

LAPORAN PRAKTIKUM Algoritma Pemrograman

MODUL 11 Switch Case



Disusun oleh:

Jeky Sebayang

103112430160

S1IF-12-07

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

DASAR TEORI

Secara umum, switch-case merupakan alternatif yang lebih terstruktur untuk menggantikan penggunaan rangkaian else-if. Perbedaannya terletak pada cara pemilihan aksi, di mana pada switch-case, pemilihan dilakukan berdasarkan nilai atau hasil dari suatu ekspresi (yang bukan berupa kondisi boolean), bukan berdasarkan pengecekan kondisi benar atau salah.

Struktur dasar switch-case terdiri dari beberapa komponen :

1. **Ekspresi:** Merupakan operasi atau nilai yang dihitung sebelum dilakukan perbandingan. Ekspresi ini dapat menghasilkan nilai dari berbagai tipe data, seperti angka, string, atau tipe lainnya.
2. **Case:** Setiap case berisi nilai yang akan dibandingkan dengan hasil ekspresi. Jika nilai dalam ekspresi cocok dengan salah satu case, aksi yang sesuai pada case tersebut akan dijalankan.
3. **Aksi:** Merupakan serangkaian perintah yang akan dieksekusi apabila nilai ekspresi cocok dengan salah satu case. Jika tidak ada case yang sesuai, maka aksi pada default akan dieksekusi (jika default disediakan).

Selain itu, switch-case juga bisa dianggap sebagai pengganti struktur kontrol if-then, di mana setiap case pada switch mewakili kondisi yang diuji. Dalam hal ini, switch-case membantu menyederhanakan penulisan kode yang melibatkan banyak pilihan dan meningkatkan keterbacaan program, terutama ketika ada banyak kondisi yang harus diuji.

Keunggulan:

- Switch-case lebih rapi dan lebih mudah dikelola dibanding rangkaian if-else ketika ada banyak kondisi yang diuji.
- Dapat mempermudah pengujian kondisi yang berhubungan dengan nilai-nilai tertentu, seperti angka atau string, dibandingkan dengan hanya mengandalkan kondisi boolean dalam if-else.

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")

