

LAPORAN PRAKTIKUM

Algoritma Pemrograman

MODUL 11

SWITCH CASE



DISUSUN OLEH:

ERLINA DWI SETYANI

103112430229

S1IF-12-07

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

DASAR TEORI

Switch-case adalah struktur kontrol dalam pemrograman yang digunakan untuk membuat keputusan berdasarkan nilai ekspresi tertentu. Struktur ini memungkinkan pengembang untuk mengevaluasi satu ekspresi dan mengeksekusi blok kode tertentu berdasarkan hasil evaluasi tersebut. **Switch case** sering dianggap lebih rapi dan mudah dibaca dibandingkan **string if-else** yang panjang jika ada banyak kondisi yang harus dibandingkan.

TUGAS PENDAHULUAN

1. Tugas 1

Pseudo code

```
Program Kategori

Kamus

var usia: int

Algoritma:

output ("Masukan Usia: ")

input (usia)

depend on usia

usia < 13:

output ("Anak-anak")

usia < 20:

output ("Remaja")

usia < 60:

output ("Dewasa")

usia 60:

output ("Lansia")

default:

output ("Tidak terdaftar dalam kategori")

end
```

```
endprogram
```

Source code

```
package main

import "fmt"

func main() {

    var usia int

    fmt.Println("Masukkan usia: ")

    fmt.Scan(&usia)

    switch {

    case usia < 13:

        fmt.Println("Anak-Anak")

    case usia < 20:

        fmt.Println("Remaja")

    case usia < 60:

        fmt.Println("Dewasa")

    case usia >= 60:

        fmt.Println("Lansia")

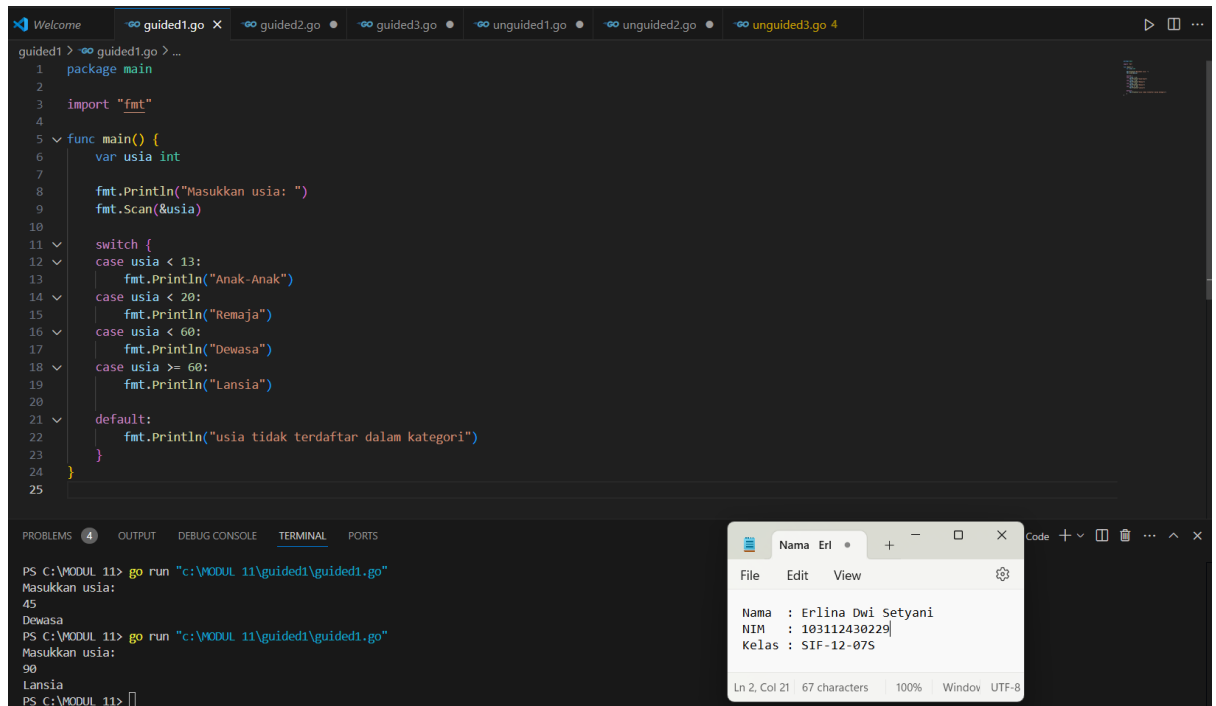
    default:

        fmt.Println("usia tidak terdaftar dalam kategori")

    }

}
```

Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is a simple age categorization tool. It prompts the user to enter their age and then prints the corresponding category based on a switch statement. The terminal shows the program being run twice: first with age 45, resulting in 'Dewasa', and then with age 90, resulting in 'Lansia'.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var usia int
7
8     fmt.Println("Masukkan usia: ")
9     fmt.Scan(&usia)
10
11     switch {
12     case usia < 13:
13         fmt.Println("Anak-Anak")
14     case usia < 20:
15         fmt.Println("Remaja")
16     case usia < 60:
17         fmt.Println("Dewasa")
18     case usia >= 60:
19         fmt.Println("Lansia")
20
21     default:
22         fmt.Println("usia tidak terdaftar dalam kategori")
23     }
24 }
25
```

Terminal Output:

```
PS C:\MODUL 11> go run "c:\MODUL 11\guided1\guided1.go"
Masukkan usia:
45
Dewasa
PS C:\MODUL 11> go run "c:\MODUL 11\guided1\guided1.go"
Masukkan usia:
90
Lansia
PS C:\MODUL 11>
```

Deskripsi program

Kode GO diatas meminta pengguna untuk memasukkan usia dan kemudian menampilkan kategori usia berdasarkan input tersebut.

Switch:

Kondisi 1: Jika usia kurang dari 13 tahun, program akan menampilkan "Anak-anak".

Kondisi 2: Jika usia kurang dari 20 tahun, program akan menampilkan "Remaja".

Kondisi 3: Jika usia kurang dari 60 tahun, program akan menampilkan "Dewasa".

Kondisi 4: Jika usia lebih dari atau sama dengan 60 tahun, program akan menampilkan "Lansia".

Default: Jika usia tidak memenuhi ketentuan di atas, program akan menampilkan "usia tidak terdaftar dalam kategori".

Program ini adalah contoh penggunaan perubahan dalam bahasa Go. Program ini menunjukkan cara memproses input pengguna dan memberikan output berdasarkan aturan yang telah ditentukan.

2. Tugas 2

Pseudo code

Program Hari

Kamus

```
var angka int

output ("Masukan angka: ")

input (angka)

depend on angka |

case

output ("senin")

case 2

output ("Selasa")

case 3:

output ("Rabu")

case 4:

output ("Kamis")

case 5:

output ("Jumat")

case 6

output ("Sabtu")

case 7:

output ("Minggu")

