

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**MODUL 11**  
**SWITCH – CASE**



**Disusun oleh:**  
**Besthian Guido Rafael Simbolon**  
**103112430258**  
**Informatika – 12 – 07**

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## Dasar Teori

Go merupakan bahasa pemrograman yang dikelola oleh Google. Google tidak bekerja sendirian, melainkan bekerja sama dengan 3 orang tokoh handal pada tahun 2009. Robert Griesemer, Rob Pike dan Ken Thompson merupakan ketiga tokoh tersebut. Pada bahasa pemrograman Go (Golang), **Switch Case** adalah sebuah struktur Kode yang digunakan untuk memeriksa nilai dari suatu ekspresi dan mengeksekusi blok kode yang sesuai dengan nilai yang cocok. Struktur ini sering digunakan untuk menggantikan penggunaan banyak pernyataan if-else yang dapat membuat kode menjadi panjang dan sulit dibaca. Kode dasar dari switch diawali dengan kata kunci switch, Kemudian diikuti dengan ekspresi yang ingin dibandingkan. Setiap kondisi yang ingin diperiksa didefinisikan menggunakan kata kunci case, diikuti dengan nilai yang akan dibandingkan dengan ekspresi tersebut. Jika ekspresi cocok dengan nilai salah satu case, maka blok kode dalam case tersebut yang akan dijalankan. Jika tidak ada yang cocok, maka bagian default akan dieksekusi jika ada.

Selain itu, dalam Go, kita juga bisa menggunakan switch tanpa menuliskan ekspresi, di mana setiap case berisi kondisi yang dievaluasi sebagai ekspresi boolean. Struktur ini memungkinkan untuk menggunakan logika yang lebih kompleks dan tetap menjaga kejelasan kode. Misalnya, kita bisa memeriksa beberapa kondisi dalam satu case dengan memisahkan nilai-nilai yang ingin dibandingkan menggunakan tanda koma. Ini membuat kode lebih efisien jika kita perlu membandingkan ekspresi dengan banyak nilai sekaligus. Secara umum, penggunaan switch – case pada Golang memberikan kemudahan dalam pembacaan kode terutama ketika ada banyak kondisi yang perlu diperiksa.

# Tugas Pendahuluan

## 1. Tugas Pendahuluan 1

**Program** Umur

**Kamus**

var umur int

**Algoritma**

input (&umur)

**switch**

case kondisi\_1 : jika umur == 0  
aksi : Output ("Masukkan Umur >= 1")

case kondisi\_2 : jika umur < 13  
aksi : Output ("anak anak")

case kondisi\_3 : jika umur < 20  
aksi : Output ("remaja")

case kondisi\_4 : jika umur < 60  
aksi : Output ("dewasa")

default  
aksi : Output ("lansia")