endprogram

Source code

```
package main

import "fmt"

func main() {

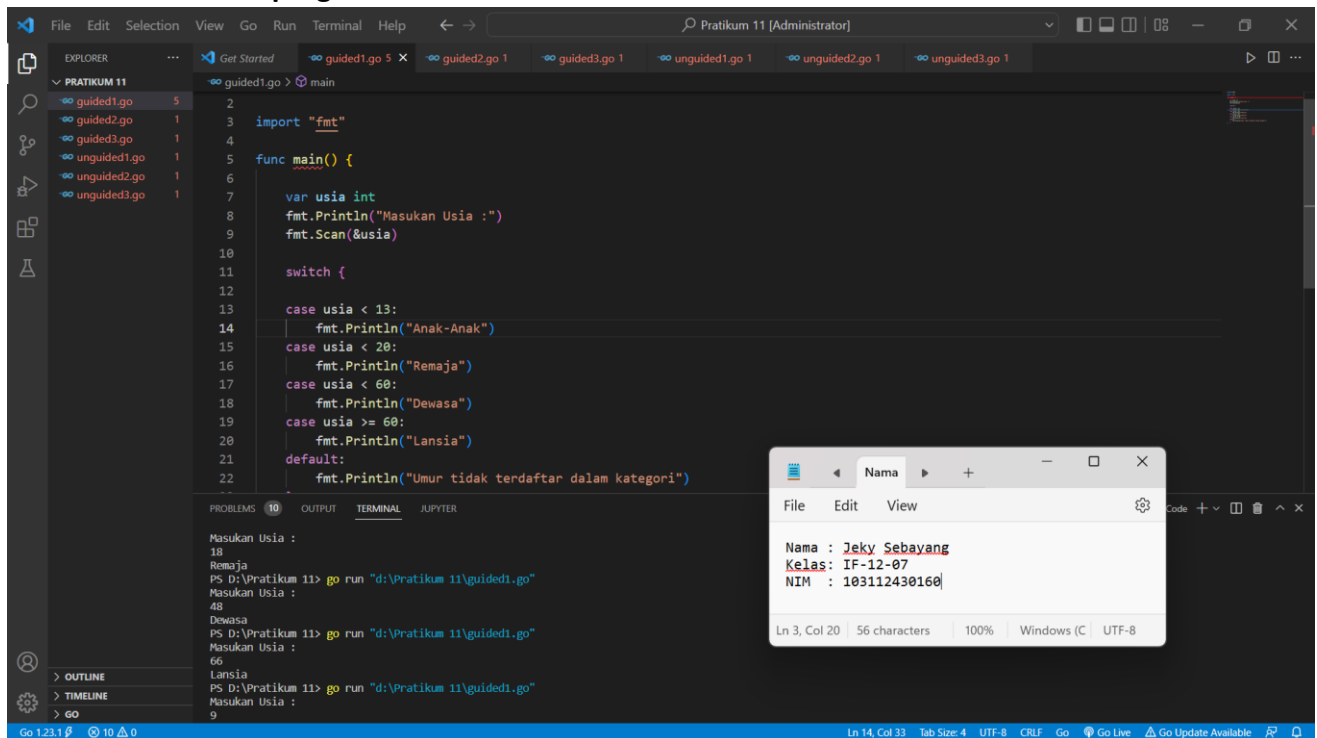
    var usia int
    fmt.Println("Masukan Usia :")
    fmt.Scan(&usia)

    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



Deskripsi program

Ini adalah contoh sederhana dari penggunaan switch-case di Go untuk mengkategorikan usia berdasarkan input pengguna.

Program di atas menggunakan bahasa pemrograman Go untuk mengklasifikasikan usia seseorang berdasarkan input yang diberikan. Dimulai dengan mendeklarasikan paket main dan mengimpor paket fmt untuk menangani input dan output.

Dalam fungsi main, variabel usia bertipe integer digunakan untuk menyimpan usia yang dimasukkan pengguna. Program kemudian meminta pengguna untuk memasukkan usia dengan menampilkan pesan "Masukan Usia :", dan `fmt.Scan(&usia)` digunakan untuk membaca input dan menyimpannya di variabel usia.

Selanjutnya, struktur switch digunakan untuk memeriksa nilai usia. Setiap kondisi case memeriksa rentang usia yang berbeda:

- Jika usia kurang dari 13, mencetak "Anak-Anak".
- Jika usia antara 13 dan 19, mencetak "Remaja".
- Jika usia antara 20 dan 59, mencetak "Dewasa".
- Jika usia 60 atau lebih, mencetak "Lansia".

Jika tidak ada yang cocok (meskipun jarang terjadi karena rentang usia sudah lengkap), bagian default akan dijalankan. Namun, di sini default jarang digunakan karena sudah ada pengaturan yang jelas untuk setiap rentang usia.

2. Tugas 2

Pseudo code

Program Hari

Kamus:

var angka :int

Algoritma:

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")

endprogram

Source code

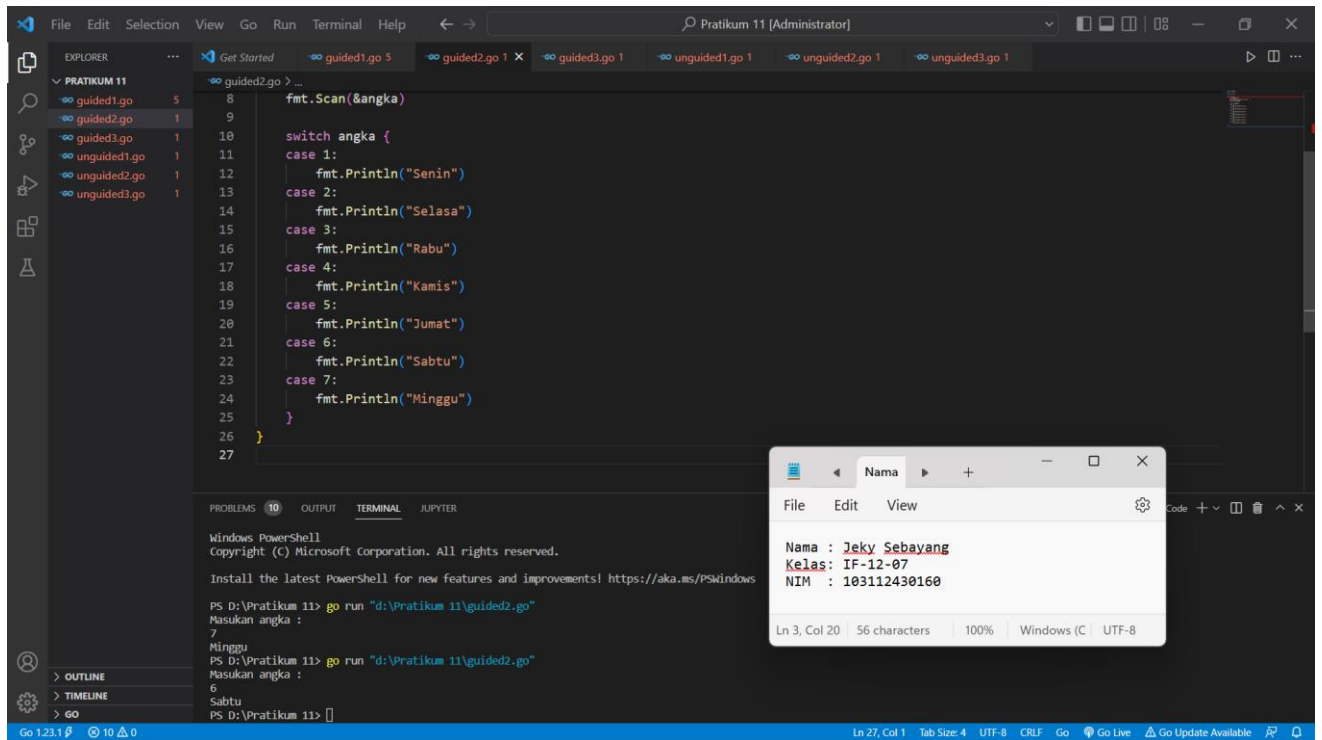
```
package main