end

endprogram
```

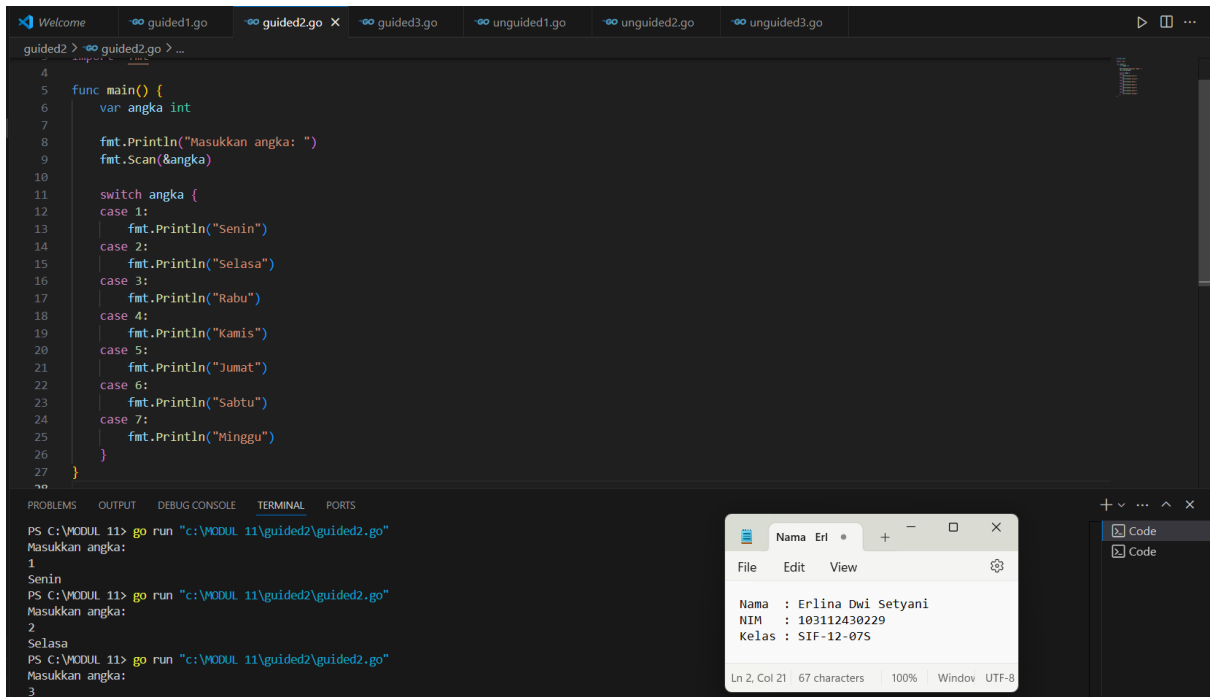
Source code

```
package main

import "fmt"
```

```
func main() {  
    var angka int  
  
    fmt.Println("Masukkan angka: ")  
    fmt.Scan(&angka)  
  
    switch angka {  
    case 1:  
        fmt.Println("Senin")  
    case 2:  
        fmt.Println("Selasa")  
    case 3:  
        fmt.Println("Rabu")  
    case 4:  
        fmt.Println("Kamis")  
    case 5:  
        fmt.Println("Jumat")  
    case 6:  
        fmt.Println("Sabtu")  
    case 7:  
        fmt.Println("Minggu")  
    }  
}
```

Screenshoot program



```
4
5 func main() {
6     var angka int
7
8     fmt.Println("Masukkan angka: ")
9     fmt.Scan(&angka)
10
11     switch angka {
12     case 1:
13         fmt.Println("Senin")
14     case 2:
15         fmt.Println("Selasa")
16     case 3:
17         fmt.Println("Rabu")
18     case 4:
19         fmt.Println("Kamis")
20     case 5:
21         fmt.Println("Jumat")
22     case 6:
23         fmt.Println("Sabtu")
24     case 7:
25         fmt.Println("Minggu")
26     }
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\MODUL 11> go run "c:\MODUL 11\guided2\guided2.go"

Masukkan angka:

1

Senin

PS C:\MODUL 11> go run "c:\MODUL 11\guided2\guided2.go"

Masukkan angka:

2

Selasa

PS C:\MODUL 11> go run "c:\MODUL 11\guided2\guided2.go"

Masukkan angka:

3

Rabu

Deskripsi program

Kode ini adalah program Go yang menampilkan hari dalam seminggu berdasarkan angka yang dimasukkan pengguna. Program meminta pengguna untuk data numerik dengan `fmt.Println("Masukkan angka: ")`.

Input angka disimpan dalam variabel `angka`.

Program ini menggunakan `switch` untuk memeriksa nilai angka dan menampilkan hari yang bersangkutan dari kasus :

- 1: menampilkan "Senin".
- 2: menampilkan "Selasa".
- 3: menampilkan "Rabu".
- 4: menampilkan "Kamis".
- 5: menampilkan "Jumat".
- 6: menampilkan "Sabtu".

kotak 7: menampilkan "Minggu".

input dan output: masukkan angka yang dimasukkan oleh pengguna.

Output : Nama hari dalam seminggu sesuai angka yang dimasukkan.

