

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 11
SWITCH-CASE**



Disusun oleh:

RIZKY TABRIZ DEANOVA

109082500177

S1IF-13-07

Asisten Praktikum

Adithana Dharma Putra

Apri Pandu Wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

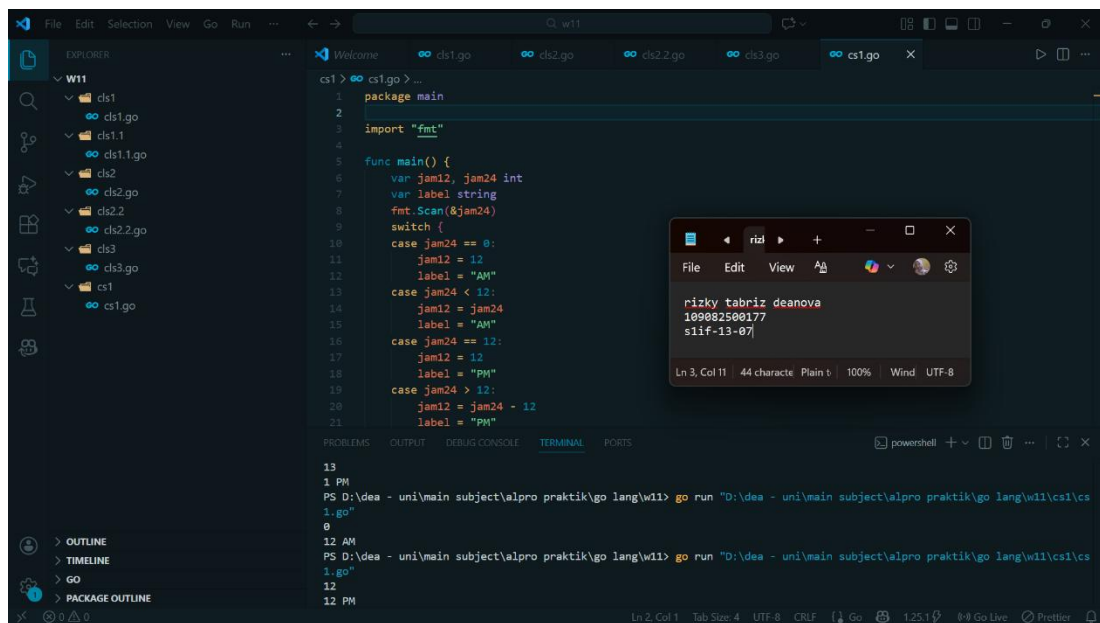
1. Guided 1 Source Code

```
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

package main sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

import "fmt" yang digunakan untuk memasukkan library "fmt" untuk dipakai menjalankan program nantinya.

func main () {} yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah func main () nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

var dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

fmt.Println yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di garis baru.

fmt.Scan yang dalam bahasa pemrograman Go digunakan untuk menunda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

switch berfungsi untuk membandingkan input ke dalam banyaknya opsi kondisi

case berfungsi sebagai tempat pemeriksaan apakah input sesuai atau sama dengan kondisi di baris case tersebut, jika tidak maka akan dilempar ke case lain atau ke default atau dapat disebut sebagai label opsi

default adalah kondisi alternatif yang mana jika input tidak sesuai dengan seluruh case, maka akan masuk ke dalam kondisi default

Yang dilakukan program di dalam gambar di atas adalah **Konversi Waktu Dari Bentuk 24 Jam Ke Dalam Bentuk 12 Jam** yang mana terlihat melalui kode dari baris 9 hingga baris 21.