**end switch**

**end program**

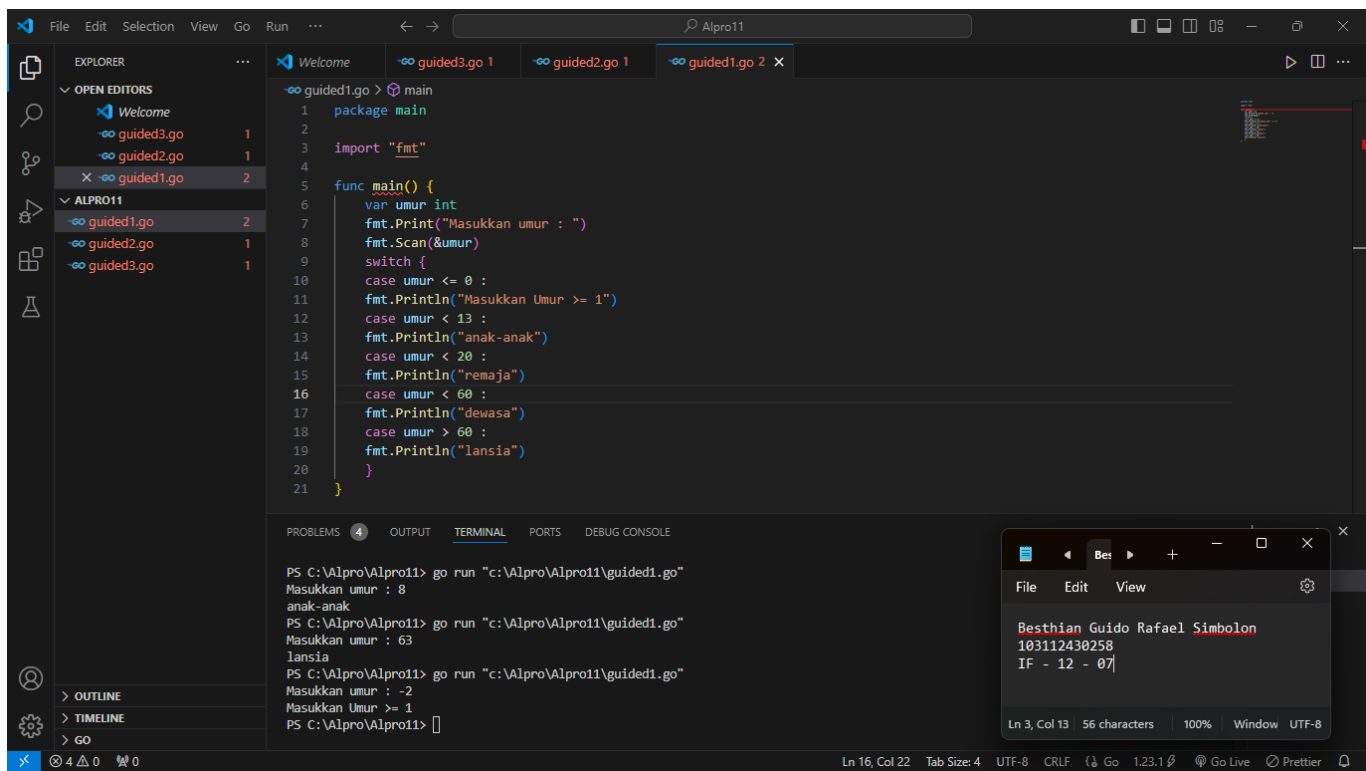
## Source Code

```
package main

import "fmt"

func main() {
    var umur int
    fmt.Print("Masukkan
umur : ")
    fmt.Scan(&umur)
    switch {
    case umur <= 0 :
        fmt.Println("Masukkan
Umur >= 1")
    case umur < 13 :
        fmt.Println("anak-
anak")
    case umur < 20 :
        fmt.Println("remaja")
    case umur < 60 :
        fmt.Println("dewasa")
    case umur > 60 :
        fmt.Println("lansia")
    }
}
```

## Screenshoot Program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var umur int
7     fmt.Print("Masukkan umur : ")
8     fmt.Scan(&umur)
9     switch {
10    case umur <= 0 :
11        fmt.Println("Masukkan Umur >= 1")
12    case umur < 13 :
13        fmt.Println("anak-anak")
14    case umur < 20 :
15        fmt.Println("remaja")
16    case umur < 60 :
17        fmt.Println("dewasa")
18    case umur > 60 :
19        fmt.Println("lansia")
20    }
21 }
```