3. Tugas 3

Pseudo code

```
Program Kode

Kamus

Var

kode: string

output ("Masukan Kode")

input (&kode)

depend on

case kode "G":

output ("Semua usia")

case kode "PG".

output ("Anak-anak di atas 7 tahun")

case kode "PG13"

output ("Anak-anak di atas 13 tahun")

case kode "R":

output ("Dewasa")

default:

output ("Kode tidak valid")

end

endprogram
```

Source code

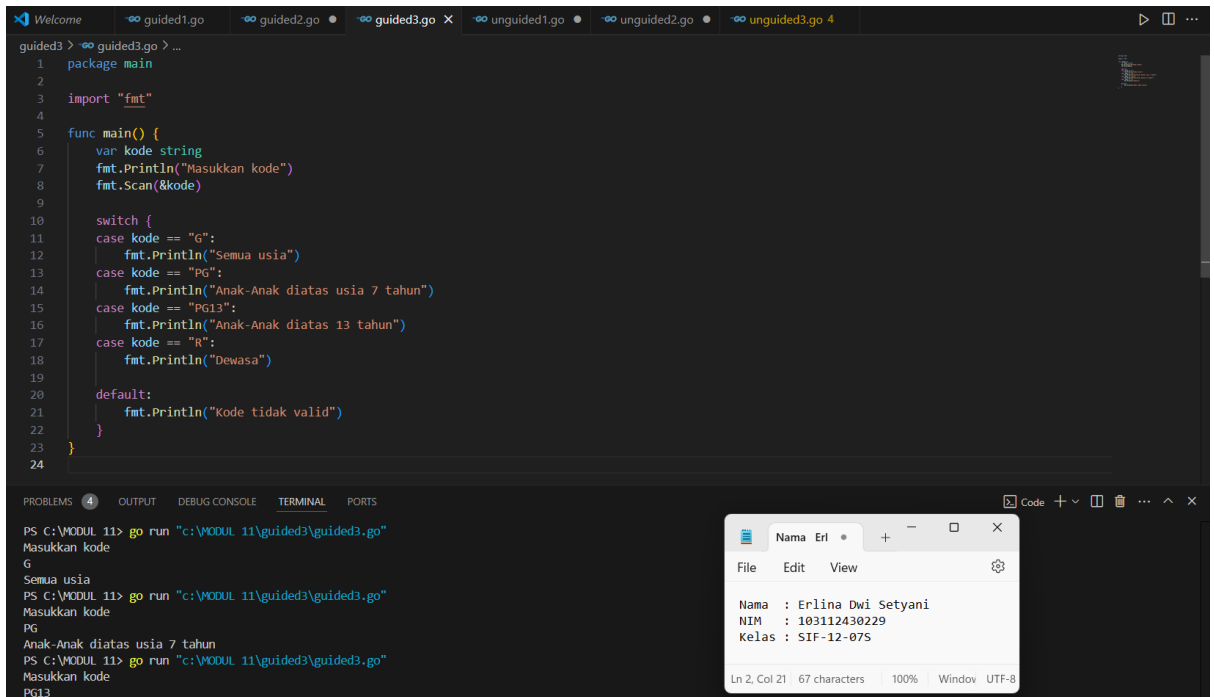
```
package main

import "fmt"
```



```
func main() {  
    var kode string  
    fmt.Println("Masukkan kode")  
    fmt.Scan(&kode)  
  
    switch {  
    case kode == "G":  
        fmt.Println("Semua usia")  
    case kode == "PG":  
        fmt.Println("Anak-Anak diatas usia 7 tahun")  
    case kode == "PG13":  
        fmt.Println("Anak-Anak diatas 13 tahun")  
    case kode == "R":  
        fmt.Println("Dewasa")  
  
    default:  
        fmt.Println("Kode tidak valid")  
    }  
}
```

Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is a simple switch statement that takes a code input and prints a corresponding message. The terminal shows the program being run multiple times with different inputs, and the output matches the expected results.

```
guided3 > go guided3.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kode string
7     fmt.Println("Masukkan kode")
8     fmt.Scan(&kode)
9
10    switch {
11    case kode == "G":
12        fmt.Println("Semua usia")
13    case kode == "PG":
14        fmt.Println("Anak-Anak diatas usia 7 tahun")
15    case kode == "PG13":
16        fmt.Println("Anak-Anak diatas 13 tahun")
17    case kode == "R":
18        fmt.Println("Dewasa")
19
20    default:
21        fmt.Println("Kode tidak valid")
22    }
23 }
24
```

