println("Air layak minum")
15     } else {
16         fmt.Println("Air tidak layak minum")
17     }
18 }

```

```

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\soal 1\soal 1.go"
Masukkan nilai pH: 8.6
Air layak minum
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\soal 1\soal 1.go"
Masukkan nilai pH: 9
Air tidak layak minum
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\soal 1\soal 1.go"
Masukkan nilai pH: 16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.
PS C:\Users\MSI>

```

Deskripsi program

Program Go ini dirancang untuk **mengecek kelayakan air minum** berdasarkan nilai **pH** yang dimasukkan oleh pengguna, sebuah ukuran tingkat keasaman atau kebasaan air. Pertama, program akan meminta pengguna memasukkan nilai pH. Selanjutnya, ia akan memvalidasi input tersebut, yaitu memastikan bahwa nilai pH berada dalam rentang standar ilmiah dari **0 hingga 14**, jika di luar rentang ini, program akan menampilkan pesan kesalahan. Apabila nilai pH valid, program akan menentukan kelayakan air. Air dianggap **layak minum** jika nilai pH-nya berada dalam kisaran ideal **6.5 hingga 8.6**. Di luar rentang **6.5 sampai 8.6**, air akan diklasifikasikan sebagai tidak layak minum.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var jenis string

    var durasi int

    fmt.Print("Masukkan jenis kendaraan\n(motor/mobil/truk): ")

    fmt.Scan(&jenis)

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

    fmt.Scan(&durasi)

    if durasi < 1 {

        durasi = 1

    }
```

```
var tarif int

if jenis == "motor" {
    tarif = 2000
} else if jenis == "mobil" {
    tarif = 5000
} else if jenis == "truk" {
    tarif = 8000
} else {
    fmt.Println("Jenis kendaraan tidak valid.")
    return
}

total := tarif * durasi

fmt.Printf("Total biaya parkir: Rp %d\n", total)
}
```

Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var jenis string
7     var durasi int
8
9     fmt.Println("Masukkan jenis kendaraan (motor/mobil/truk): ")
10    fmt.Scan(&jenis)
11
12    fmt.Println("Masukkan durasi parkir (jam): ")
13    fmt.Scan(&durasi)
14
15    if durasi < 1 {
16        durasi = 1
17    }
18
19    var tarif int
20
21    if jenis == "motor" {
22        tarif = 2000
23    } else if jenis == "mobil" {
24        tarif = 5000
25    } else if jenis == "truk" {
26        tarif = 8000
27    } else {
28        fmt.Println("Jenis kendaraan tidak valid.")
29        return
30    }
```

```
23 } else if jenis == "mobil" {
24     tarif = 5000
25 } else if jenis == "truk" {
26     tarif = 8000
27 } else {
28     fmt.Println("Jenis kendaraan tidak valid.")
29     return
30 }
31
32 total := tarif * durasi
33 fmt.Printf("Total biaya parkir: Rp %d\n", total)
34 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\soal 2\soal 2.go"
Masukkan jenis kendaraan (motor/mobil/truk): motor
Masukkan durasi parkir (jam): 3
Total biaya parkir: Rp 6000
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\soal 2\soal 2.go"
Masukkan jenis kendaraan (motor/mobil/truk): mobil
Masukkan durasi parkir (jam): 1
Total biaya parkir: Rp 5000
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\soal 2\soal 2.go"
Masukkan jenis kendaraan (motor/mobil/truk): truk
Masukkan durasi parkir (jam): 5
Total biaya parkir: Rp 40000
PS C:\Users\MSI>
```

Deskripsi program

Program Go ini berfungsi sebagai **kalkulator sederhana untuk menghitung total biaya parkir** berdasarkan dua data input dari pengguna, **jenis kendaraan** (motor, mobil, atau truk) dan **lama durasi parkir** dalam satuan jam. Setelah menerima input, program akan segera memastikan bahwa durasi parkir minimum dihitung sebagai **satu jam**, bahkan jika pengguna memasukkan angka nol atau kurang, untuk menjamin adanya biaya parkir minimum. Selanjutnya, menggunakan struktur if-else if-else, program menetapkan tarif dasar per jam untuk kendaraan yaitu Rp 2.000 untuk motor, Rp 5.000 untuk mobil, dan Rp 8.000 untuk truk. Jika jenis kendaraan yang dimasukkan tidak sesuai, program akan menampilkan pesan kesalahan dan berhenti. Terakhir, program mengalikan tarif dasar yang sudah ditentukan dengan durasi parkir (yang

sudah dimodifikasi) untuk mendapatkan dan menampilkan **total biaya parkir** yang harus dibayar oleh pengguna.

3. Tugas 3

Source code

```
package main

import "fmt"

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

    if n%10 == 0 {
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        hasil := n / 10
        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, hasil)

    } else if n%5 == 0 {
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        hasil := n * n
        fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", n, hasil)

    } else if n%2 == 0 {
        fmt.Println("Kategori: Bilangan Genap")
        next := n + 1
        hasil := n * next
        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", n, next, hasil)

    } else {
        fmt.Println("Kategori: Bilangan Ganjil")
        next := n + 1
```

```

        hasil := n + next
        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n", n, next, hasil)
    }
}

```

Screenshoot program

```

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

```

```

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

```

```

Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\soal 3\soal 3.go"
Masukkan bilangan: 25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^2 = 625
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\soal 3\soal 3.go"
Masukkan bilangan: 20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS C:\Users\MSI>

```

Deskripsi program

Program Go ini berfungsi untuk **menganalisis sebuah angka** yang dimasukkan oleh pengguna, **mengelompokkannya** ke dalam salah satu dari empat kategori, dan kemudian **melakukan perhitungan** yang berbeda-beda untuk setiap kategori.

Cara Kerja dan Ketentuan Klasifikasi (Prioritas Berjenjang):

Program ini menggunakan sistem pengecekan berurutan, artinya jika satu kondisi sudah terpenuhi, kondisi di bawahnya tidak akan dicek lagi.

1. Kategori: Kelipatan 10

- **Ketentuan:** Angka yang dimasukkan **habis dibagi 10** (contoh: 20, 30, 100).
- **Tindakan:** Program akan mencetak bahwa angka tersebut adalah "Bilangan Kelipatan 10" dan kemudian **membagi angka tersebut dengan 10**.

2. Kategori: Kelipatan 5

- **Ketentuan:** Angka **tidak habis dibagi 10**, tetapi **habis dibagi 5** (contoh: 5, 15, 25).
- **Tindakan:** Program akan mencetak "Bilangan Kelipatan 5" dan kemudian **mengkuadratkan** angka tersebut.

3. Kategori: Bilangan Genap

- **Ketentuan:** Angka **tidak termasuk** kelipatan 10 atau kelipatan 5, tetapi **habis dibagi 2** (contoh: 2, 4, 6, 8).
- **Tindakan:** Program akan mencetak "Bilangan Genap" dan kemudian **mengalikannya dengan bilangan bulat berikutnya** (misalnya, jika input 8, dihitung $8+9$).

4. Kategori: Bilangan Ganjil

- **Ketentuan:** Angka **tidak memenuhi** semua kondisi di atas.
- **Tindakan:** Program akan mencetak "Bilangan Ganjil" dan kemudian **menjumlahkan** angka tersebut dengan bilangan bulat berikutnya (misalnya, jika input 7, dihitung $7 + 8$).