Pemeriksaan Switch-Case:

1. Input dari user akan diperiksa pada baris case pertama (`case jam24 == 0:`) yang akan dilanjutkan ke dalam tahapan eksekusi lanjutan jika memenuhi syarat case pertama. Permisalan input: 0, karena sesuai dengan kondisi case, maka akan dikonversi ke bentuk 12 jam, yaitu 0 dengan label AM (12 AM)
2. Jika tidak sesuai dengan case pertama, akan diturunkan ke case kedua (`case jam24 < 12:`), jika input kurang dari angka 12, maka akan dilanjutkan ke eksekusi di dalam case kedua. Permisalan input: 11, karena sesuai dengan kondisi case kedua (input kurang dari 12), maka nilai konversi tetap sama, yaitu 11 dengan label AM (11 AM)
3. Jika tidak sesuai dengan case kedua, akan diturunkan ke case ketiga (`case jam24 == 12:`), jika input sama dengan angka 12, maka akan dilanjutkan ke eksekusi dalam kondisi case ketiga tersebut. Permisalan input: 12, maka nilai konversi akan tetap sama, yaitu 12 dengan label PM (12 PM)
4. Jika tidak sesuai dengan case ketiga, akan diturunkan ke case keempat (`case jam24 > 12:`), jika input lebih dari angka 12, maka akan dilanjutkan ke eksekusi dalam kondisi case keempat. Permisalan input: 13, karena lebih dari 12, maka akan dikonversi melalui pengurangan ($13 - 12$), yang menghasilkan 1, lalu untuk labelnya adalah PM (1 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.")

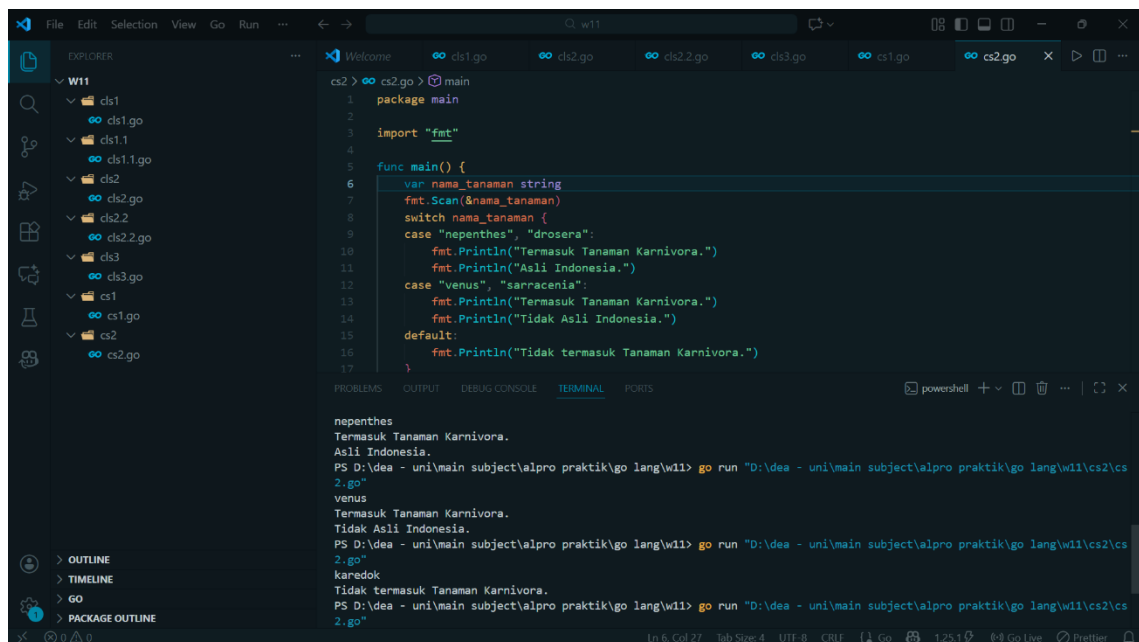
    default:

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

    }

}
```