Terminal Output:

```
PS C:\MODUL 11> go run "c:\MODUL 11\guided3\guided3.go"
Masukkan kode
G
Semua usia
PS C:\MODUL 11> go run "c:\MODUL 11\guided3\guided3.go"
Masukkan kode
PG
Anak-Anak diatas usia 7 tahun
PS C:\MODUL 11> go run "c:\MODUL 11\guided3\guided3.go"
Masukkan kode
PG13
```

Deskripsi program

Ini adalah program Go dengan fitur-fitur berikut: Fungsi program Program meminta pengguna memasukkan kode. Berdasarkan pada kode yang dimasukkan, program mencetak pesan yang sesuai:

"G" : "Semua umur".

"PG" : "Anak di atas 7 tahun".

"PG13" : "Anak-anak di atas 13 tahun".

"R" : "Dewasa".

"O" : "Kode Tidak Valid".

Program ini mendefinisikan fungsi bernama main yang merupakan titik masuk untuk eksekusi. Pernyataan switch memeriksa nilai variabel kode dan mengeksekusi kasus yang sesuai. Case default menangani semua input lain yang tidak tercakup dalam pernyataan kasus. Output program ditampilkan di terminal, menunjukkan hasil pemanggilan fungsi Println. Ada beberapa contoh eksekusi program dengan kode masukan berbeda, yang menampilkan keluaran. Program ini menggunakan paket fmt untuk operasi input/output. Program ini mendemonstrasikan penggunaan pernyataan switch untuk eksekusi bersyarat.

LATIHAN MODUL

1. Tugas 1

Pseudo code

```
Program Menghitung pH

var karpH: float64

input (&karpH)

depend on:

case karpH >= 6.5 && karpH <= 8.6:

output ("Air layak minum")

case karpH < 6.5 || karpH > 8.6 && karpH <14:

output ("Air tidak layak minum")

case karpH > 14:

output ("Nilai pH tidak valid, Nilai harus antara 0 dan 14 ")

end

endprogram
```

Source code

```
package main

import "fmt"

func main() {

    var karpH float64

    fmt.Scan(&karpH)

    switch {
```

```

case kadarph >= 6.5 && kadarph <= 8.6:

    fmt.Print("Air layak minum")

case kadarph < 6.5 || kadarph > 8.6 && kadarph < 14:

    fmt.Print("Air tidak layak minum")

case kadarph > 14:

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

}

}

```

Screenshoot program

The screenshot shows a Go program in VS Code. The source code defines a `main` function that reads a pH value and prints a message based on its range. The terminal shows the program being run three times with different inputs, producing the expected output messages.

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var kadarph float64
7     fmt.Scan(&kadarph)
8
9     switch {
10    case kadarph >= 6.5 && kadarph <= 8.6:
11        fmt.Print("Air layak minum")
12    case kadarph < 6.5 || kadarph > 8.6 && kadarph < 14:
13        fmt.Print("Air tidak layak minum")
14    case kadarph > 14:
15        fmt.Print("Nilai pH tidak valid, Nilai harus antara 0 dan 14")
16    }
17 }
18
19

```

```

PS C:\MODULE 11> go run "c:\MODULE 11\unguided1\unguided1.go"
8.6
Air layak minum
PS C:\MODULE 11> go run "c:\MODULE 11\unguided1\unguided1.go"
9
Air tidak layak minum
PS C:\MODULE 11> go run "c:\MODULE 11\unguided1\unguided1.go"
16
Nilai pH tidak valid, Nilai harus antara 0 dan 14

```

Output window details:

```

Nama : Erlina Dwi Setyani
NIM : 103112430229
Kelas : SIF-12-07S

```

Ln 2, Col 21 | 67 characters | 100% | Window | UTF-8

Deskripsi program

Kode Golang di atas mendefinisikan program yang meminta data berupa nilai pH kemudian menampilkan output berdasarkan kategori nilai pH tersebut.

Penjelasan:

package main : Kode ini menyatakan bahwa program ini adalah program utama.

Impor "fmt": Kode ini mengimpor paket "fmt" yang berisi fungsi untuk format output dan input.

Func main() { : Kode ini mendefinisikan fungsi utama yang akan dijalankan saat program dijalankan.

var float64 tingkat pH : Kode ini mendeklarasikan variabel bernama "kadarPH" dengan tipe data float64 untuk menampung nilai pH masukan.

fmt.Scan(&kadarPH): Kode ini membaca input pengguna dan menyimpannya dalam variabel "kadarPH".

Switch: Kode ini adalah instruksi yang mengevaluasi nilai "tingkat pH" dan mengeksekusi blok kode yang sesuai dengan nilai tersebut.

case kadarPH >= 6.5 && kadarPH <= 8.6. Kode ini memeriksa apakah nilai "kadarPH" antara 6.5 dan 8.6. Jika benar, maka akan muncul pesan "Air layak minum".

case kadarPH <6.5 || kadarPH > 8.6 && kadarPH < 14: Kode ini memeriksa apakah nilai "kadarPH" kurang dari 6.5 atau lebih besar dari 8.6 tetapi kurang dari 14. Jika benar, maka akan muncul pesan "Air tidak layak untuk diminum".

case kadarPH > 14: Kode ini memeriksa apakah nilai "kadarPH" lebih besar dari 14. Jika benar, maka akan muncul pesan "Nilai pH tidak valid, nilainya harus antara 0 atau 14".

Output: Program akan menampilkan pesan tergantung pada kategori nilai pH yang dimasukkan oleh pengguna. Misalnya:

Jika pengguna memasukkan 7.2, program akan menampilkan "Air layak minum".

Jika pengguna memasukkan angka 9, program akan menampilkan "Air tidak layak untuk diminum".

Jika pengguna memasukkan 16, program akan menampilkan "Nilai pH tidak valid, nilai harus antara 0 dan 14".

2. Tugas 2

Pseudo code

Program Parkir

Kamus

var kendaraan: string

var durasi: int

output ("Jenis Kendaraan & Durasi Parkir")

input (&kendaraan, & durasi)

motor: 2000

mobil: 5000

truk := 8000

depend on:

case kendaraan == "motor" && durasi <= 1:

output (motor)

case kendaraan == "motor" && durasi >= 2:

output (motor * durasi)

case kendaraan == "mobil" && durasi <= 1:

output (mabil)

case kendaraan "mobil" && durasi >= 2:

output (mobil durasi)

case kendaraan "truk" && durasi <= 1:

output (truk)

case kendaraan "truk" && durasi >= 1:

output (truk durasi)

end

endprogram

Source code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi int

    fmt.Println("Jenis kendaraan & durasi parkir")
    fmt.Scan(&kendaraan, &durasi)

    motor := 2000
    mobil := 5000
    truck := 8000

    switch {
    case kendaraan == "motor" && durasi <= 1:
        fmt.Println(motor)
    case kendaraan == "motor" && durasi >= 2:
        fmt.Println(motor * durasi)
    case kendaraan == "mobil" && durasi <= 1:
        fmt.Println(mobil)
    case kendaraan == "mboil" && durasi >= 2:
        fmt.Println(mobil * durasi)
    case kendaraan == "truck" && durasi <= 1:
        fmt.Println(truck)
    case kendaraan == "truck" && durasi >= 2:
```

```

        fmt.Println(truck * durasi)

    }

}