```
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\guided1.go"
Masukkan umur : 8
anak-anak
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\guided1.go"
Masukkan umur : 63
lansia
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\guided1.go"
Masukkan umur : -2
Masukkan Umur >= 1
PS C:\Alpro\Alpro11> []
```

Best

Besthian Guido Rafael Simbolon  
103112430258  
IF - 12 - 07

## Deskripsi Program

Program meminta input dari pengguna berupa umur, lalu memeriksa kategori usia berdasarkan nilai yang dimasukkan. Pertama, program mendeklarasikan variabel umur bertipe int untuk menyimpan nilai umur yang dimasukkan oleh pengguna. Dengan menggunakan `fmt.Print`, program meminta pengguna untuk memasukkan umur, dan `fmt.Scan` digunakan untuk membaca input tersebut dan menyimpannya ke dalam variabel umur. Setelah itu, program memasuki struktur `switch`, yang memeriksa beberapa kondisi berdasarkan nilai umur. Struktur `switch` di sini tidak memiliki ekspresi setelah kata kunci `switch`, sehingga setiap `case` berfungsi sebagai kondisi yang dievaluasi secara boolean. Program pertama-tama memeriksa apakah umur kurang dari atau sama dengan 0. Jika iya, program akan menampilkan pesan "Masukkan Umur >= 1". Jika kondisi tersebut tidak terpenuhi, program akan memeriksa kondisi lainnya. Untuk umur yang kurang dari 13 tahun, program akan mencetak "anak-anak", sedangkan umur yang kurang dari 20 tahun akan mencetak "remaja". Jika umur kurang dari 60 tahun, maka kategori yang dicetak adalah "dewasa", dan jika umur lebih dari 60 tahun, program akan menampilkan "lansia". Dengan demikian, kode ini memberikan kategori usia berdasarkan input yang diberikan oleh pengguna dan memberikan respon yang sesuai dengan rentang usia yang telah ditentukan.

## 2. Tugas Pendahuluan 2

**Program Hari**

**Kamus**

var hari int

**Algoritma**

Input (&hari)

**Switch**

case kondisi 1 : hari bernilai 1

aksi : hasil := "Senin"

case kondisi 2 : hari bernilai 2

aksi : hasil := "Selasa"

case kondisi 3 : hari bernilai 3

aksi : hasil := "Rabu"

case kondisi 4 : hari bernilai 4

aksi : hasil := "Kamis"

case kondisi 5 : hari bernilai 5

aksi : hasil := "Jumat"

case kondisi 6 : hari bernilai 6

aksi : hasil := "Sabtu"

case kondisi 7 : hari bernilai 7

aksi : hasil := "Minggu"

default

aksi : Output("Masukkan Hari 1  
– 7 ")

Output ( hasil )

**end switch**

**end program**

## Source Code

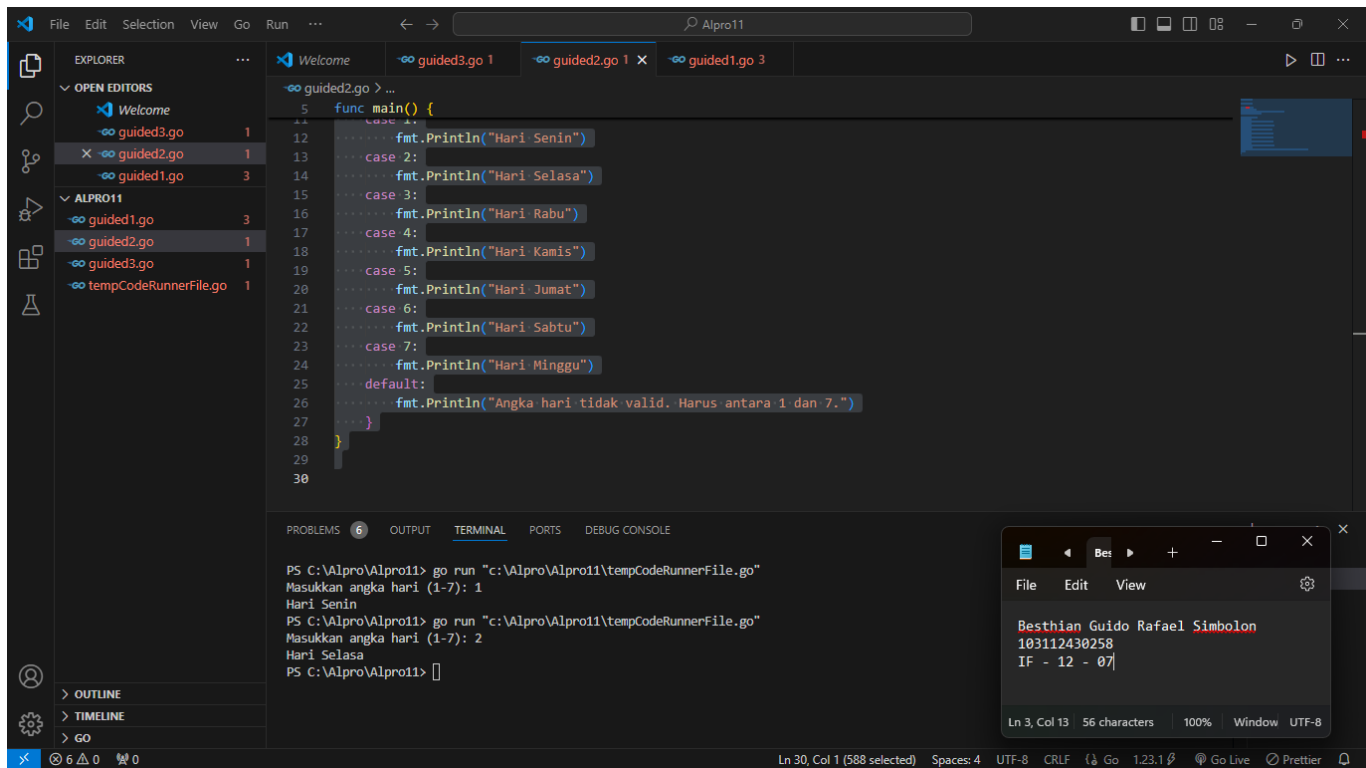
```
package main