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



```
File Edit Selection View Go Run Terminal Help
Pratikum 11 [Administrator]

EXPLORER
PRATIKUM 11
  guided1.go 5
  guided2.go 1
  guided3.go 1
  unguided1.go 1
  unguided2.go 1
  unguided3.go 1

guided2.go
8  fmt.Scan(&angka)
9
10 switch angka {
11 case 1:
12     fmt.Println("Senin")
13 case 2:
14     fmt.Println("Selasa")
15 case 3:
16     fmt.Println("Rabu")
17 case 4:
18     fmt.Println("Kamis")
19 case 5:
20     fmt.Println("Jumat")
21 case 6:
22     fmt.Println("Sabtu")
23 case 7:
24     fmt.Println("Minggu")
25 }
26
27

PROBLEMS 10 OUTPUT TERMINAL JUPYTER
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Pratikum 11> go run "d:\Pratikum 11\guided2.go"
Masukan angka :
7
Minggu
PS D:\Pratikum 11> go run "d:\Pratikum 11\guided2.go"
Masukan angka :
6
Sabtu
PS D:\Pratikum 11>

Nama
File Edit View
Nama : Jeky Sebayang
Kelas : IF-12-07
NIM : 103112430160
Ln 3, Col 20 | 56 characters | 100% | Windows (C) UTF-8
```

Deskripsi program

Program ini ditulis menggunakan bahasa pemrograman Go untuk mengonversi angka yang dimasukkan pengguna menjadi nama hari dalam seminggu. Program dimulai dengan mendeklarasikan paket main dan mengimpor paket fmt untuk menangani input dan output. Di dalam fungsi main, variabel angka bertipe integer digunakan untuk menyimpan angka yang dimasukkan. Program meminta pengguna untuk memasukkan angka dengan menampilkan pesan "Masukan angka :". Setelah itu, `fmt.Scan(&angka)` digunakan untuk membaca input dan menyimpannya dalam variabel angka.