```

Screenshoot program

The screenshot shows a Go program in VS Code. The source code in `unguided2.go` defines a `main` function that prompts the user for vehicle type and parking duration, then calculates the parking fee based on a switch statement. The terminal shows three successful runs with different inputs.

```

5 func main() {
6     var kendaraan string
7     var durasi int
8     fmt.Println("Jenis kendaraan & durasi parkir")
9     fmt.Scan(&kendaraan, &durasi)
10
11     motor := 2000
12     mobil := 5000
13     truck := 8000
14
15     switch {
16     case kendaraan == "motor" && durasi <= 1:
17         fmt.Println(motor)
18     case kendaraan == "motor" && durasi >= 2:
19         fmt.Println(motor * durasi)
20     case kendaraan == "mobil" && durasi <= 1:
21         fmt.Println(mobil)
22     case kendaraan == "mobil" && durasi >= 2:
23         fmt.Println(mobil * durasi)
24     case kendaraan == "truck" && durasi <= 1:
25         fmt.Println(truck)
26     case kendaraan == "truck" && durasi >= 2:
27         fmt.Println(truck * durasi)
28     }
29 }

```

```

PS C:\MODUL 11> go run "c:\MODUL 11\unguided2\unguided2.go"
Jenis kendaraan & durasi parkir
motor 3 jam
6000
PS C:\MODUL 11> go run "c:\MODUL 11\unguided2\unguided2.go"
Jenis kendaraan & durasi parkir
mobil 1 jam
5000
PS C:\MODUL 11> go run "c:\MODUL 11\unguided2\unguided2.go"
Jenis kendaraan & durasi parkir
truck 5 jam

```

Deskripsi program

Ini adalah program Go yang menghitung biaya parkir berdasarkan jenis kendaraan dan durasi parkir.

Penjelasan:

fungsi utama(): Fungsi ini adalah titik masuk program.
var kendaraan(string): Menyimpan jenis kendaraan (misalnya "sepeda motor", "mobil", "truk").

durasi(int): menyimpan durasi parkir dalam jam.

Masukkan:

Cetak pesan kepada pengguna yang menanyakan jenis kendaraan dan durasi parkir.

Digunakan `fmt.Scan()` untuk membaca input pengguna dan menyimpannya dalam variabel.

Pernyataan Switch: Gunakan switch untuk menentukan biaya parkir berdasarkan jenis dan durasi kendaraan. Kasus: kendaraan == "motor" && durasi <= 1: Jika kendaraannya adalah sepeda motor dan durasinya kurang dari atau sama dengan 1 jam, maka biayanya adalah sebesar 3000.

kendaraan == "motor" && durasi >= 2: Jika kendaraannya sepeda motor dan durasinya 2 jam atau lebih, biayanya adalah 3000 dikalikan durasi.

kendaraan == "mobil" && durasi <= 1: Jika kendaraannya mobil dan durasinya kurang dari atau sama dengan 1 jam, biayanya adalah sebesar 5000.

kendaraan == "car" && durasi >= 2: Jika kendaraannya mobil dan durasinya 2 jam atau lebih, biayanya 5000 dikali durasi.

kendaraan == "truck" && durasi <= 1: Jika kendaraannya truk dan durasinya kurang dari atau sama dengan 1 jam, maka biayanya adalah truck 8000.

kendaraan == "truck" && durasi >= 2: Jika kendaraannya truk dan durasinya 2 jam atau lebih, biayanya adalah truck 8000 dikalikan durasi.

Output: Program menampilkan perhitungan biaya parkir. Program ini menggunakan pernyataan switch untuk menangani banyak kasus secara efisien dan memberikan hasil yang jelas kepada pengguna. Program ini dapat diperluas untuk mencakup jenis kendaraan lain, biaya parkir berbeda.

3. Tugas 3

Pseudo code

```
Program operasi MTK

Kamus

var input: int

output ("Masukan Angka: ")

input (&input)

depend on:

case input%2!= 0 && input <25:

hasil: input + (input + 1)
```

```

output ("Kategori: bilangan ganjil\nHasil penjumlahan dengan
bilangan berikutnya %d + %d = %d\n", input, input+1, hasil)

case input%2 0 && input%5!= 0 && input%10! 0:

hasil input (input + 1)

output ("Kategori: bilangan genap\nHasil perkalian dengan
bilangan berikutnya %d %d = %d\n", input, input+1, hasil)

case input%50 && input%10 != 0:

Hasil (input * input)

fmt.Printf("Hasil kuadrat dari 5\nHasil kuadrat dari %d ^2-
%d", input, Hasil)

case input%10-0:

hasil: input/10

output ("Kategori bilangan kelipatan 10 \nHasil pembagian
antara %d/%d %d", input, input/10, hasil)

end

endprogram

