

LAPORAN PRAKTIKUM

Algoritma Pemrograman

MODUL 11

SWITCH-CASE



Disusun oleh:

RAYSA RAHMA IRAHIM

109082500167

S1IF-13-04

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 bil int

    fmt.Scan(&bil)


    switch bil {

    case 0:

        fmt.Print("12 AM")

    case 12:

        fmt.Print("12 PM")

    case 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11:

        fmt.Print(bil, "AM")

    case 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23:

        bil = bil - 12

        fmt.Print(bil, "PM")

    default:

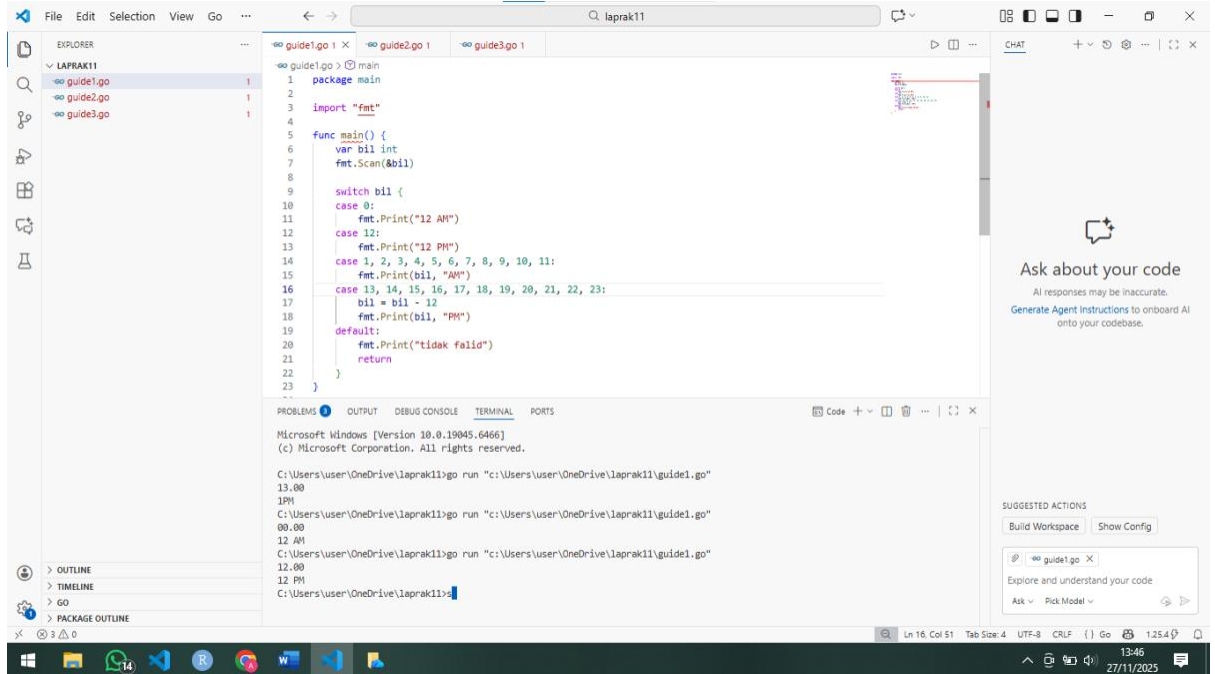
        fmt.Print("tidak valid")