import "fmt"

func main() {
    var hari int
    fmt.Print("Masukkan
angka hari (1-7): ")
    fmt.Scanln(&hari)

    switch hari {
    case 1:
        fmt.Println("Hari
Senin")
    case 2:
        fmt.Println("Hari
Selasa")
    case 3:
        fmt.Println("Hari
Rabu")
    case 4:
        fmt.Println("Hari
Kamis")
    case 5:
        fmt.Println("Hari
Jumat")
    case 6:
        fmt.Println("Hari
Sabtu")
    case 7:
        fmt.Println("Hari
Minggu")
    default:
        fmt.Println("Angka
hari tidak valid. Harus
antara 1 dan 7.")
    }
}
```

## Screenshoot Program



```
5 func main() {
12     fmt.Println("Hari Senin")
13     case 2:
14     fmt.Println("Hari Selasa")
15     case 3:
16     fmt.Println("Hari Rabu")
17     case 4:
18     fmt.Println("Hari Kamis")
19     case 5:
20     fmt.Println("Hari Jumat")
21     case 6:
22     fmt.Println("Hari Sabtu")
23     case 7:
24     fmt.Println("Hari Minggu")
25     default:
26     fmt.Println("Angka hari tidak valid. Harus antara 1 dan 7.")
27 }
28
29
30
```