```

Source code

```

package main

import "fmt"

func main() {

    var input int

    fmt.Print("Masukkan angka: ")

    fmt.Scan(&input)

```

```
switch {

case input%2 != 0 && input < 25:

    hasil := input + (input + 1)

    fmt.Printf("Kategori: bilangan ganjil\nhasil penjumlahan
dengan bilangan berikutnya %d + %d = %d\n", input, input+1, hasil)


case input%2 == 0 && input%5 != 0 && input%10 != 0:

    hasil := input * (input + 1)

    fmt.Printf("Kategori: bilangan genap\nhasil perkalian
bilangan berikutnya %d * %d = %d\n", input, input+1, hasil)


case input%5 == 0 && input%10 != 0:

    hasil := (input * input)

    fmt.Printf("Hasil kuadrat dari 5\nnhasil kuadrat dari %d^2
== %d", input, hasil)


case input%10 == 0:

    hasil := input / 10

    fmt.Printf("Kategori bilangan kelipatan 10\nhasil pembagian
antara %d/%d = %d", input, input/10, hasil)


}

}
```

Screenshoot program

```
4 func main() {
5     var input int
6
7     fmt.Print("Masukkan angka: ")
8     fmt.Scan(&input)
9
10
11     switch {
12     case input%2 != 0 && input < 25:
13         hasil := input + (input + 1)
14         fmt.Printf("Kategori: bilangan ganjil\nhasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", input, input+1, hasil)
15
16     case input%2 == 0 && input%5 != 0 && input%10 != 0:
17         hasil := input * (input + 1)
18         fmt.Printf("Kategori: bilangan genap\nhasil perkalian bilangan berikutnya %d * %d = %d\n", input, input+1, hasil)
19
20     case input%5 == 0 && input%10 != 0:
21         hasil := (input * input)
22         fmt.Printf("Hasil kuadrat dari 5\nhasil kuadrat dari %d^2 = %d", input, hasil)
23
24     case input%10 == 0:
25         hasil := input / 10
26         fmt.Printf("Kategori bilangan kelipatan 10\nhasil pembagian antara %d/%d = %d", input, input/10, hasil)
27
28     }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\MODUL 11> go run "c:\MODUL 11\unguided3\unguided3.go"
Masukkan angka: 5
Kategori: bilangan ganjil
hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
%(EXTRA int=5, int=6, int=11)
PS C:\MODUL 11> go run "c:\MODUL 11\unguided3\unguided3.go"
Masukkan angka: 8
Kategori: bilangan genap
hasil penjumlahan dengan bilangan berikutnya 8 + 8 = 16
hasil perkalian bilangan berikutnya 8 * 9 = 72
```

Deskripsi program

kode golangKode ini adalah fungsi utama yang menerima masukan angka dari pengguna.Selanjutnya kode menentukan kategori angka yang dimasukkan.

Ada 4 kategori:

Bilangan ganjil : bilangan yang tidak habis dibagi 2 dan kurang dari 25.

Bilangan genap : Bilangan yang habis dibagi 2, tidak habis dibagi 5, dan tidak habis dibagi 10.

Kuadrat: bilangan habis dibagi 5, tetapi tidak habis dibagi 10.

Kelipatan 10 : Bilangan yang habis dibagi 10.

Kemudian menampilkan hasil perhitungan berdasarkan kategori yang ditentukan.

Contoh hasil:

Jika pengguna menetapkan angka 5, program akan menampilkan "Kategori: angka ganjil" dan hasil penjumlahan angka 5 dan 6.

Jika pengguna memasukkan angka 8, program akan menampilkan Kategori: "bilangan genap" dan hasil perkalian angka 8 dan 9.

Kode ini menggunakan operator modulo (%) untuk memeriksa apakah suatu bilangan habis dibagi suatu bilangan tertentu.Kode ini juga menggunakan fungsi Printf untuk mencetak hasilnya ke layar dalam format tertentu.