Program kemudian menggunakan struktur switch untuk memeriksa nilai dari variabel angka. Setiap case dalam switch mewakili angka yang sesuai dengan hari dalam seminggu: angka 1 akan mencetak "Senin", angka 2 akan mencetak "Selasa", angka 3 untuk "Rabu", dan seterusnya hingga angka 7 yang mencetak "Minggu".

Namun, jika angka selain 1 hingga 7 dimasukkan, program tidak akan memberikan respons, karena tidak ada penanganan untuk kasus tersebut.

3. Tugas 3

Pseudo code

Program Kode**Kamus:**

Var kode: string

Algoritma:

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")

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:

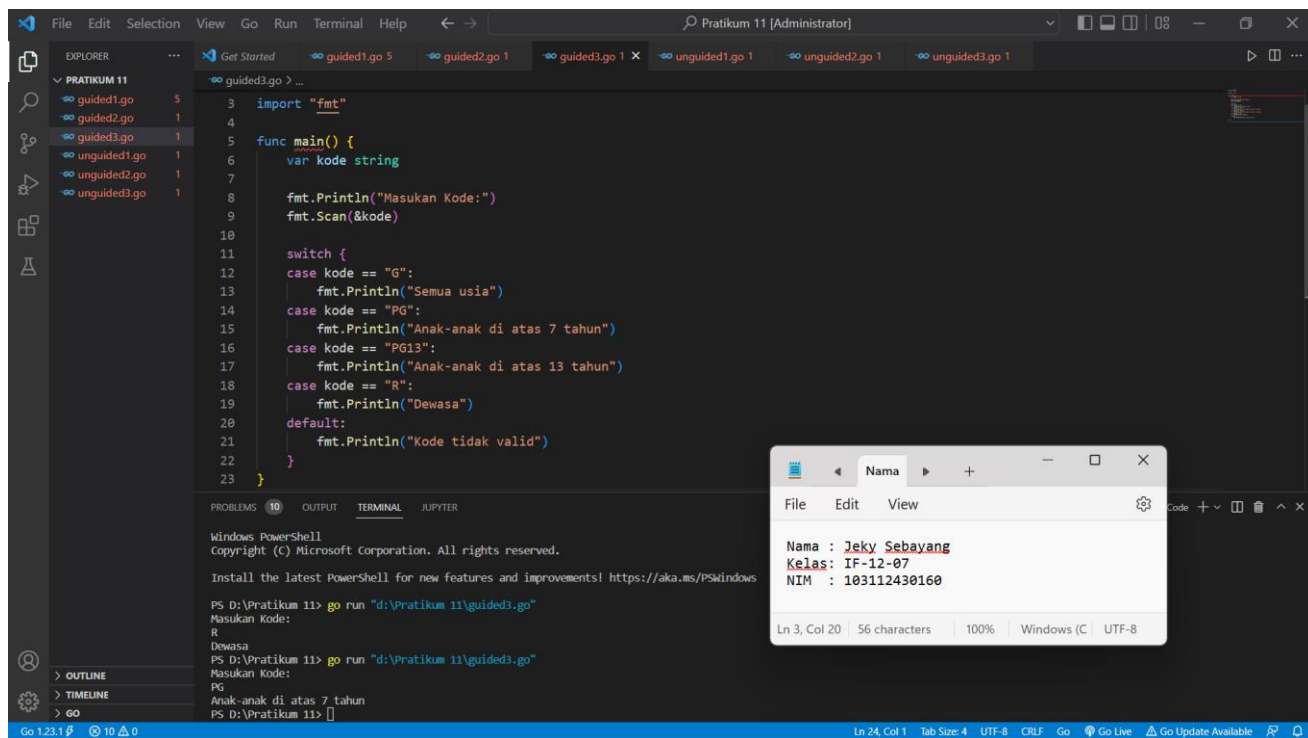
    fmt.Println("Kode tidak valid")