        return

    }

}
```

Sreenshoot Program



Deskripsi program

1. package main

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

2. import "fmt"

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

3. func main()

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

4. Tanda buka dan tutup kurung kurawal { }

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

5. var bil int

Baris var bil int digunakan untuk membuat sebuah variabel bernama bil dengan tipe data integer. Variabel ini nantinya akan digunakan untuk menyimpan angka jam yang dimasukkan. Tipe integer dipilih karena nilai jam merupakan bilangan bulat, bukan teks atau bilangan desimal. Dengan mendeklarasikan variabel ini, kita menyediakan tempat penyimpanan yang akan diisi dengan input sebelum dilakukan proses pengecekan dan konversi waktu.

6. fmt.Scan(&bil)

Bagian fmt.Scan(&bil) berfungsi membaca masukan dari pengguna dan menyimpannya ke dalam variabel bil. Tanda & digunakan untuk memberikan alamat memori dari variabel tersebut kepada fungsi Scan, sehingga nilai yang dimasukkan bisa langsung disimpan di dalam variabel bil.

7. switch bil { ... }

Kode switch bil digunakan untuk memeriksa nilai dari variabel bil dan kemudian menentukan tindakan yang sesuai berdasarkan nilai tersebut. Switch bekerja seperti percabangan yang lebih rapi dibandingkan if-else, karena kita dapat membuat banyak kondisi berbeda dalam struktur yang lebih terorganisir. Pada program konversi waktu ini, switch digunakan untuk menentukan apakah nilai bil menunjukkan jam AM, PM, atau tidak valid. Setiap case mewakili situasi jam tertentu yang harus ditangani.

8. case 0: fmt.Print("12 AM")

Pada bagian case 0, program menangani kondisi ketika pengguna memasukkan angka 0. Dalam format 24 jam, angka 0 berarti pukul 12 malam, yang dalam format 12 jam ditulis sebagai 12 AM. Karena itu, ketika nilai bil adalah 0, program langsung menampilkan tulisan "12 AM" tanpa melakukan perhitungan tambahan.

9. case 12: fmt.Print("12 PM")

Case ini menangani kondisi ketika nilai bil adalah 12. Dalam format 24 jam, 12 berarti jam 12 siang, dan dalam format 12 jam tetap dituliskan sebagai 12 PM. Berbeda dengan jam lain yang perlu dikonversi, angka 12 tidak diubah menjadi angka lain karena sudah memiliki bentuk yang sama dalam format 12 jam.

10. case 1–11

Untuk nilai bil yang berada dalam rentang 1 sampai 11, program menganggap jam tersebut sebagai waktu AM. Nilai jam tidak perlu dikonversi, karena jam 1 pagi hingga jam 11 pagi memang langsung sama bentuknya dalam format 12 jam.

11. case 13–23

Ketika nilai bil berada pada rentang 13 hingga 23, itu berarti jam tersebut dalam format 24 jam yang harus dikonversi menjadi format PM. Proses konversinya adalah dengan mengurangi angka tersebut sebesar 12. Misalnya, jam 13 dalam format 24 jam setara dengan jam 1 PM, jam 18 menjadi 6 PM, dan jam 23 menjadi 11 PM. Setelah pengurangan dilakukan, program menampilkan hasilnya lalu menambahkan label "PM".

12. default: fmt.Print("tidak valid")

Bagian default akan dijalankan ketika nilai jam yang dimasukkan pengguna tidak berada dalam rentang 0 hingga 23. Nilai seperti -2, 50, atau 100 tidak mungkin mewakili jam yang valid dalam satu hari. Karena itu, jika input berada di luar rentang tersebut, program menampilkan pesan "tidak valid" sebagai tanda bahwa input tidak

dapat diproses. Bagian ini berfungsi untuk semua masukan yang tidak termasuk ke dalam daftar tanaman yang dikenali program, sehingga program tetap dapat memberikan output yang jelas meskipun inputnya di luar daftar.

2. Guided 2

Source Code

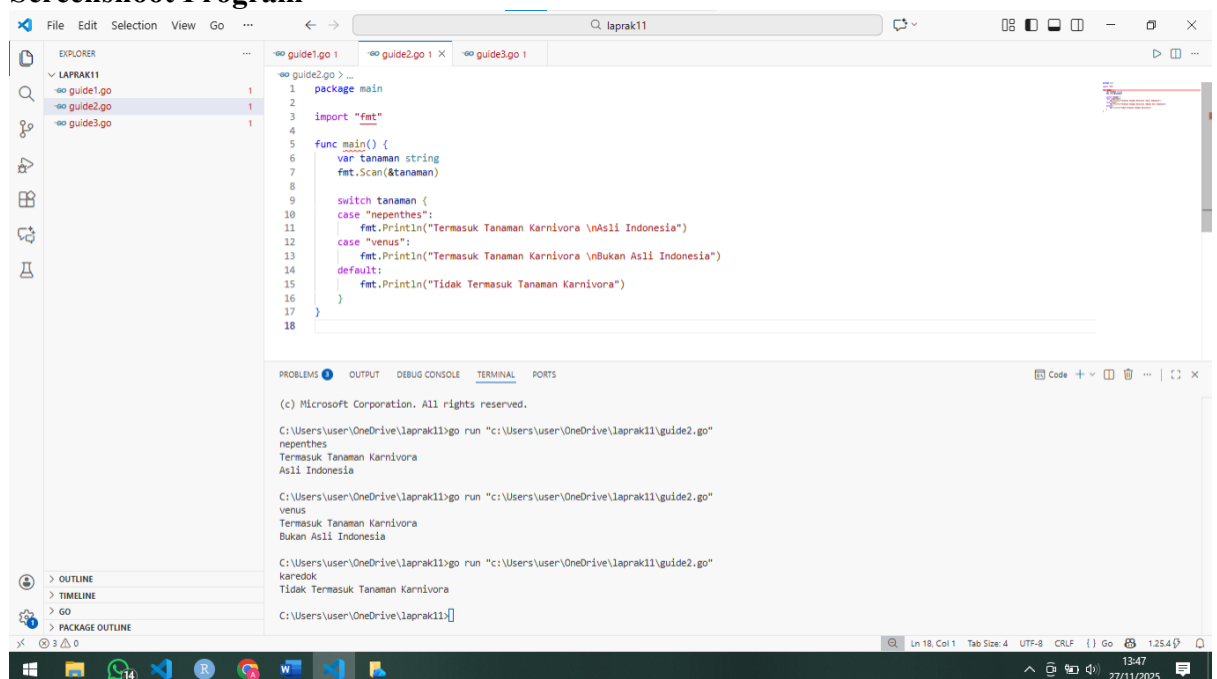
```
package main

import "fmt"

func main() {
    var tanaman string
    fmt.Scan(&tanaman)