Screenshoot program



Deskripsi program

`package main` sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

`import "fmt"` yang digunakan untuk memasukkan library `"fmt"` untuk dipakai menjalankan program nantinya.

`func main () {}` yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung `{}` setelah `func main ()` nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

`var` dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

`fmt.Println` yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di garis baru.

`fmt.Scan` yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

`switch` berfungsi untuk membandingkan input ke dalam banyaknya opsi kondisi

`case` berfungsi sebagai tempat pemeriksaan apakah input sesuai atau sama dengan kondisi di baris `case` tersebut, jika tidak maka akan dilempar ke `case` lain atau ke default atau dapat disebut sebagai label opsi

`default` adalah kondisi alternatif yang mana jika input tidak sesuai dengan seluruh `case`, maka akan masuk ke dalam kondisi default

Yang dilakukan program di dalam gambar di atas adalah **Pengecekan Jenis dan Asal Tanaman** yang mana terlihat melalui kode dari **baris 8 hingga baris 16**.

Pemeriksaan Switch-Case:

1. Memeriksa apakah input sesuai dengan `case` pertama (`case "nepenthes", "drosera":`), jika sesuai, maka akan dilanjutkan dengan eksekusi kode dalam `case` pertama.
2. Jika input tidak sesuai dengan kondisi `case` pertama, maka akan diturunkan ke `case` kedua (`case "venus", "sarracenia":`), jika sesuai, maka akan dilanjutkan dengan eksekusi kode dalam `case` kedua
3. Jika input tidak sesuai dengan kondisi seluruh `case`, maka akan masuk ke dalam eksekusi default (`fmt.Println("Tidak termasuk Tanaman Karnivora.")`)

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

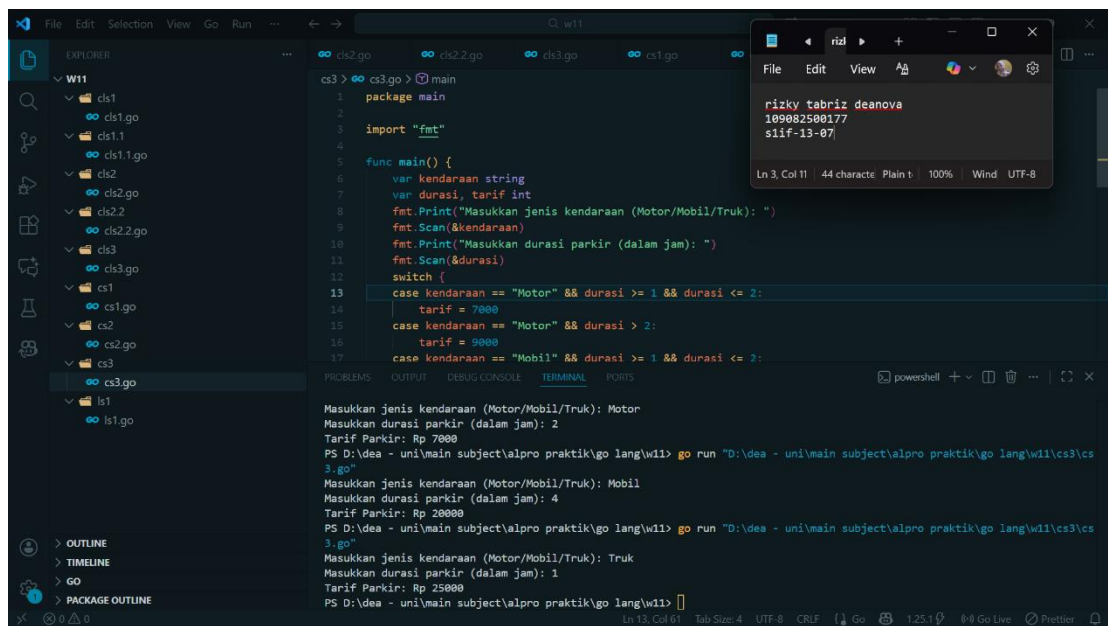
    var kendaraan string

    var durasi, 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. The Explorer pane on the left shows a project structure with folders 'W11' and 'cs3'. The main editor displays the source code for 'cs3.go'. The code defines a package 'main', imports 'fmt', and a 'main' function. The function declares variables for vehicle type, duration, and tariff. It uses 'fmt.Print' for prompts, 'fmt.Scan' for input, and a 'switch' statement to calculate the parking tariff based on vehicle type and duration. The terminal at the bottom shows the execution of the program with three test cases: Motor (2 hours, Rp 7000), Mobil (4 hours, Rp 20000), and Truk (1 hour, Rp 25000). A small window in the top right corner shows the user's name 'rizky tabriz deanova' and a timestamp '109082500177 s11f-13-07'.

```
cs3 > cd cs3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var durasi, tarif int
8     fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
9     fmt.Scan(&kendaraan)
10    fmt.Print("Masukkan durasi parkir (dalam jam): ")
11    fmt.Scan(&durasi)
12    switch {
13    case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
14        tarif = 7000
15    case kendaraan == "Motor" && durasi > 2:
16        tarif = 9000
17    case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
18        tarif = 20000
19    case kendaraan == "Mobil" && durasi > 2:
20        tarif = 25000
21    case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
22        tarif = 25000
23    case kendaraan == "Truk" && durasi > 2:
24        tarif = 30000
25    default:
26        tarif = 0
27    }
28    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
29 }
```

