

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 11
Switch Case



Disusun oleh:
RADEN AUREL ADITYA KUSUMAANINGYUN
103112430267
S1IF-12-07

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

DASAR TEORI

Pada dasarnya, switch-case adalah metode alternatif yang digunakan untuk menggantikan penggunaan else-if, dengan perbedaan utama bahwa pemilihan aksi dilakukan berdasarkan nilai atau hasil dari operasi tipe data tertentu, bukan berdasarkan kondisi boolean. Struktur penulisan switch-case terdiri dari beberapa komponen utama. Pertama, ekspresi, yaitu operasi tipe data yang menghasilkan nilai selain boolean. Kedua, nilai, yang merujuk pada pilihan-pilihan yang ada dalam case yang didefinisikan pada instruksi switch-case. Ketiga, aksi, yakni serangkaian perintah yang akan dijalankan jika nilai ekspresi cocok dengan salah satu pilihan case. Jika tidak ada nilai yang sesuai, aksi lain dapat dijalankan dengan menggunakan kata kunci *default*. Selain itu, switch-case juga dapat berfungsi sebagai pengganti struktur kontrol if-then, di mana setiap case yang ditentukan pada switch-case mewakili kondisi yang diuji dalam percabangan.

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

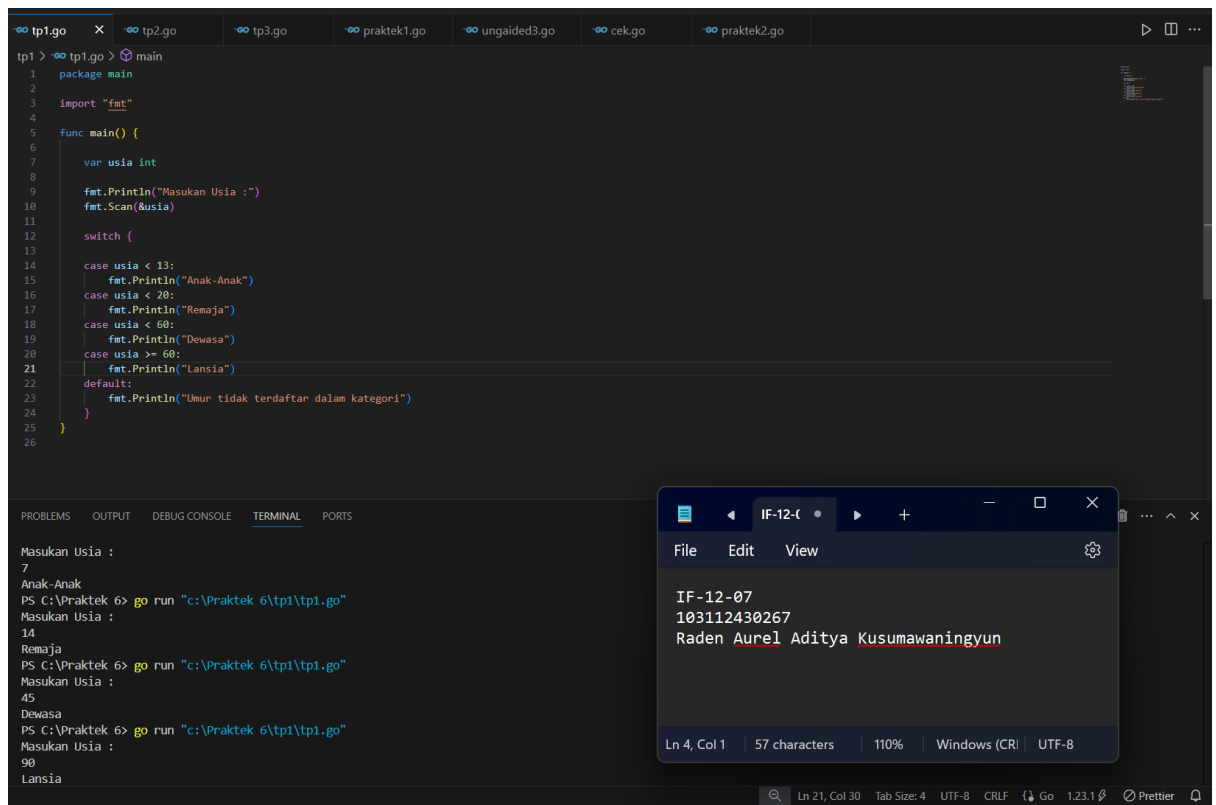
    fmt.Println("Masukan Usia :")
    fmt.Scan(&usia)