}

}

```

Screenshoot program



Deskripsi program

Program ini menggunakan bahasa pemrograman Go untuk mengklasifikasikan kategori usia berdasarkan kode yang dimasukkan oleh pengguna. Dengan menggunakan struktur switch-case, program memeriksa kode yang dimasukkan dan menampilkan kategori usia yang sesuai. Program dimulai dengan mengimpor paket fmt untuk menangani input dan output. Di dalam fungsi main, variabel kode bertipe string digunakan untuk menyimpan input kode dari pengguna. Program kemudian menampilkan pesan "Masukan Kode" dan menggunakan fmt.Scan(&kode) untuk membaca input.

Struktur switch digunakan untuk memeriksa kode yang dimasukkan. Jika kode "G" dimasukkan, program akan mencetak "Semua usia". Untuk kode "PG", program mencetak "Anak-anak di atas 7 tahun", dan untuk kode "PG13" mencetak "Anak-anak di atas 13 tahun". Jika kode "R" dimasukkan, program akan mencetak "Dewasa".

Jika kode yang dimasukkan tidak sesuai dengan salah satu nilai yang ada, program akan menampilkan pesan default "Kode tidak valid". Program ini berguna untuk mengkategorikan usia berdasarkan jenis film atau acara, seperti "G", "PG", "PG13", dan "R".

LATIHAN MODUL

1. Tugas 1

Pseudo code

Program Menghitung pH

Kamus:

```
var karpH: float64
```

Algoritma:

```
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 ")
```

```
endprogram
```

Source code

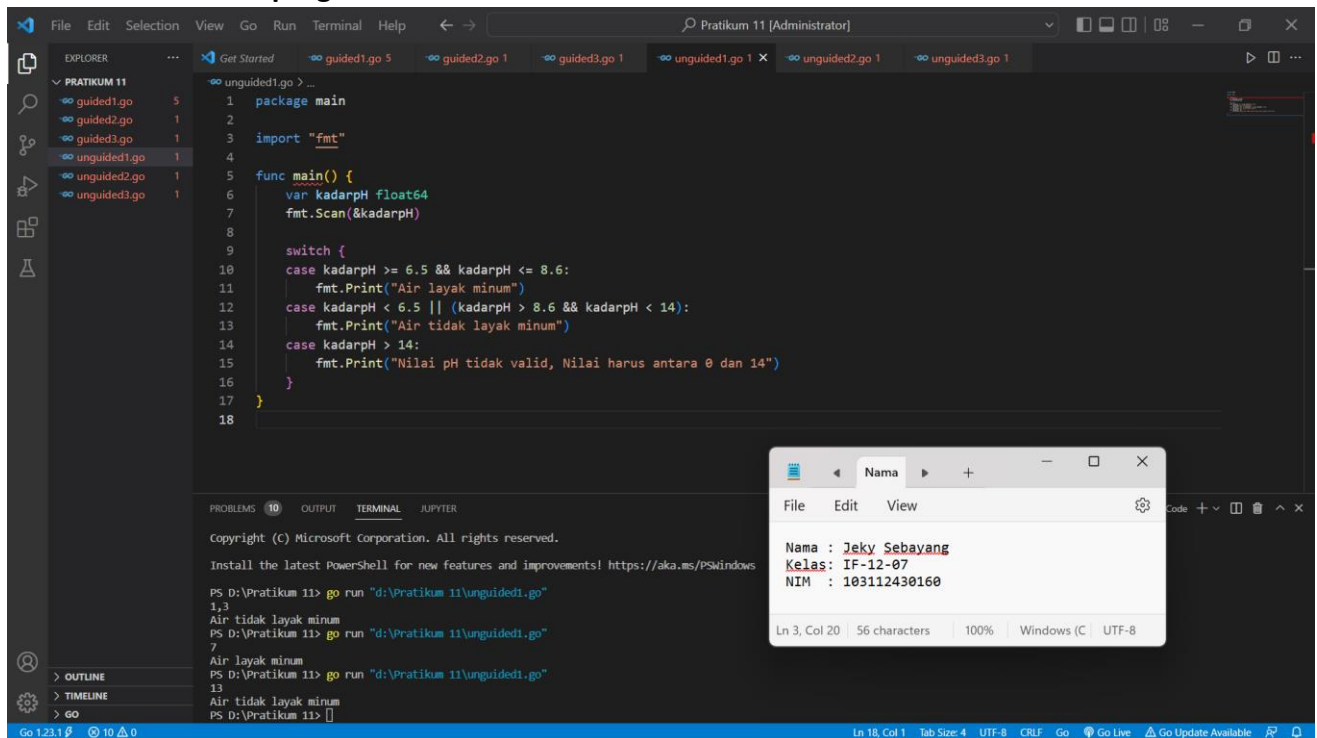
```
package main