Terminal Output:

```
PS D:\dea - uni\main subject\alpro praktik\go lang\w11> go run "D:\dea - uni\main subject\alpro praktik\go lang\w11\cs3\cs3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS D:\dea - uni\main subject\alpro praktik\go lang\w11> go run "D:\dea - uni\main subject\alpro praktik\go lang\w11\cs3\cs3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS D:\dea - uni\main subject\alpro praktik\go lang\w11> go run "D:\dea - uni\main subject\alpro praktik\go lang\w11\cs3\cs3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS D:\dea - uni\main subject\alpro praktik\go lang\w11>
```

Deskripsi program

`package main` sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

`import "fmt"` yang digunakan untuk memasukkan library `"fmt"` untuk dipakai menjalankan program nantinya.

`func main () {}` yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung `{}` setelah `func main ()` nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

`var` dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

`fmt.Print` yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output.

`fmt.Println` yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di baris baru.

`fmt.Printf` berfungsi sebagai pencetakan output terformat.

`fmt.Scan` yang dalam bahasa pemrograman Go digunakan untuk menunda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

`switch` berfungsi untuk membandingkan input ke dalam banyaknya opsi kondisi

`case` berfungsi sebagai tempat pemeriksaan apakah input sesuai atau sama dengan kondisi di baris `case` tersebut, jika tidak maka akan dilempar ke `case` lain atau ke default atau dapat disebut sebagai label opsi

`default` adalah kondisi alternatif yang mana jika input tidak sesuai dengan seluruh `case`, maka akan masuk ke dalam kondisi default

`%d` adalah tempat penyimpanan (placeholder) untuk bilangan bulat (integer)

`\n` berfungsi untuk membuat baris baru

Yang dilakukan program di dalam gambar di atas adalah **Menentukan Tarif Parkir Sesuai Jenis Kendaraan & Durasi Parkir** yang mana terlihat melalui kode dari **baris 12 hingga baris 28**.

Pemeriksaan Switch-Case:

1. Memeriksa input ke dalam case pertama (`case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:`), jika sesuai dengan kondisi case, maka akan dilanjutkan dengan eksekusi lanjutan dari case pertama.
2. Jika tidak sesuai, akan diturunkan atau lanjut memeriksa kesesuaian input ke dalam kondisi case yang tercantum di dalam program satu-persatu.
3. Jika tidak terdapat kondisi yang sesuai dengan input, maka akan diturunkan ke dalam kondisi default (`fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")`).
4. Terakhir, akan dilaksanakan cetak output (`fmt.Printf("Tarif Parkir: Rp %d\n", tarif)`) dengan cara melakukan pencetakan format integer dari variabel `tarif` yang dieksekusi.