    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("Umur tidak terdaftar dalam kategori")
    }

}
```

Screenshoot program



```
tp1 > tp1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var usia int
8
9     fmt.Println("Masukan Usia :")
10    fmt.Scan(&usia)
11
12    switch {
13
14    case usia < 13:
15        fmt.Println("Anak-Anak")
16    case usia < 20:
17        fmt.Println("Remaja")
18    case usia < 60:
19        fmt.Println("Dewasa")
20    case usia >= 60:
21        fmt.Println("Lansia")
22    default:
23        fmt.Println("Umur tidak terdaftar dalam kategori")
24    }
25 }
26
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Masukan Usia :
7
Anak-Anak
PS C:\Praktek 6> go run "c:\Praktek 6\tp1\tp1.go"
Masukan Usia :
14
Remaja
PS C:\Praktek 6> go run "c:\Praktek 6\tp1\tp1.go"
Masukan Usia :
45
Dewasa
PS C:\Praktek 6> go run "c:\Praktek 6\tp1\tp1.go"
Masukan Usia :
90
Lansia

IF-12-07
103112430267
Raden Aurel Aditya Kusumawaningyun

Ln 4, Col 1 | 57 characters | 110% | Windows (CRLF) | UTF-8

Deskripsi program

Pada kode di atas, kita dapat melihat program yang menggunakan bahasa pemrograman Go untuk mengklasifikasikan usia seseorang ke dalam kategori tertentu berdasarkan rentang usia yang dimasukkan. Program dimulai dengan mendeklarasikan paket **main** dan mengimpor paket **fmt**, yang digunakan untuk melakukan input dan output pada program.

Di dalam fungsi **main**, variabel **usia** bertipe data integer dideklarasikan untuk menyimpan nilai usia yang akan dimasukkan oleh pengguna. Kemudian, program meminta pengguna untuk memasukkan usia dengan menampilkan pesan "Masukan Usia : " menggunakan fungsi **fmt.Println**. Setelah itu, fungsi **fmt.Scan(&usia)** digunakan untuk membaca input usia dari pengguna dan menyimpannya dalam variabel **usia**.

Selanjutnya, struktur **switch** digunakan untuk memeriksa nilai dari variabel **usia**. Switch ini tidak memerlukan ekspresi kondisi eksplisit, sehingga program akan memeriksa setiap kasus yang ada. Kasus pertama, **usia < 13**, akan mencetak "Anak-Anak" jika usia yang dimasukkan kurang dari 13. Kasus kedua, **usia < 20**, akan mencetak "Remaja" jika usia kurang dari 20 tetapi lebih besar atau sama dengan 13. Kasus ketiga, **usia < 60**, mencetak "Dewasa" jika usia kurang dari 60. Kasus terakhir, **usia >= 60**, mencetak "Lansia" untuk usia yang lebih dari atau sama dengan 60.

Jika tidak ada kondisi yang terpenuhi, yang sangat jarang terjadi dalam hal ini karena semua rentang usia sudah tercakup, maka perintah di bawah **default** akan dieksekusi. **Default**

digunakan sebagai pengaman jika tidak ada nilai yang sesuai dengan kondisi yang diberikan. Namun, dalam konteks kode ini, *default* jarang terpakai karena semua rentang usia sudah diatur dengan jelas.

Program ini adalah contoh sederhana penggunaan switch-case dalam Go untuk mengkategorikan usia secara otomatis berdasarkan input pengguna.