    switch tanaman {
    case "nepenthes":
        fmt.Println("Termasuk Tanaman Karnivora \nAsli Indonesia")
    case "venus":
        fmt.Println("Termasuk Tanaman Karnivora \nBukan Asli Indonesia")
    default:
        fmt.Println("Tidak Termasuk Tanaman Karnivora")
    }
}
```

Screenshoot Program



Deskripsi program

1. package main

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

2. import "fmt"

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

3. func main()

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layar. Tanpa main(), program tidak akan bisa dijalankan.

4. Tanda buka dan tutup kurung kurawal { }

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

5. var tanaman string

Baris var tanaman string digunakan untuk membuat variabel baru bernama tanaman dengan tipe data string. Variabel ini disiapkan sebagai penampung input yang diberikan oleh pengguna, yaitu nama tanaman yang ingin dicek kategorinya. Karena nama tanaman berupa teks, tipe data string menjadi pilihan yang tepat.

6. fmt.Scan(&tanaman)

Baris fmt.Scan(&tanaman) berfungsi untuk membaca masukan (input) dari pengguna dan menyimpannya ke dalam variabel tanaman. Tanda & sebelum nama variabel berarti memberikan alamat memori dari variabel tersebut kepada fungsi Scan, agar nilai input dapat langsung disimpan ke dalam variabel tersebut.

7. switch tanaman { ... }

Struktur switch tanaman digunakan untuk memeriksa nilai yang tersimpan di dalam variabel tanaman dan menentukan perintah mana yang harus dijalankan berdasarkan nilai tersebut. Pada program ini, switch digunakan untuk membandingkan nama tanaman yang dimasukkan pengguna dengan kategori tanaman tertentu, yaitu “nepenthes” dan “venus”. Setiap case akan memberikan output sesuai kategori tanaman tersebut.

8. case "nepenthes": fmt.Println("Termasuk Tanaman Karnivora \nAsli Indonesia")

Case pertama menangani situasi ketika pengguna memasukkan kata “nepenthes”. Jika nilai variabel tanaman sama dengan “nepenthes”, maka program menampilkan informasi bahwa tanaman tersebut adalah tanaman karnivora dan merupakan spesies asli Indonesia.

9. case "venus": fmt.Println("Termasuk Tanaman Karnivora \nBukan Asli Indonesia")

Case berikutnya menangani kondisi ketika pengguna mengetikkan “venus”. Jika nilai variabel tanaman sesuai dengan kata tersebut, maka program akan menampilkan teks yang menyatakan bahwa tanaman venus juga merupakan tanaman karnivora, tetapi bukan berasal dari Indonesia.

10. default: fmt.Println("Tidak Termasuk Tanaman Karnivora")

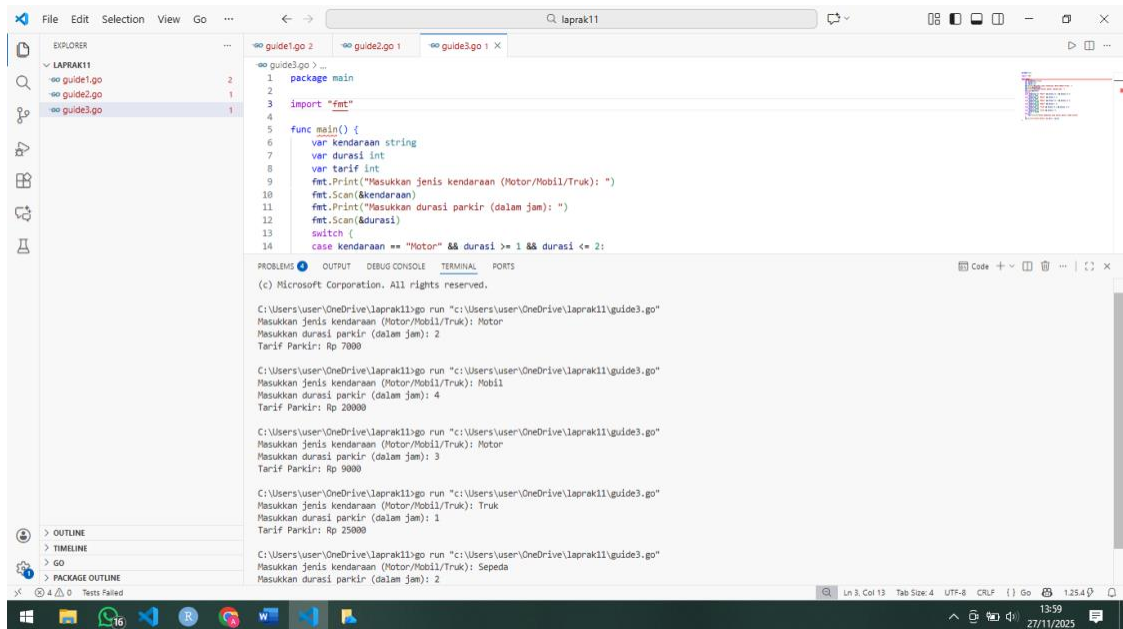
Bagian default akan dijalankan ketika input pengguna tidak cocok dengan case manapun, yaitu bukan “nepenthes” maupun “venus”. Jika terjadi hal tersebut, program memberikan respons berupa pesan “Tidak Termasuk Tanaman Karnivora”.

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 (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 displays the Visual Studio Code interface with a Go project named 'laprak11'. The Explorer sidebar on the left shows the project structure with files 'guide1.go', 'guide2.go', and 'guide3.go'. The main editor window shows the code for 'guide3.go', which includes a package declaration, an import for 'fmt', and a main function. The main function uses 'fmt.Scan' to read user input for vehicle type and parking duration, and a switch statement to calculate the parking fee based on the vehicle type and duration.