import "fmt"

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

    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



Deskripsi program

Program Go ini digunakan untuk menentukan apakah air layak diminum berdasarkan nilai pH yang dimasukkan oleh pengguna. Program dimulai dengan mendeklarasikan package main, yang menandakan bahwa ini adalah program utama. Paket fmt diimpor untuk menangani input dan output.

Di dalam fungsi main(), variabel kadarpH dideklarasikan dengan tipe float64 untuk menyimpan nilai pH yang dimasukkan. Fungsi fmt.Scan(&kadarpH) digunakan untuk mengambil input dari pengguna.

Setelah itu, program menggunakan pernyataan switch untuk mengevaluasi nilai pH dan menentukan kategori air. Berikut adalah penjelasan untuk setiap kondisi:

- Case pertama** (kadarpH >= 6.5 && kadarpH <= 8.6):
Jika pH berada di rentang 6.5 hingga 8.6, program akan mencetak "Air layak minum", karena pH tersebut aman untuk dikonsumsi.
- Case kedua** (kadarpH < 6.5 || kadarpH > 8.6 && kadarpH < 14):
Jika pH lebih rendah dari 6.5 atau lebih tinggi dari 8.6 tetapi kurang dari 14, program akan mencetak "Air tidak layak minum", karena pH di luar rentang tersebut tidak aman untuk dikonsumsi.
- Case ketiga** (kadarpH > 14):
Jika pH lebih besar dari 14, program akan mencetak "Nilai pH tidak valid, Nilai harus antara 0 dan 14", karena skala pH hanya berlaku antara 0 dan 14.

Secara keseluruhan, program ini memeriksa nilai pH yang dimasukkan oleh pengguna dan memberikan respons yang sesuai berdasarkan kategori air yang layak atau tidak layak minum.

2. Tugas 2

Pseudo code

Program Parkir

Kamus:

var kendaraan: string

var durasi: int

Algoritma:

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)

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)
    }
}
```

Screenshoot program

The screenshot shows a Go IDE with a file explorer on the left containing files named `guided1.go`, `guided2.go`, `guided3.go`, `unguided1.go`, `unguided2.go`, and `unguided3.go`. The main editor displays the code for `unguided2.go`, which defines base parking fees for motor (2000), mobil (5000), and truk (8000), and uses a switch statement to calculate the total fee based on the vehicle type and duration. The terminal at the bottom shows the execution of the program, which prompts for vehicle type and duration, and outputs the calculated parking fee. A small dialog box titled 'Nama' is also visible, containing the text: 'Nama : Jeky Sebayang', 'Kelas: IF-12-07', and 'NIM : 103112430160'.