```
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\tempCodeRunnerFile.go"
Masukkan angka hari (1-7): 1
Hari Senin
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\tempCodeRunnerFile.go"
Masukkan angka hari (1-7): 2
Hari Selasa
PS C:\Alpro\Alpro11>
```

Besthian Guido Rafael Simbolon  
103112430258  
IF - 12 - 07

## Deskripsi Program

Program di atas berjalan dimulai Ketika program meminta input dari pengguna berupa angka yang mewakili hari dalam seminggu, kemudian mencetak nama hari yang sesuai dengan angka yang dimasukkan. Program mulai dengan mendeklarasikan variabel hari bertipe `int` untuk menampung input yang dimasukkan oleh pengguna. Setelah itu, program menggunakan `fmt.Print` untuk meminta pengguna memasukkan angka antara 1 hingga 7, yang mewakili hari dalam seminggu. Dengan `fmt.Scanln`, input yang dimasukkan oleh pengguna kemudian disimpan ke dalam variabel hari. Selanjutnya, program menggunakan struktur `switch` untuk memeriksa nilai yang dimasukkan oleh pengguna. Setiap case memeriksa apakah angka yang dimasukkan cocok dengan angka yang sesuai dengan hari dalam seminggu. Misalnya, jika pengguna memasukkan angka 1, maka program akan mencetak "Hari Senin", jika angka 2, program akan mencetak "Hari Selasa", dan seterusnya hingga angka 7 yang mewakili "Hari Minggu". Jika angka yang dimasukkan tidak berada dalam rentang 1 hingga 7, maka program akan menampilkan pesan default, yaitu "Angka hari tidak valid. Harus antara 1 dan 7." Dengan demikian, kode ini memberikan respon yang sesuai berdasarkan input hari yang valid atau tidak valid.



### 3. Tugas Pendahuluan 3

**Program** KodeFilm

**Kamus**

Var Kode String

**Algoritma**

Input (&Kode)

**switch kode**

case "G" :

Output("untuk semua umur")

case "PG" :

Output("untuk anak-anak diatas  
7 tahun")

case "PG-13" :

Output("untuk remaja diatas 13  
tahun")

case "R" :

Output("untuk dewasa")

default :

Output("kode tidak valid")

**End Switch**

**End Program**

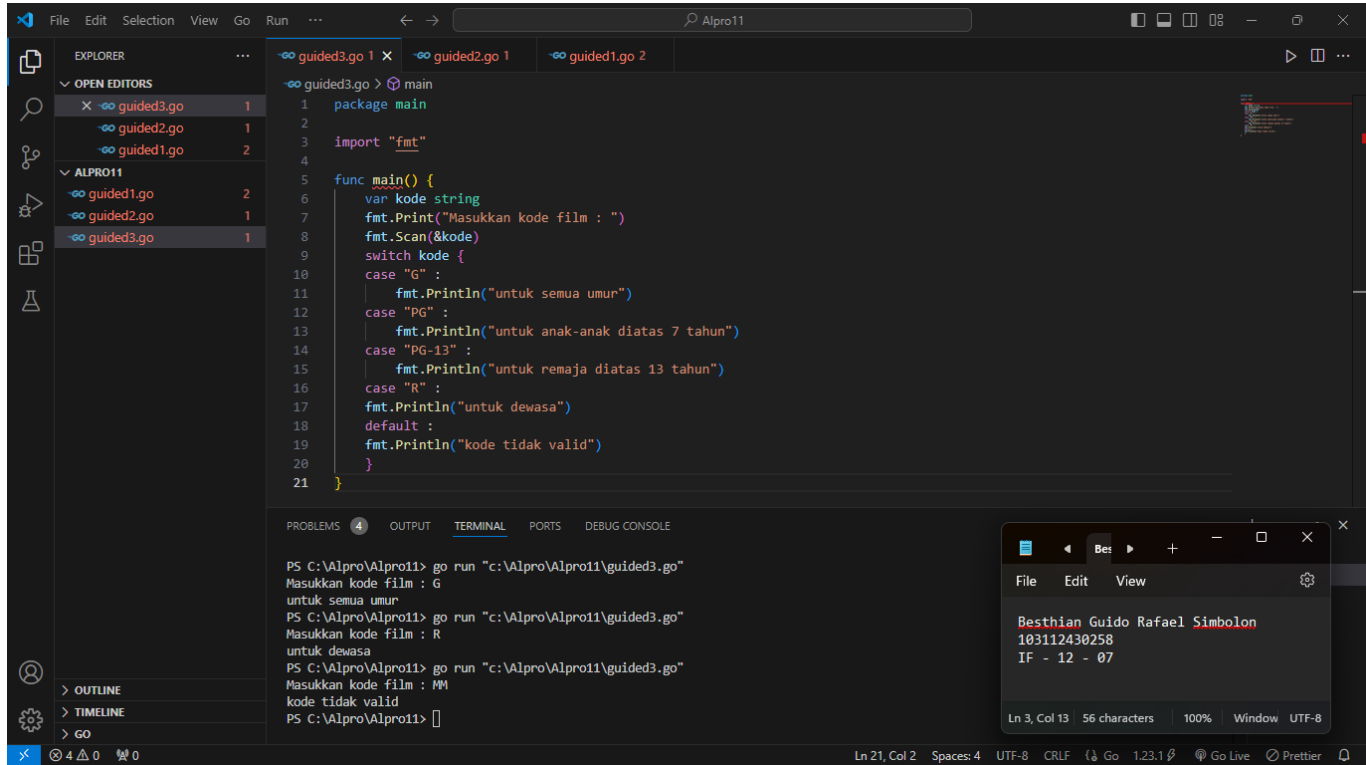
## Source Code