2. Tugas 2

Pseudo code

```
Program Hari  
  
Kamus  
  
var angka :int  
  
output ("Masukan angka : ")  
  
input (angka)  
  
depend on angka {  
  
  case 1:  
    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"

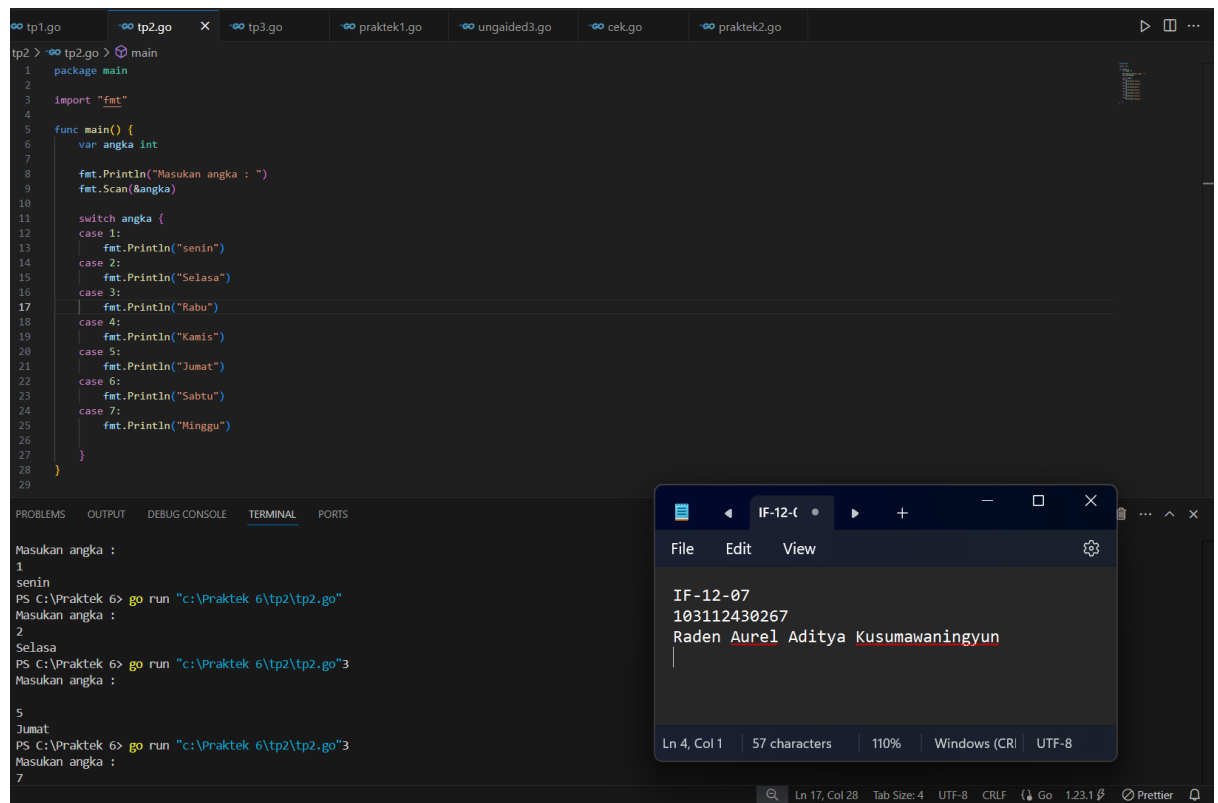
func main() {
    var angka int

    fmt.Println("Masukan 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



```
tp2 > tp2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var angka int
7
8     fmt.Println("Masukan 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 }
28
29
```

Masukan angka : 1
senin
PS C:\Praktek 6> go run "c:\Praktek 6\tp2\tp2.go"
Masukan angka : 2
Selasa
PS C:\Praktek 6> go run "c:\Praktek 6\tp2\tp2.go"3
Masukan angka : 3
Rabu
PS C:\Praktek 6> go run "c:\Praktek 6\tp2\tp2.go"5
Jumat
PS C:\Praktek 6> go run "c:\Praktek 6\tp2\tp2.go"7
Minggu

IF-12-07
103112430267
Raden Aurel Aditya Kusumawaningyun

Deskripsi program

Pada kode di atas, program ini ditulis menggunakan bahasa pemrograman Go dan bertujuan untuk mengkonversi angka yang dimasukkan oleh pengguna menjadi nama hari dalam seminggu. Program dimulai dengan mendeklarasikan paket *main* dan mengimpor paket *fmt*, yang berfungsi untuk melakukan operasi input dan output.

Di dalam fungsi *main*, variabel *angka* yang bertipe integer dideklarasikan untuk menyimpan angka yang dimasukkan oleh pengguna. Program kemudian menampilkan pesan "Masukan angka : " menggunakan *fmt.Println* untuk meminta input dari pengguna. Setelah itu, fungsi *fmt.Scan(&angka)* digunakan untuk membaca input angka yang diberikan oleh pengguna dan menyimpannya dalam variabel *angka*.

Selanjutnya, program menggunakan struktur *switch* yang memeriksa nilai dari variabel *angka*. Switch ini memeriksa nilai yang dimasukkan oleh pengguna dan membandingkannya dengan beberapa kasus (*case*) yang ada. Setiap *case* mewakili angka yang sesuai dengan hari dalam seminggu: jika angka yang dimasukkan adalah 1, maka program akan mencetak "Senin"; jika 2, maka program mencetak "Selasa"; jika 3, maka program mencetak "Rabu"; dan seterusnya hingga angka 7, yang akan mencetak "Minggu". Setiap *case* berfungsi untuk mencocokkan angka yang dimasukkan dengan hari yang sesuai.

Namun, jika pengguna memasukkan angka selain 1 hingga 7, tidak ada tindakan yang diambil karena tidak ada *case* lain atau *default* yang menangani kasus tersebut. Program ini memberikan hasil sesuai dengan input angka yang valid antara 1 hingga 7, yang merupakan

representasi hari-hari dalam seminggu. Program ini sederhana namun efektif dalam memperkenalkan penggunaan struktur *switch-case* di dalam bahasa Go.

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("Masukan Kode")

    fmt.Scan(&kode)

    switch {

    case kode == "G":

        fmt.Println("Semua usia")

    case kode == "PG":

        fmt.Println("Anak-anak di atas 7 tahun")

    case kode == "PG13":

        fmt.Println("Anak-anak di atas 13 tahun")

    case kode == "R":

        fmt.Println("Dewasa")

    default:

        fmt.Println("Kode tidak valid")

    }

}