TUGAS

1. Tugas 1

Source code

```
package main

import (
    "fmt"
)

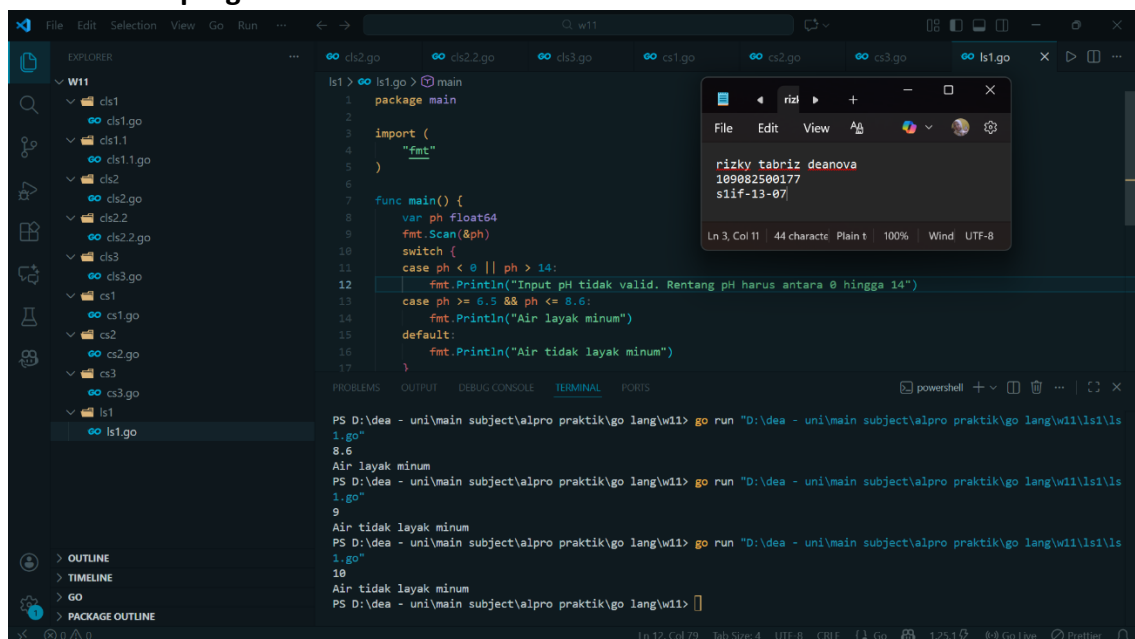
func main() {
    var ph float64
    fmt.Scan(&ph)

    switch {
    case ph < 0 || ph > 14:
        fmt.Println("Input pH tidak valid. Rentang pH harus antara 0 hingga 14")

    case ph >= 6.5 && ph <= 8.6:
        fmt.Println("Air layak minum")

    default:
        fmt.Println("Air tidak layak minum")
    }
}
```

Screenshoot program



Deskripsi program

`package main` sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

`import "fmt"` yang digunakan untuk memasukkan library `"fmt"` untuk dipakai menjalankan program nantinya.

`func main () {}` yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung `{}` setelah `func main ()` nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

`var` dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

`fmt.Println` yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output di baris baru.

`fmt.Scan` yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

`switch` berfungsi untuk membandingkan input ke dalam banyaknya opsi kondisi

`case` berfungsi sebagai tempat pemeriksaan apakah input sesuai atau sama dengan kondisi di baris `case` tersebut, jika tidak maka akan dilempar ke `case` lain atau ke default atau dapat disebut sebagai label opsi

`default` adalah kondisi alternatif yang mana jika input tidak sesuai dengan seluruh `case`, maka akan masuk ke dalam kondisi default

Yang dilakukan program di dalam gambar di atas adalah **Kelayakan Air Minum Dari pH Yang Diberikan** yang mana terlihat melalui kode dari **baris 10 hingga baris 16**.

Pemeriksaan Switch-Case:

1. Memeriksa kesesuaian input ke dalam kondisi `case` pertama (`case ph < 0 || ph > 14:`) terlebih dahulu, jika input sesuai, maka akan dilanjutkan dengan eksekusi kode dalam `case` pertama.
2. Jika input tidak sesuai dengan `case` pertama, maka akan diturunkan ke dalam `case` kedua (`case ph >= 6.5 && ph <= 8.6:`) untuk diperiksa kesesuaiannya, jika sesuai, maka akan dilanjutkan ke dalam eksekusi kode dalam `case` kedua.
3. Jika input tidak sesuai dengan seluruh `case`, maka akan masuk ke dalam kondisi default (`fmt.Println("Air tidak layak minum")`)

2. Tugas 2

Source code

```
package main

import (
    "fmt"
)
```

```

func main() {

    var kendaraan string

    var durasi, tarif int

    fmt.Scan(&kendaraan, &durasi)

    switch kendaraan {

    case "motor":

        tarif = 2000

    case "mobil":

        tarif = 5000

    case "truk":

        tarif = 8000

    default:

        fmt.Println("invalid kendaraan")

        return

    }

    if durasi < 1 {

        durasi = 1

    }

    biaya := tarif * durasi

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

}

```

Screenshoot program

The screenshot shows an IDE with a Go program and its execution results. The code defines a function `main` that takes a vehicle type and duration as input, calculates the cost based on a switch statement, and prints the result. The execution output shows the program running successfully for three different inputs: motor, mobil, and truk.