```
package main

import "fmt"

func main() {
    var kode string
    fmt.Print("Masukkan
kode film : ")
    fmt.Scan(&kode)
    switch kode {
    case "G" :
        fmt.Println("untuk
semua umur")
    case "PG" :
        fmt.Println("untuk
anak-anak diatas 7 tahun")
    case "PG-13" :
        fmt.Println("untuk
remaja diatas 13 tahun")
    case "R" :
        fmt.Println("untuk
dewasa")
    default :
        fmt.Println("kode
tidak valid")
    }
}
```

## Screenshoot Program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kode string
7     fmt.Print("Masukkan kode film : ")
8     fmt.Scan(&kode)
9     switch kode {
10     case "G" :
11         fmt.Println("untuk semua umur")
12     case "PG" :
13         fmt.Println("untuk anak-anak diatas 7 tahun")
14     case "PG-13" :
15         fmt.Println("untuk remaja diatas 13 tahun")
16     case "R" :
17         fmt.Println("untuk dewasa")
18     default :
19         fmt.Println("kode tidak valid")
20     }
21 }
```

```
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\guided3.go"
Masukkan kode film : G
untuk semua umur
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\guided3.go"
Masukkan kode film : R
untuk dewasa
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\guided3.go"
Masukkan kode film : PM
kode tidak valid
PS C:\Alpro\Alpro11>
```

Best

Besthian Guido Rafael Simbolon  
103112430258  
IF - 12 - 07

## Deskripsi Program

Program di atas berjalan dimulai Ketika program meminta input berupa kode film yang menunjukkan kategori usia penonton yang sesuai, lalu mencetak informasi terkait kategori tersebut. Program dimulai dengan mendeklarasikan variabel kode bertipe string untuk menampung input kode film yang dimasukkan oleh pengguna. Setelah itu, program meminta pengguna untuk memasukkan kode film menggunakan `fmt.Print`, dan input tersebut dibaca menggunakan `fmt.Scan(&kode)`. Setelah input diterima, program menggunakan struktur **switch** untuk memeriksa nilai yang dimasukkan oleh pengguna. Setiap case memeriksa apakah kode film yang dimasukkan sesuai dengan salah satu kategori yang telah ditentukan. Misalnya, jika pengguna memasukkan kode "G", program akan menampilkan pesan "untuk semua umur". Jika kode yang dimasukkan adalah "PG", maka program akan mencetak "untuk anak-anak di atas 7 tahun", dan seterusnya untuk kode "PG-13" dan "R". Jika pengguna memasukkan kode yang tidak valid atau tidak sesuai dengan yang ada di case, maka program akan menampilkan pesan "kode tidak valid" melalui blok default. Dengan demikian, kode ini digunakan untuk memverifikasi kategori usia berdasarkan kode film yang dimasukkan oleh pengguna dan memberikan respons sesuai dengan kategori tersebut.

## LATIHAN MODUL

### 1. Latihan Modul 1

**Program pH**

**Kamus**

var pH float64

**Algoritma**

**Input (&pH)**

**switch**

case 1 : jika ph >= 6.5 && ph <=8.6

aksi 1 : Output )"Air layak  
diminum")

case 2 : jika ph < 6.5 && ph > 0 ||  
ph > 8.6 && ph <= 14

aksi 2 : Output ("air tidak layak  
minum")

case 3 : jika ph > 14 && ph < 0

aksi 3 : Output ("inputan harus 0-  
14")