```

Screenshoot program

The screenshot shows a Go program in VS Code. The code is as follows:

```

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

```

The terminal output shows the program being run multiple times with the following inputs and outputs:

```

G
Semua usia
PS C:\Praktek 6> go run "c:\Praktek 6\tp3\tp3.go"
Masukan Kode
PG
Anak-anak di atas 7 tahun
PS C:\Praktek 6> go run "c:\Praktek 6\tp3\tp3.go"
Masukan Kode
R
Dewasa
PS C:\Praktek 6> go run "c:\Praktek 6\tp3\tp3.go"
Masukan Kode
0
Kode tidak valid
PS C:\Praktek 6>

```

A floating window titled "IF-12-" is also visible, containing the text:

```

IF-12-07
103112430267
Raden Aurel Aditya Kusumawaningyun

```

The status bar at the bottom indicates the current file is "Ln 23, Col 2", the tab size is 4, the encoding is UTF-8, and the line ending is CRLF. The zoom level is 123.1% and the theme is Prettier.

Deskripsi program

Pada kode di atas, program yang ditulis menggunakan bahasa pemrograman Go ini bertujuan untuk menentukan kategori usia berdasarkan kode yang dimasukkan oleh pengguna. Program ini menggunakan struktur *switch-case* untuk memeriksa kode yang diberikan dan memberikan hasil yang sesuai dengan kategori usia yang relevan.

Program dimulai dengan mendeklarasikan paket *main* dan mengimpor paket *fmt*, yang digunakan untuk melakukan operasi input dan output. Di dalam fungsi *main*, variabel *kode* bertipe string dideklarasikan untuk menyimpan input kode yang dimasukkan oleh pengguna.

Selanjutnya, program menampilkan pesan "Masukan Kode" menggunakan *fmt.Println* untuk meminta pengguna memasukkan sebuah kode. Fungsi *fmt.Scan(&kode)* digunakan untuk membaca input kode yang dimasukkan oleh pengguna dan menyimpannya dalam variabel *kode*.

Setelah itu, struktur *switch* digunakan untuk memeriksa nilai yang ada pada variabel *kode*. *Switch* ini tidak memerlukan ekspresi kondisi, sehingga setiap *case* yang ada berisi pengecekan apakah nilai *kode* sesuai dengan string tertentu. Pada *case* pertama, jika kode yang dimasukkan adalah "G", program akan mencetak "Semua usia". Pada *case* kedua, jika kode "PG" dimasukkan, program mencetak "Anak-anak di atas 7 tahun". Untuk kode "PG13", program akan mencetak "Anak-anak di atas 13 tahun". Jika kode yang dimasukkan adalah "R", program mencetak "Dewasa".

Jika pengguna memasukkan kode yang tidak sesuai dengan salah satu nilai yang ada dalam *case* tersebut, maka program akan mengeksekusi *default* dan mencetak "Kode tidak valid". Dengan demikian, program ini memberikan hasil yang sesuai berdasarkan kode yang dimasukkan, dan menggunakan *switch-case* untuk memeriksa dan memberikan respon yang tepat sesuai dengan kode yang diberikan oleh pengguna. Program ini berguna untuk mengklasifikasikan usia berdasarkan kategori film atau acara tertentu, seperti yang terlihat pada kode-kode seperti "G", "PG", "PG13", dan "R".