```
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
```

```
PS D:\Pratikum 11> go run "d:\Pratikum 11\unguided2.go"
Jenis Kendaraan & Durasi Parkir
motor 10jam
20000
PS D:\Pratikum 11> go run "d:\Pratikum 11\unguided2.go"
Jenis Kendaraan & Durasi Parkir
mobil 5 jam
25000
PS D:\Pratikum 11> go run "d:\Pratikum 11\unguided2.go"
Jenis Kendaraan & Durasi Parkir
truk 3 jam
24000
PS D:\Pratikum 11>
```

Deskripsi program

Program ini ditulis dalam bahasa Go dan bertujuan untuk menghitung biaya parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Program dimulai dengan mengimpor paket `fmt` untuk menangani input dan output. Di dalam fungsi `main()`, ada dua variabel: `kendaraan` (untuk menyimpan jenis kendaraan) dan `durasi` (untuk menyimpan durasi parkir dalam jam).

Setelah menampilkan pesan untuk meminta input, program membaca jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Kemudian, program menetapkan tarif dasar parkir untuk masing-masing jenis kendaraan: 2000 untuk motor, 5000 untuk mobil, dan 8000 untuk truk.

Dengan menggunakan struktur kontrol `switch`, program memeriksa kombinasi jenis kendaraan dan durasi parkir, lalu menghitung biaya parkir yang sesuai. Berikut adalah rincian setiap kondisi yang dievaluasi:

1. Jika kendaraan "motor" dan durasi parkir 1 jam atau kurang, biaya parkir adalah 2000.
2. Jika kendaraan "motor" dan durasi lebih dari 1 jam, biaya dihitung dengan mengalikan tarif dasar (2000) dengan durasi.
3. Jika kendaraan "mobil" dan durasi parkir 1 jam atau kurang, biaya parkir adalah 5000.
4. Jika kendaraan "mobil" dan durasi lebih dari 1 jam, biaya dihitung dengan mengalikan tarif dasar (5000) dengan durasi.
5. Jika kendaraan "truk" dan durasi parkir 1 jam atau kurang, biaya parkir adalah 8000.
6. Jika kendaraan "truk" dan durasi lebih dari 1 jam, biaya dihitung dengan mengalikan tarif dasar (8000) dengan durasi.

Secara keseluruhan, program ini memastikan bahwa biaya parkir dihitung dengan tepat berdasarkan input yang diberikan pengguna, menggunakan struktur `switch` untuk memeriksa kondisi yang relevan.

3. Tugas 3

Pseudo code

Program operasi MTK**Kamus:**

var input: int

Algoritma:

output ("Masukan Angka : ")

input (&input)

depend on:

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

hasil := input + (input + 1)

output ("Kategori: bilangan ganjil\n Hasil 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)

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:

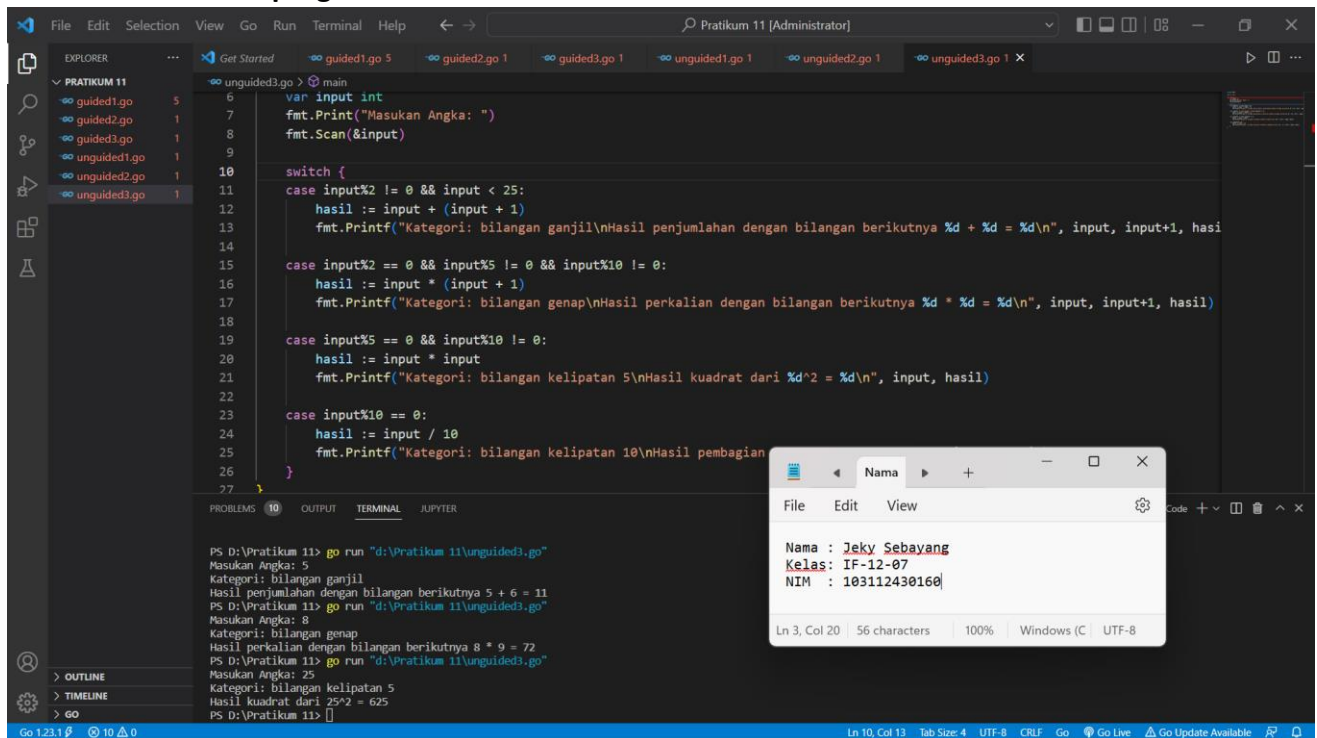
    hasil := input / 10

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

}

}
```