```
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    switch {
14    case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
```

The Output window at the bottom shows the execution results of the program. It displays the command used to run the program and the user input for each test case, along with the calculated parking fee.

```
C:\Users\User\OneDrive\laprak11>go run "c:\Users\User\OneDrive\laprak11\guide3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000

C:\Users\User\OneDrive\laprak11>go run "c:\Users\User\OneDrive\laprak11\guide3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000

C:\Users\User\OneDrive\laprak11>go run "c:\Users\User\OneDrive\laprak11\guide3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir: Rp 9000

C:\Users\User\OneDrive\laprak11>go run "c:\Users\User\OneDrive\laprak11\guide3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000

C:\Users\User\OneDrive\laprak11>go run "c:\Users\User\OneDrive\laprak11\guide3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Sepeda
Masukkan durasi parkir (dalam jam): 2
```

The status bar at the bottom indicates the current file is 'Ln 3, Col 13', the tab size is 4, the encoding is UTF-8, and the line ending is CRLF. The system clock shows 13:59 on 27/11/2025.

Deskripsi program

1. package main

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

2. import "fmt"

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

3. func main()

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layar. Tanpa main(), program tidak akan bisa dijalankan.

4. Tanda buka dan tutup kurung kurawal { }

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

5. var kendaraan string, var durasi int, var tarif int

Baris ini dideklarasikan variabel untuk membuat tiga variabel yang akan digunakan sepanjang program. Variabel kendaraan bertipe string dan digunakan untuk menampung jenis kendaraan yang dimasukkan pengguna, seperti Motor, Mobil, atau Truk. Variabel durasi bertipe integer dan digunakan untuk menyimpan lama parkir dalam satuan jam. Sedangkan variabel tarif juga bertipe integer dan akan menjadi tempat penyimpanan hasil perhitungan biaya parkir berdasarkan jenis kendaraan dan lama durasi yang diinputkan

6. `fmt.Print(...)` dan `fmt.Scan(&kendaraan / &durasi)`

Baris `fmt.Print` digunakan untuk menampilkan pesan ke layar sebagai instruksi agar pengguna mengetahui data apa yang harus dimasukkan. Setelah menampilkan teks tersebut, program menggunakan `fmt.Scan(&kendaraan)` untuk membaca input jenis kendaraan dari pengguna dan menyimpannya ke variabel `kendaraan`. Proses yang sama untuk membaca durasi parkir melalui `fmt.Scan(&durasi)`. Tanda `&` menunjukkan bahwa nilai input akan disimpan langsung ke alamat memori variabel tersebut.

7. `switch { ... }`

Struktur `switch` dalam program ini digunakan untuk menentukan tarif parkir berdasarkan kombinasi jenis kendaraan dan durasi parkir. Tidak seperti `switch` biasa yang memeriksa satu nilai, `switch` ini menggunakan kondisi-kondisi boolean di setiap casenya. Dengan cara ini, program dapat mengecek beberapa syarat sekaligus, seperti jenis kendaraan tertentu dan durasi parkir dalam rentang tertentu.

8. `case kendaraan == "Motor" && durasi >= 1 && durasi <= 2`