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var kendaraan string
9     var durasi, tarif int
10    fmt.Scan(&kendaraan, &durasi)
11    switch kendaraan {
12    case "motor":
13        tarif = 2000
14    case "mobil":
15        tarif = 5000
16    case "truk":
17        tarif = 8000
18    }
19    if durasi < 1 {
20        durasi = 1
21    }
22    biaya := tarif * durasi
23    fmt.Printf("Rp %d\n", biaya)
24 }

```

Terminal Output:

```

PS D:\dea - uni\main subject\alpro praktik\go lang\w11> go run "D:\dea - uni\main subject\alpro praktik\go lang\w11\ls2
.go"
motor 3 jam
Rp 6000
PS D:\dea - uni\main subject\alpro praktik\go lang\w11> go run "D:\dea - uni\main subject\alpro praktik\go lang\w11\ls2
.go"
mobil 1 jam
Rp 5000
PS D:\dea - uni\main subject\alpro praktik\go lang\w11> go run "D:\dea - uni\main subject\alpro praktik\go lang\w11\ls2
.go"
truk 5 jam
Rp 40000
PS D:\dea - uni\main subject\alpro praktik\go lang\w11>

```

Deskripsi program

`package main` sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

`import "fmt"` yang digunakan untuk memasukkan library `"fmt"` untuk dipakai menjalankan program nantinya.

`func main () {}` yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung `{}` setelah `func main ()` nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

`var` dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

`fmt.Print` yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output.

`fmt.Scan` yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

`switch` berfungsi untuk membandingkan input ke dalam banyaknya opsi kondisi

`case` berfungsi sebagai tempat pemeriksaan apakah input sesuai atau sama dengan kondisi di baris `case` tersebut, jika tidak maka akan dilempar ke `case` lain atau ke default atau dapat disebut sebagai label opsi