Screenshoot program



```
var input int
fmt.Println("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("Kategori: bilangan kelipatan 5\nHasil kuadrat dari %d^2 = %d\n", input, hasil)
case input%10 == 0:
    hasil := input / 10
    fmt.Printf("Kategori: bilangan kelipatan 10\nHasil pembagian
```

```
PS D:\Pratikum 11> go run "d:\Pratikum 11\unguided3.go"
Masukan Angka: 5
Kategori: bilangan ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS D:\Pratikum 11> go run "d:\Pratikum 11\unguided3.go"
Masukan Angka: 8
Kategori: bilangan genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS D:\Pratikum 11> go run "d:\Pratikum 11\unguided3.go"
Masukan Angka: 25
Kategori: bilangan kelipatan 5
Hasil kuadrat dari 25^2 = 625
PS D:\Pratikum 11>
```

Nama : Jeky Sebayang
Kelas : IF-12-07
NIM : 103112430160

Deskripsi program

Program ini ditulis dalam bahasa Go untuk mengklasifikasikan angka yang dimasukkan oleh pengguna dan melakukan operasi matematika sesuai dengan kategori angka tersebut. Dimulai dengan mendeklarasikan variabel input untuk menyimpan angka dari pengguna. Program kemudian meminta input melalui pesan "Masukan Angka: " dan membacanya menggunakan `fmt.Scan(&input)`.

Setelah menerima input, program menggunakan struktur switch untuk memeriksa kategori angka yang dimasukkan, berdasarkan beberapa kondisi berikut:

- Ganjil dan kurang dari 25:** Jika angka adalah ganjil dan lebih kecil dari 25, program menghitung hasil penjumlahan angka dengan angka berikutnya, kemudian menampilkan hasilnya bersama kategori "bilangan ganjil".
- Genap tetapi bukan kelipatan 5 atau 10:** Jika angka genap namun tidak kelipatan 5 atau 10, program menghitung hasil perkalian angka dengan angka berikutnya dan menampilkan hasilnya dengan kategori "bilangan genap".
- Kelipatan 5 tetapi bukan kelipatan 10:** Jika angka kelipatan 5 (bukan kelipatan 10), program menghitung hasil kuadrat angka tersebut dan menampilkan hasilnya.
- Kelipatan 10:** Jika angka kelipatan 10, program membagi angka dengan 10 dan menampilkan hasil pembagian tersebut.

Program ini menggunakan percabangan dan operator modulus untuk mengidentifikasi kategori angka dan menjalankan operasi matematika yang sesuai. Hasilnya ditampilkan menggunakan `fmt.Printf` agar output lebih terstruktur.