Case pertama menangani kondisi ketika kendaraan yang diparkir adalah Motor dan durasi parkir berada dalam rentang satu hingga dua jam. Jika kedua syarat ini terpenuhi, maka tarif parkir ditetapkan sebesar 7000 rupiah. Penggunaan operator logika `&&` memastikan bahwa kedua kondisi harus sama-sama benar .

9. `case kendaraan == "Motor" && durasi > 2`

Case ini menangani kondisi ketika kendaraan yang diparkir masih Motor tetapi durasi parkirnya lebih dari dua jam. Dalam kondisi ini, tarif berbeda akan diterapkan yaitu sebesar 9000 rupiah. Dengan adanya case terpisah ini, program mampu memberikan tarif yang berbeda antara durasi singkat dan durasi panjang untuk kendaraan Motor. Penggunaan operator logika `&&` memastikan bahwa kedua kondisi harus sama-sama benar .

10. `case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2`

Case berikutnya menangani jenis kendaraan Mobil dengan durasi satu hingga dua jam. Jika kondisi ini terpenuhi, maka tarif parkir untuk Mobil dalam rentang durasi tersebut adalah 15000 rupiah. Penggunaan operator logika `&&` memastikan bahwa kedua kondisi harus sama-sama benar .

11. `case kendaraan == "Mobil" && durasi > 2`

Case ini menangani kendaraan Mobil dengan durasi parkir lebih dari dua jam. Dalam situasi ini, tarif parkir meningkat menjadi 20000 rupiah. Penggunaan operator logika `&&` memastikan bahwa kedua kondisi harus sama-sama benar.

12. case kendaraan == "Truk" && durasi >= 1 && durasi <= 2

Case ini digunakan ketika kendaraan yang diparkir adalah Truk dengan durasi satu hingga dua jam. Truk, sebagai kendaraan besar, memiliki tarif khusus yang lebih mahal, yakni 25000 rupiah. Penggunaan operator logika && memastikan bahwa kedua kondisi harus sama-sama benar .

13. case kendaraan == "Truk" && durasi > 2

Case terakhir sebelum default mengatur tarif untuk Truk dengan durasi parkir lebih dari dua jam. Dalam kondisi ini, tarif yang dikenakan adalah 35000 rupiah, tarif tertinggi dalam semua kategori. Penggunaan operator logika && memastikan bahwa kedua kondisi harus sama-sama benar .

14. default: fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")

Bagian default merupakan penanganan ketika input pengguna tidak sesuai dengan seluruh case sebelumnya. Jika pengguna memasukkan jenis kendaraan yang tidak dikenal atau durasi parkir yang kurang dari satu jam, program akan menampilkan pesan bahwa input tidak valid. Default ini berfungsi sebagai jaring pengaman agar program tetap memberikan respons yang jelas meskipun terjadi kesalahan input.

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

Baris terakhir ini digunakan untuk menampilkan hasil akhir berupa tarif parkir yang telah dihitung berdasarkan case-case sebelumnya. Dengan menggunakan `fmt.Printf`, program dapat menampilkan teks sekaligus menampilkan nilai variabel tarif secara rapi menggunakan placeholder `%d`. Ini merupakan bagian penutup yang memberi pengguna informasi final berupa jumlah tarif yang harus dibayarkan.

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 tidak valid, rentang pH 0 - 14")

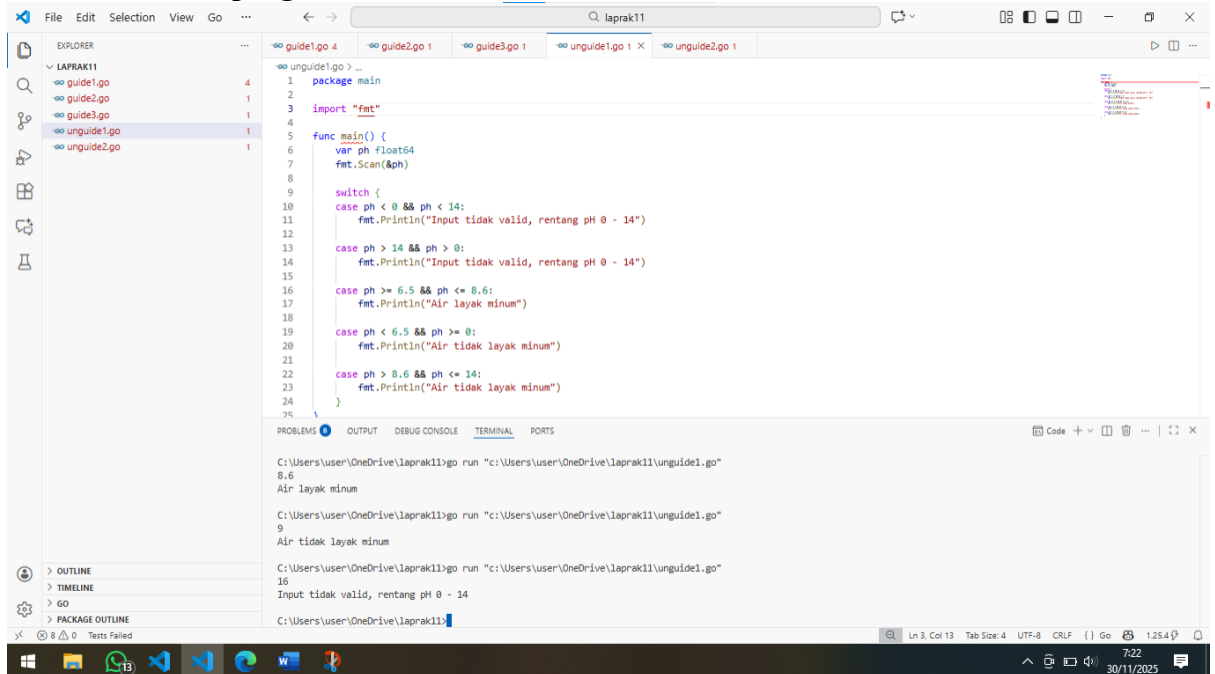
    case ph > 14 && ph > 0:
        fmt.Println("Input tidak valid, rentang pH 0 - 14")

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

    case ph < 6.5 && ph >= 0:
        fmt.Println("Air tidak layak minum")

    case ph > 8.6 && ph <= 14:
        fmt.Println("Air tidak layak minum")
    }
}
```