LATIHAN MODUL

1. Tugas 1

Pseudo code

Program Menghitung pH

var kادارPH: float64

input (&kادارPH)

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

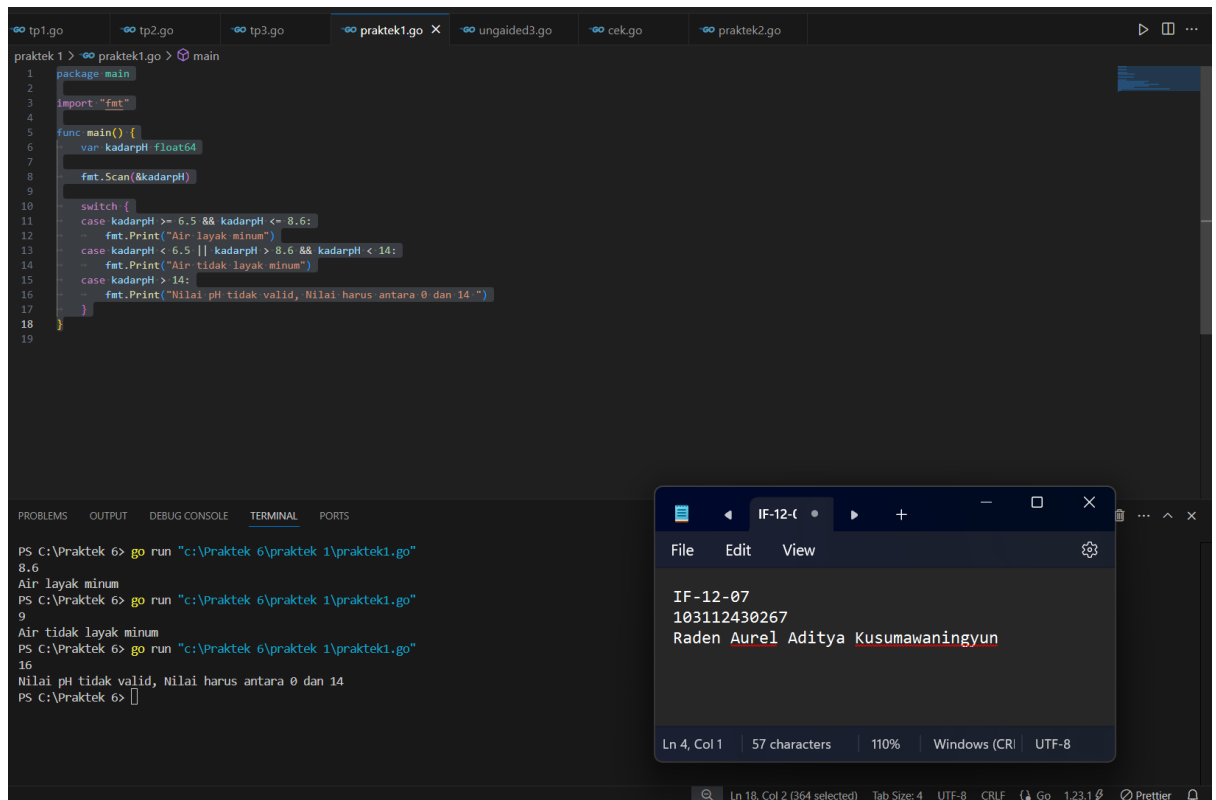
import "fmt"

func main() {
    var karpH float64

    fmt.Scan(&karpH)

    switch {
    case karpH >= 6.5 && karpH <= 8.6:
        fmt.Print("Air layak minum")
    case karpH < 6.5 || karpH > 8.6 && karpH < 14:
        fmt.Print("Air tidak layak minum")
    case karpH > 14:
        fmt.Print("Nilai pH tidak valid, Nilai harus antara 0
dan 14 ")
    }
}
```

Screenshoot program



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