`fmt.Printf` adalah pencetakan output yang terformat

`default` adalah kondisi alternatif yang mana jika input tidak sesuai dengan seluruh `case`, maka akan masuk ke dalam kondisi default

`%d` adalah tempat penyimpanan (placeholder) untuk bilangan bulat (integer)

`\n` berfungsi untuk membuat baris baru

Yang dilakukan program di dalam gambar di atas adalah **Menentukan Biaya Parkir Sesuai Jenis Kendaraan & Durasi Parkir** yang mana terlihat melalui kode dari **baris 11 dan 27**.

Pemeriksaan Switch-Case:

1. Memeriksa kesesuaian input ke dalam kondisi `case` pertama (`case "motor":`) terlebih dahulu, jika input sesuai, maka akan dilanjutkan dengan eksekusi kode dalam `case` pertama.
2. Jika input tidak sesuai dengan `case` pertama, maka akan diturunkan ke dalam `case` kedua (`case "mobil":`) untuk diperiksa kesesuaiannya, jika sesuai, maka akan dilanjutkan ke dalam eksekusi kode dalam `case` kedua.
3. Jika input tidak sesuai dengan seluruh `case`, maka akan masuk ke dalam kondisi default (`fmt.Println("invalid kendaraan ")`)
4. Setelahnya, dilakukan perubahan nilai input durasi menjadi 1, jika nilainya kurang dari 1. Bila nilai durasi lebih dari 1 atau tidak sesuai dengan kondisi yang tercantum dalam `if`, maka akan dihiraukan dan lanjut ke dalam operasi biaya.
5. Dalam variabel biaya, dilakukan perkalian terhadap nilai variabel tarif dengan durasi (`biaya := tarif * durasi`). Nilai dari operasi ini akan masuk ke dalam variabel biaya.
6. Melakukan pencetakan terformat (`fmt.Printf("Rp %d\n", biaya)`) untuk menampilkan total biaya yang perlu dibayarkan dalam integer.

3. Tugas 3

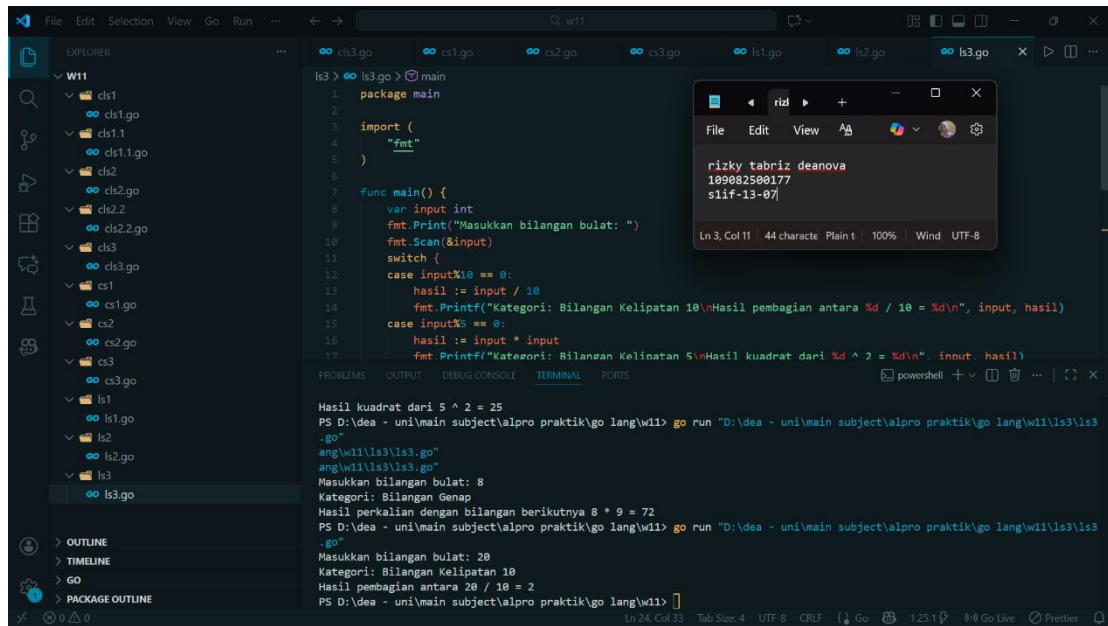
Source code