Screenshoot program



```
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("Input tidak valid, rentang pH 0 - 14")
12
13    case ph > 14 && ph > 0:
14        fmt.Println("Input tidak valid, rentang pH 0 - 14")
15
16    case ph >= 6.5 && ph <= 8.6:
17        fmt.Println("Air layak minum")
18
19    case ph < 6.5 && ph >= 8.6:
20        fmt.Println("Air tidak layak minum")
21
22    case ph > 8.6 && ph <= 14:
23        fmt.Println("Air tidak layak minum")
24    }
25 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

C:\Users\user\OneDrive\laprak11> go run "C:\Users\user\OneDrive\laprak11\unguide1.go"

8.6

Air layak minum

C:\Users\user\OneDrive\laprak11> go run "C:\Users\user\OneDrive\laprak11\unguide1.go"

9

Air tidak layak minum

C:\Users\user\OneDrive\laprak11> go run "C:\Users\user\OneDrive\laprak11\unguide1.go"

16

Input tidak valid, rentang pH 0 - 14

C:\Users\user\OneDrive\laprak11>

Deskripsi program

1. package main

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

2. import "fmt"

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

3. func main()

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

4. Tanda buka dan tutup kurung kurawal { }

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

5. var ph float64

Baris `var ph float64` mendeklarasikan variabel bernama `ph` dengan tipe data `float64`. Variabel ini berfungsi sebagai tempat penyimpanan nilai pH yang akan dimasukkan.

6. fmt.Scan(&ph)

Instruksi `fmt.Scan(&ph)` digunakan untuk membaca input dari pengguna dan menyimpannya ke dalam variabel `ph`. Penggunaan tanda `&` berarti bahwa fungsi `Scan` akan menempatkan hasil input langsung ke alamat memori variabel tersebut.

7. switch { ... }

Struktur `switch { ... }` pada program ini merupakan bentuk `switch` tanpa ekspresi, yang berarti setiap case di dalamnya berupa kondisi boolean. Setiap case akan diperiksa dari atas ke bawah, dan program menjalankan case pertama yang bernilai benar.

8. case ph < 0 && ph < 14

Pada case pertama, program memeriksa apakah nilai pH kurang dari 0. Kondisi ini digunakan untuk menangani input yang berada di bawah batas minimum rentang pH, yaitu 0. Jika pengguna memasukkan pH negatif, maka hasil tersebut langsung dianggap tidak valid. Karena itu, program menampilkan pesan “Input tidak valid, rentang pH 0 – 14”.

9. case ph > 14 && ph > 0

Case kedua mengecek apakah nilai pH lebih besar dari 14. Sama seperti sebelumnya, pH yang berada di atas 14 juga dianggap tidak valid, karena rentang pH secara umum hanya 0 hingga 14. Jika pengguna memasukkan angka seperti 15 atau 20, program langsung memberikan pesan bahwa input tidak valid.

10. case ph >= 6.5 && ph <= 8.6

Case ini menangani kondisi ketika nilai pH berada dalam rentang 6.5 hingga 8.6. Rentang ini merupakan standar umum pH air yang masih aman dan layak untuk dikonsumsi. Jika nilai pH memenuhi kondisi tersebut, program menampilkan “Air layak minum”.

11. case $ph < 6.5 \ \&\& \ ph \geq 0$

Case berikutnya menangani nilai pH yang kurang dari 6.5 tetapi tetap berada di atas atau sama dengan 0. Kondisi ini menunjukkan bahwa air bersifat terlalu asam. Air dengan tingkat keasaman tinggi tidak direkomendasikan untuk dikonsumsi karena dapat memengaruhi kesehatan. Oleh karena itu, program menampilkan pesan “Air tidak layak minum” .

12. case $ph > 8.6 \ \&\& \ ph \leq 14$

Case terakhir memeriksa apakah pH lebih dari 8.6 namun masih berada dalam batas maksimum 14. Nilai pH yang lebih tinggi dari 8.6 menunjukkan bahwa air bersifat terlalu basa. Air yang terlalu basa juga tidak layak dikonsumsi. Karena itu, program memberikan hasil yang sama seperti sebelumnya yaitu “Air tidak layak minum”.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var jenis string
    var durasi int
    var tarifPerJam int
    var total int

    fmt.Scan(&jenis)
    fmt.Scan(&durasi)

    if durasi < 1 {
        durasi = 1
    }
    switch jenis {
    case "motor":
        tarifPerJam = 2000
    case "mobil":
        tarifPerJam = 5000
    case "truk":
        tarifPerJam = 8000
    default:
        tarifPerJam = 0
    }

    total = tarifPerJam * durasi
    fmt.Println("Rp", total)
}
```

Screenshoot program

```
1 package main
2 import "fmt"
3 func main() {
4     var jenis string
5     var durasi int
6     var tarifPerJam int
7     var total int
8
9     fmt.Scan(&jenis)
10    fmt.Scan(&durasi)
11
12    if durasi < 1 {
13        durasi = 1
14    }
15    switch jenis {
16    case "motor":
17        tarifPerJam = 2000
18    case "mobil":
19        tarifPerJam = 5000
20    case "truk":
21        tarifPerJam = 8000
22    default:
23        tarifPerJam = 0
24    }
25 }
```

C:\Users\user\OneDrive\laprak11> go run "C:\Users\user\OneDrive\laprak11\unguide2.go"

motor 3 jam
Rp 6000

C:\Users\user\OneDrive\laprak11> go run "C:\Users\user\OneDrive\laprak11\unguide2.go"

mobil 1 jam
Rp 5000

C:\Users\user\OneDrive\laprak11> go run "C:\Users\user\OneDrive\laprak11\unguide2.go"

truk 5 jam
Rp 40000

C:\Users\user\OneDrive\laprak11>

Deskripsi program

1. package main

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

2. import "fmt"

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

3. func main()

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

4. Tanda buka dan tutup kurung kurawal { }

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

5. Deklarasi variabel: jenis, durasi, tarifPerJam, total

Baris deklarasi variabel seperti `var jenis string`, `var durasi int`, `var tarifPerJam int`, dan `var total int` digunakan untuk membuat empat variabel yang dibutuhkan program. Variabel `jenis` menyimpan jenis kendaraan, sedangkan `durasi` menyimpan lama parkir dalam bentuk bilangan bulat. Variabel `tarifPerJam` digunakan untuk menampung tarif berdasarkan jenis kendaraan yang dimasukkan, dan variabel `total` akan menyimpan hasil akhir perhitungan tarif parkir.