```
PS C:\Praktek 6> go run "c:\Praktek 6\praktek 1\praktek1.go"
8.6
Air layak minum
PS C:\Praktek 6> go run "c:\Praktek 6\praktek 1\praktek1.go"
9
Air tidak layak minum
PS C:\Praktek 6> go run "c:\Praktek 6\praktek 1\praktek1.go"
16
Nilai pH tidak valid, Nilai harus antara 0 dan 14
PS C:\Praktek 6>
```

IF-12-07
103112430267
Raden Aurel Aditya Kusumawaningyun

Deskripsi program

Kode yang diberikan adalah program Go yang digunakan untuk menentukan apakah air layak diminum berdasarkan nilai pH yang dimasukkan oleh pengguna. Program ini dimulai dengan mendeklarasikan package **main**, yang menandakan bahwa ini adalah program utama yang dapat dijalankan. Pada bagian pertama, kita mengimpor paket **fmt** yang menyediakan fungsi untuk input dan output dalam Go.

Dalam fungsi **main()**, pertama-tama dideklarasikan variabel **kadarPH** dengan tipe data **float64** untuk menyimpan nilai pH air. Kemudian, fungsi **fmt.Scan(&kadarPH)** digunakan untuk membaca input dari pengguna dan menyimpannya ke dalam variabel **kadarPH**. Fungsi **fmt.Scan()** digunakan untuk mendapatkan input dari pengguna melalui terminal atau konsol.

Selanjutnya, program menggunakan pernyataan **switch** untuk menentukan kategori air berdasarkan nilai pH yang dimasukkan. Pernyataan **switch** ini tidak memiliki kondisi tertentu di dalamnya, artinya program akan mengevaluasi ekspresi kondisi dalam setiap **case**. Berikut adalah detail masing-masing **case** yang ada:

1. **Case pertama ($kadarPH \geq 6.5 \ \&\& \ kadarPH \leq 8.6$):**
 - o Jika nilai pH berada di antara 6.5 hingga 8.6, maka program akan mencetak "Air layak minum", karena pH air dalam rentang ini dianggap aman untuk dikonsumsi.

2. **Case kedua** (`kadarPH < 6.5 || kadarPH > 8.6 && kadarPH < 14`):
 - Jika nilai pH lebih kecil dari 6.5 atau lebih besar dari 8.6 namun kurang dari 14, maka program akan mencetak "Air tidak layak minum". Nilai pH di luar rentang 6.5 hingga 8.6 dapat menunjukkan air yang terlalu asam atau terlalu basa, yang tidak aman untuk diminum.
3. **Case ketiga** (`kadarPH > 14`):
 - Jika nilai pH lebih besar dari 14, maka program akan mencetak "Nilai pH tidak valid, Nilai harus antara 0 dan 14". Ini menunjukkan bahwa pH air yang lebih besar dari 14 tidak valid, karena skala pH hanya berlaku antara 0 dan 14.

Secara keseluruhan, program ini menggunakan logika percabangan untuk mengevaluasi apakah air tersebut layak diminum berdasarkan nilai pH, dan memberikan pesan yang sesuai berdasarkan input yang diberikan oleh pengguna.

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 (mobil)

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
    truk := 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 == "mobil" && durasi >= 2:
        fmt.Println(mobil * durasi)
    case kendaraan == "truk" && durasi <= 1:
        fmt.Println(truk)
    case kendaraan == "truk" && durasi >= 1:
        fmt.Println(truk * durasi)
    }
}
```

$\left. \begin{array}{l} \text{ } \end{array} \right\}$

Screenshot program

```
praktek2.go > go run .
4
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     truk := 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 == "truk" && durasi <= 1:
25         fmt.Println(truk)
26     case kendaraan == "truk" && durasi >= 2:
27         fmt.Println(truk * durasi)
28     }
29 }
30
31
32
Jenis Kendaraan & Durasi Parkir
motor 3 jam
6000
PS C:\Praktek 6> go run "C:\Praktek 6\praktek2\praktek2.go"
Jenis Kendaraan & Durasi Parkir
mobil 1 jam
5000
PS C:\Praktek 6> go run "C:\Praktek 6\praktek2\praktek2.go"
Jenis Kendaraan & Durasi Parkir
truk 5 jam
40000
PS C:\Praktek 6>
```

Deskripsi program

Kode yang diberikan adalah program dalam bahasa Go yang digunakan untuk menghitung biaya parkir berdasarkan jenis kendaraan dan durasi parkir. Program ini dimulai dengan mendeklarasikan package **main**, yang berarti program ini adalah program utama yang dapat dijalankan. Paket **fmt** diimpor untuk menyediakan fungsi input dan output seperti **fmt.Println** dan **fmt.Scan**, yang digunakan untuk menampilkan pesan ke layar dan membaca input dari pengguna.

Dalam fungsi **main()**, terdapat dua variabel yang dideklarasikan: **kendaraan** dengan tipe data **string** untuk menyimpan jenis kendaraan (seperti motor, mobil, atau truk), dan **durasi** dengan tipe data **int** untuk menyimpan durasi parkir dalam satuan jam. Program menampilkan pesan "Jenis Kendaraan & Durasi Parkir" untuk memberi petunjuk kepada pengguna mengenai input yang diperlukan, kemudian menggunakan **fmt.Scan(&kendaraan, &durasi)** untuk membaca jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna.