```
package main

import (
    "fmt"
)

func main() {
    var input int
    fmt.Print("Masukkan bilangan bulat: ")
    fmt.Scan(&input)
    switch {
    case input%10 == 0:
        hasil := input / 10
        fmt.Printf("Kategori: Bilangan Kelipatan 10\nHasil pembagian antara %d / 10 = %d\n", input, hasil)
    case input%5 == 0:
        hasil := input * input
        fmt.Printf("Kategori: Bilangan Kelipatan 5\nHasil kuadrat dari %d ^ 2 = %d\n", input, hasil)
    case input%2 == 0:
        tambah1 := input + 1
        hasil := input * tambah1
        fmt.Printf("Kategori: Bilangan Genap\nHasil perkalian dengan bilangan berikutnya %d * %d = %d\n", input, tambah1, hasil)
    default:
        tambah1 := input + 1
        hasil := input + tambah1
        fmt.Printf("Kategori: Bilangan Ganjil\nHasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", input, tambah1, hasil)
    }
}
```

Screenshoot program



```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var input int
9     fmt.Print("Masukkan bilangan bulat: ")
10    fmt.Scan(&input)
11    switch {
12    case input%10 == 0:
13        hasil := input / 10
14        fmt.Printf("Kategori: Bilangan Kelipatan 10\nHasil pembagian antara %d / 10 = %d\n", input, hasil)
15    case input%5 == 0:
16        hasil := input * input
17        fmt.Printf("Kategori: Bilangan Kelipatan 5\nHasil kuadrat dari %d ^ 2 = %d\n", input, hasil)
18    default:
19        hasil := input * 2
20        fmt.Printf("Kategori: Bilangan Genap\nHasil perkalian dengan 2 = %d\n", input, hasil)
21    }
22}
```

Hasil kuadrat dari 5 ^ 2 = 25
PS D:\dea - uni\main subject\alpro praktik\go lang\w11> go run "D:\dea - uni\main subject\alpro praktik\go lang\w11\ls3\ls3.go"
ang\w11\ls3\ls3.go
ang\w11\ls3\ls3.go
Masukkan bilangan bulat: 8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 2 = 16
PS D:\dea - uni\main subject\alpro praktik\go lang\w11> go run "D:\dea - uni\main subject\alpro praktik\go lang\w11\ls3\ls3.go"
Masukkan bilangan bulat: 20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS D:\dea - uni\main subject\alpro praktik\go lang\w11>

Deskripsi program

`package main` sebagai penanda bahwa program ini merupakan program *executable* atau program yang dapat dieksekusi (dijalankan).

`import "fmt"` yang digunakan untuk memasukkan library "fmt" untuk dipakai menjalankan program nantinya.

`func main () {}` yang berperan sebagai penanda tempat program mulai berjalan, seluruh kode ataupun instruksi yang terdapat di dalam kurung {} setelah `func main ()` nantinya akan dieksekusi atau dijalankan secara sistematis (berurutan).

`var` dalam program adalah sebagai wadah utama untuk penyimpanan data sebelum nanti dibagi lagi sesuai tipe data.

`fmt.Print` yang dalam bahasa pemrograman Go digunakan sebagai instruksi untuk mencetak atau menampilkan output.

`fmt.Scan` yang dalam bahasa pemrograman Go digunakan untuk menjeda hasil output sejenak guna membaca input user, yang kemudian disimpan di dalam variabel.

`switch` berfungsi untuk membandingkan input ke dalam banyaknya opsi kondisi

`case` berfungsi sebagai tempat pemeriksaan apakah input sesuai atau sama dengan kondisi di baris `case` tersebut, jika tidak maka akan dilempar ke `case` lain atau ke default atau dapat disebut sebagai label opsi

`fmt.Printf` adalah pencetakan output yang terformat

`default` adalah kondisi alternatif yang mana jika input tidak sesuai dengan seluruh `case`, maka akan masuk ke dalam kondisi default

`%d` adalah tempat penyimpanan (placeholder) untuk bilangan bulat (integer)

`\n` berfungsi untuk membuat baris baru

Yang dilakukan program di dalam gambar di atas adalah **Operasi Matematika Dasar (Bagi, Kuadrat, Kali, Jumlah) Dengan Persyaratan Tercantum** yang mana terlihat melalui kode dari **baris 11 hingga baris 25**.

Pemeriksaan Switch-Case:

1. Pemeriksaan kesesuaian input berdasarkan kondisi dari setiap case, jika terdapat kondisi case yang sesuai, maka akan dilanjutkan dengan eksekusi operasi dalam kondisi case tersebut.
 2. Jika tidak terdapat kondisi yang sesuai dengan case, maka akan masuk ke dalam kondisi default.
- Kelipatan 10: Hasil modulo input dan angka 10 sama dengan 0, maka akan dilanjutkan dengan pencetakan kategori bilangan input dengan pembagian antara nilai input dan angka 10 beserta hasilnya
 - Kelipatan 5: Hasil modulo input dan angka 5 sama dengan 0, maka akan dilanjutkan dengan pencetakan kategori bilangan input dengan kuadrat dari nilai input beserta hasilnya
 - Bilangan Genap: Jika nilai input tidak habis dimodulo dengan 10, akan masuk ke dalam operasi ini. Dalam operasi ini, jika hasil modulo nilai input dan angka 2 sama dengan 0, maka akan dilanjutkan dengan pencetakan kategori dan perkalian dengan bilangan berikutnya (jika nilai input adalah 8, maka bilangan berikutnya adalah 9) beserta hasilnya
 - Bilangan Ganjil: Jika nilai input tidak habis dimodulo dengan 5, akan masuk ke dalam operasi default. Dalam operasi ini, akan dicetak kategori bilangan dari input dan penjumlahan dengan bilangan berikutnya (jika nilai input adalah 9, maka bilangan berikutnya adalah 10) beserta hasilnya