6. `fmt.Scan(&jenis)` dan `fmt.Scan(&durasi)`

Perintah `fmt.Scan(&jenis)` dan `fmt.Scan(&durasi)` digunakan untuk membaca input dari pengguna. Variabel `jenis` akan diisi dengan teks seperti "motor", "mobil", atau "truk", sedangkan `durasi` akan diisi dengan angka lama parkir. Penggunaan tanda `&` berarti bahwa nilai input dimasukkan langsung ke alamat memori variabel tersebut.

7. `if durasi < 1 { durasi = 1 }`

Baris ini digunakan sebagai validasi agar durasi parkir tidak kurang dari satu jam. Jika pengguna memasukkan durasi 0 atau angka negatif, program otomatis mengubah durasi menjadi 1. Hal ini dilakukan karena durasi parkir minimal adalah satu jam dan tidak diperbolehkan bernilai kurang dari itu.

8. `switch jenis { ... }`

Struktur `switch jenis` digunakan untuk menentukan tarif parkir berdasarkan jenis kendaraan. Setiap `case` mewakili satu jenis kendaraan yang valid. Jika input sesuai dengan salah satu pilihan, program akan memasukkan nilai tarif ke variabel `tarifPerJam` sesuai ketentuan.

9. `case "motor"`, `case "mobil"`, dan `case "truk"`

Pada bagian ini, setiap `case` menentukan tarif per jam untuk masing-masing jenis kendaraan. Jika pengguna memasukkan "motor", tarif per jam diberi nilai 2000. Untuk "mobil", tarif adalah 5000 per jam, dan untuk "truk", tarif ditetapkan sebesar 8000 per jam.

10. default: tarifPerJam = 0

Bagian default pada switch menangani kondisi ketika pengguna memasukkan jenis kendaraan yang tidak dikenal, misalnya “sepeda” atau “pesawat”. Jika jenis kendaraan tidak sesuai dengan pilihan yang valid, tarifPerJam diisi dengan nilai 0.

11. total = tarifPerJam * durasi

Baris ini melakukan perhitungan akhir tarif parkir. Total biaya dihitung dengan cara mengalikan tarif per jam dengan durasi parkir yang telah ditentukan. Perhitungan ini baru dapat dilakukan setelah tarif per jam ditentukan melalui switch dan durasi divalidasi. Nilai total kemudian disimpan di variabel total untuk ditampilkan kepada pengguna.

12. fmt.Println("Rp", total)

Perintah `fmt.Println("Rp", total)` digunakan untuk menampilkan hasil akhir tarif parkir kepada pengguna. Program akan mencetak teks “Rp” diikuti dengan nilai total yang telah dihitung sebelumnya. Baris ini menutup proses program dengan memberikan output yang jelas sesuai input jenis kendaraan dan durasi parkir.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)

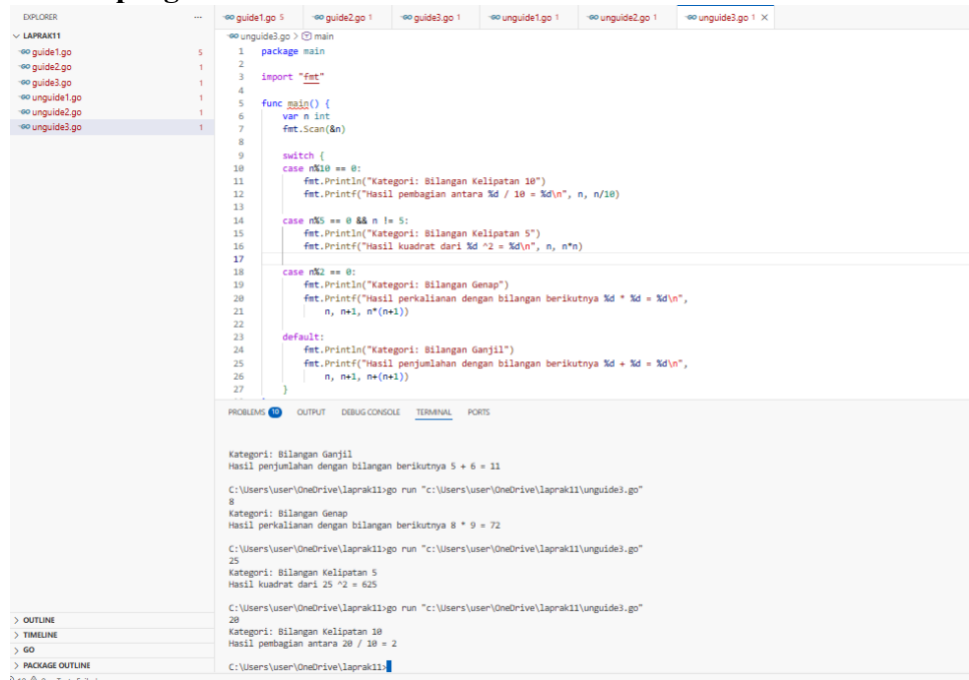
    switch {
    case n%10 == 0:
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, n/10)