Setelah input diterima, program mendefinisikan tiga variabel: **motor**, **mobil**, dan **truk** yang masing-masing berisi tarif parkir dasar untuk setiap jenis kendaraan, yaitu 2000 untuk motor, 5000 untuk mobil, dan 8000 untuk truk. Kemudian, program menggunakan struktur kontrol **switch** untuk mengevaluasi kombinasi antara jenis kendaraan dan durasi parkir, lalu menghitung biaya parkir yang sesuai.

Pada setiap **case** di dalam **switch**, kondisi yang dievaluasi adalah kombinasi dari jenis kendaraan dan durasi parkir. Berikut adalah penjelasan dari masing-masing **case**:

1. **Case pertama** (**kendaraan == "motor" && durasi <= 1**):
 - Jika jenis kendaraan adalah "motor" dan durasi parkir 1 jam atau kurang, maka program mencetak biaya parkir sebesar 2000 (biaya untuk motor parkir selama 1 jam).
2. **Case kedua** (**kendaraan == "motor" && durasi >= 2**):
 - Jika kendaraan adalah "motor" dan durasi parkir lebih dari 1 jam, maka biaya parkir dihitung dengan mengalikan tarif dasar motor (2000) dengan durasi parkir yang dimasukkan.
3. **Case ketiga** (**kendaraan == "mobil" && durasi <= 1**):
 - Jika kendaraan adalah "mobil" dan durasi parkir 1 jam atau kurang, maka program mencetak biaya parkir sebesar 5000 (biaya untuk mobil parkir selama 1 jam).
4. **Case keempat** (**kendaraan == "mobil" && durasi >= 2**):
 - Jika kendaraan adalah "mobil" dan durasi parkir lebih dari 1 jam, maka biaya parkir dihitung dengan mengalikan tarif dasar mobil (5000) dengan durasi parkir yang dimasukkan.
5. **Case kelima** (**kendaraan == "truk" && durasi <= 1**):
 - Jika kendaraan adalah "truk" dan durasi parkir 1 jam atau kurang, maka program mencetak biaya parkir sebesar 8000 (biaya untuk truk parkir selama 1 jam).
6. **Case keenam** (**kendaraan == "truk" && durasi >= 1**):
 - Jika kendaraan adalah "truk" dan durasi parkir lebih dari 1 jam, maka biaya parkir dihitung dengan mengalikan tarif dasar truk (8000) dengan durasi parkir yang dimasukkan.

Secara keseluruhan, program ini berfungsi untuk menghitung biaya parkir sesuai dengan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Dengan menggunakan struktur **switch**, program memeriksa berbagai kemungkinan jenis kendaraan dan durasi parkir, lalu mencetak biaya parkir yang sesuai.

3. Tugas 4

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%5 == 0 && 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("Masukan 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
dengan 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\nHasil 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

The screenshot shows a Go program in VS Code. The source code in `praktek3.go` uses a `switch` statement to categorize numbers based on their remainder when divided by 5 and 10, and performs mathematical operations like addition, multiplication, squaring, and division. The terminal shows the program being executed with inputs 5, 8, 25, and 20, producing the corresponding category and result for each.

```

praktek3 > go run "c:\Praktek 6\praktek 3\praktek3.go"
1 package main
2
3 import "fmt"
4
5 func main() {
6     var input int
7
8     fmt.Print("Masukan Angka : ")
9     fmt.Scan(&input)
10
11     switch {
12
13     case input%2 != 0 && input < 25:
14         hasil := input + (input + 1)
15         fmt.Printf("Kategori: bilangan ganjil\nHasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", input, input+1, hasil)
16     case input%2 == 0 && input%5 != 0 && input%10 != 0:
17         hasil := input * (input + 1)
18         fmt.Printf("Kategori: bilangan genap\nHasil perkalian dengan bilangan berikutnya %d * %d = %d\n", input, input+1, hasil)
19     case input%5 == 0 && input%10 != 0:
20         Hasil := (input * input)
21         fmt.Printf("Hasil kuadrat dari 5\nHasil kuadrat dari %d ^2 == %d", input, Hasil)
22     case input%10 == 0:
23         hasil := input / 10
24         fmt.Printf("Kategori bilangan kelipatan 10 \nHasil pembagian antara %d / %d = %d", input, input/10, hasil)
25     }
26 }
27
PS C:\Praktek 6> go run "c:\Praktek 6\praktek 3\praktek3.go"
Masukan Angka : 5
Kategori: bilangan ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS C:\Praktek 6> go run "c:\Praktek 6\praktek 3\praktek3.go"
Masukan Angka : 8
Kategori: bilangan genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Praktek 6> go run "c:\Praktek 6\praktek 3\praktek3.go"
Masukan Angka : 25
Hasil kuadrat dari 5
Hasil kuadrat dari 25 ^2 == 625
PS C:\Praktek 6> go run "c:\Praktek 6\praktek 3\praktek3.go"
Masukan Angka : 20
Kategori bilangan kelipatan 10
Hasil pembagian antara 20 / 2 = 2
PS C:\Praktek 6>

```

Deskripsi program

Kode yang diberikan adalah program dalam bahasa Go yang digunakan untuk mengkategorikan angka yang dimasukkan oleh pengguna dan melakukan operasi matematika berdasarkan kategori angka tersebut. Program ini dimulai dengan mendeklarasikan package `main`, yang menandakan bahwa ini adalah program utama yang dapat dijalankan. Program juga mengimpor paket `fmt`, yang digunakan untuk melakukan input dan output, seperti `fmt.Print`, `fmt.Scan`, dan `fmt.Printf`.

Pada bagian awal fungsi `main()`, sebuah variabel `input` bertipe `int` dideklarasikan untuk menyimpan angka yang dimasukkan oleh pengguna. Program kemudian mencetak pesan "Masukan Angka: " untuk meminta input dari pengguna dan menggunakan `fmt.Scan(&input)` untuk membaca angka yang dimasukkan.

Setelah input diterima, program menggunakan struktur kontrol `switch` untuk mengevaluasi kondisi yang berbeda berdasarkan nilai `input`. Berikut adalah penjelasan masing-masing `case` dalam `switch`:

1. **Case pertama** (`input%2 != 0 && input < 25`):
 - Jika angka yang dimasukkan adalah bilangan ganjil (terdeteksi dengan `input%2 != 0`) dan lebih kecil dari 25, maka program akan menghitung hasil penjumlahan angka dengan angka berikutnya. Hasil penjumlahan tersebut dihitung dengan menambahkan angka dengan angka berikutnya (misalnya, jika `input` adalah 3, maka hasilnya adalah $3 + 4$). Program kemudian mencetak kategori "bilangan ganjil" dan hasil penjumlahan tersebut.
2. **Case kedua** (`input%2 == 0 && input%5 != 0 && input%10 != 0`):
 - Jika angka yang dimasukkan adalah bilangan genap (terdeteksi dengan `input%2 == 0`), tetapi bukan kelipatan 5 atau kelipatan 10 (dengan pengecekan `input%5 != 0` dan `input%10 != 0`), maka program akan menghitung hasil perkalian angka dengan angka berikutnya. Program kemudian mencetak kategori "bilangan genap" dan hasil perkalian tersebut.
3. **Case ketiga** (`input%5 == 0 && input%10 != 0`):
 - Jika angka yang dimasukkan adalah kelipatan 5, tetapi bukan kelipatan 10 (terdeteksi dengan `input%5 == 0` dan `input%10 != 0`), maka program akan menghitung hasil kuadrat dari angka tersebut (misalnya, jika `input` adalah 5, hasilnya adalah 5^2). Program kemudian mencetak hasil kuadrat tersebut bersama dengan pesan "Hasil kuadrat dari 5".
4. **Case keempat** (`input%10 == 0`):
 - Jika angka yang dimasukkan adalah kelipatan 10 (terdeteksi dengan `input%10 == 0`), maka program akan membagi angka dengan 10 dan mencetak hasil pembagian tersebut. Misalnya, jika `input` adalah 20, hasil pembagiannya adalah $20 / 10 = 2$.

Setelah program memeriksa kondisi-kondisi tersebut dengan `switch`, hasil dari operasi matematika yang sesuai akan ditampilkan oleh program dengan menggunakan `fmt.Printf`, yang memungkinkan untuk mencetak variabel dalam format tertentu.

Secara keseluruhan, program ini mengklasifikasikan angka yang dimasukkan oleh pengguna ke dalam beberapa kategori berdasarkan kondisi yang ditentukan, dan kemudian melakukan operasi matematika yang relevan sesuai dengan kategori tersebut. Program ini menggunakan logika percabangan dan operator modulus untuk memeriksa sifat angka dan memberikan hasil yang sesuai.