    case n%5 == 0 && n != 5:
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", n, n*n)

    case n%2 == 0:
        fmt.Println("Kategori: Bilangan Genap")
        fmt.Printf("Hasil perkalianan dengan bilangan berikutnya %d * %d = %d\n", n, n+1, n*(n+1))

    default:
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, n+1, n+(n+1))
    }
}
```


Screenshoot program



The screenshot shows a Go program in an IDE. The Explorer pane on the left shows a project named 'LAPRAK11' with files 'guide1.go', 'guide2.go', 'guide3.go', 'unguide1.go', 'unguide2.go', and 'unguide3.go'. The main editor shows the source code of 'unguide3.go'.

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

The terminal pane at the bottom shows the output of running the program three times:

```
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11

C:\Users\user\OneDrive\laprak11> go run "c:\Users\user\OneDrive\laprak11\unguide3.go"
8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72

C:\Users\user\OneDrive\laprak11> go run "c:\Users\user\OneDrive\laprak11\unguide3.go"
25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 * 2 = 625

C:\Users\user\OneDrive\laprak11> go run "c:\Users\user\OneDrive\laprak11\unguide3.go"
20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2

C:\Users\user\OneDrive\laprak11>
```

Deskripsi program

1. package main

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

2. import "fmt"

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

3. func main()

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

4. Tanda buka dan tutup kurung kurawal { }

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

5. `var n int`

Baris `var n int` digunakan untuk mendeklarasikan sebuah variabel bernama `n` yang bertipe integer. Variabel ini berfungsi untuk menampung nilai angka yang dimasukkan.

6. `fmt.Scan(&n)`

Baris ini berfungsi untuk membaca input angka dari pengguna dan menyimpannya ke dalam variabel `n`. Penggunaan tanda `&` berarti bahwa input disimpan langsung ke alamat memori variabel tersebut.

7. `switch { ... }`

Struktur `switch { ... }` pada program ini digunakan untuk melakukan pengecekan kondisi berdasarkan beberapa aturan. Tidak seperti `switch` pada umumnya yang memeriksa satu variabel, `switch` ini berisi kondisi logis yang bebas untuk menentukan kategori bilangan berdasarkan nilai `n`. Program akan memeriksa setiap case secara berurutan dan menjalankan case pertama yang bernilai benar.

8. `case n%10 == 0`

Pada case pertama, program memeriksa apakah bilangan tersebut merupakan kelipatan 10 dengan mengecek apakah sisa hasil bagi `n` dengan 10 sama dengan 0. Jika kondisi ini terpenuhi, program menampilkan kategori “Bilangan Kelipatan 10”. Tidak hanya itu, program juga menampilkan hasil pembagian `n` dengan 10 menggunakan `fmt.Printf`. Bagian ini menjadi case dengan prioritas tertinggi karena kelipatan 10 juga merupakan kelipatan 5—oleh sebab itu case ini harus muncul paling awal agar kondisi yang lebih spesifik tertangani terlebih dahulu.

9. `case n%5 == 0 dan n != 5`

Case kedua akan dijalankan jika bilangan tersebut merupakan kelipatan 5 tetapi bukan bilangan 5 itu sendiri. Pengecekan dilakukan menggunakan kondisi `n%5 == 0 dan n != 5`. Jika kondisi ini benar, program mengategorikan angka tersebut sebagai “Bilangan Kelipatan 5”. Program juga menghitung hasil kuadrat dari angka tersebut dan menampilkannya.

10. case $n \% 2 == 0$

Case ketiga dijalankan ketika bilangan yang dimasukkan adalah bilangan genap. Pengecekan dilakukan dengan $n \% 2 == 0$, yang menunjukkan bahwa angka tersebut habis dibagi 2. Ketika kondisi ini terpenuhi, program menampilkan kategori “Bilangan Genap” dan melakukan perhitungan berupa perkalian antara angka tersebut dengan bilangan berikutnya ($n * (n+1)$). Hasilnya kemudian ditampilkan menggunakan `fmt.Printf`.

11. default

Bagian default akan dijalankan ketika semua kondisi sebelumnya tidak terpenuhi. Kondisi yang tersisa hanya satu, yaitu bilangan ganjil yang bukan kelipatan 5 atau kelipatan 10. Program memasukkan bilangan tersebut ke dalam kategori “Bilangan Ganjil”. Setelah itu, program menampilkan hasil penjumlahan antara bilangan tersebut dengan bilangan setelahnya ($n + (n+1)$). Bagian default ini memastikan bahwa setiap nilai input tetap mendapatkan kategori dan hasil perhitungan yang sesuai