**end switch**

**end program**

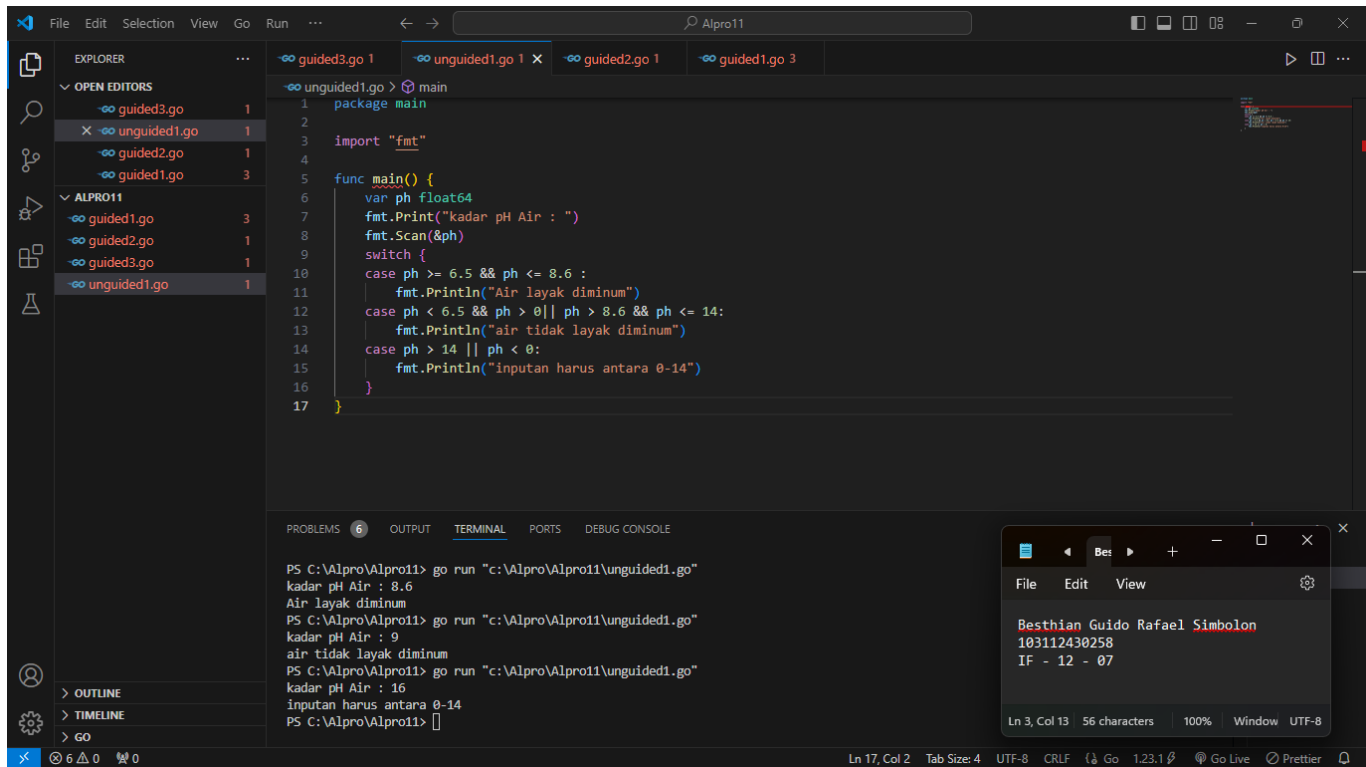
## Source Code

```
package main

import "fmt"

func main() {
    var ph float64
    fmt.Print("kadar pH Air
: ")
    fmt.Scan(&ph)
    switch {
    case ph >= 6.5 && ph <=
8.6 :
        fmt.Println("Air
layak diminum")
    case ph < 6.5 && ph >
0 || ph > 8.6 && ph <= 14:
        fmt.Println("air
tidak layak diminum")
    case ph > 14 || ph < 0:
        fmt.Println("inputan
harus antara 0-14")
    }
}
```

## Screenshoot Program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float64
7     fmt.Print("kadar pH Air : ")
8     fmt.Scan(&ph)
9     switch {
10    case ph >= 6.5 && ph <= 8.6 :
11        fmt.Println("Air layak diminum")
12    case ph < 6.5 && ph > 0 || ph > 8.6 && ph <= 14:
13        fmt.Println("air tidak layak diminum")
14    case ph > 14 || ph < 0:
15        fmt.Println("inputan harus antara 0-14")
16    }
17 }
```

PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\unguided1.go"  
kadar pH Air : 8.6  
Air layak diminum  
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\unguided1.go"  
kadar pH Air : 9  
air tidak layak diminum  
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\unguided1.go"  
kadar pH Air : 16  
inputan harus antara 0-14  
PS C:\Alpro\Alpro11>

## Deskripsi Program

Program digunakan untuk menentukan apakah air layak diminum berdasarkan kadar pH yang dimasukkan oleh pengguna. Program dimulai dengan mendeklarasikan variabel `ph` bertipe `float64` untuk menyimpan nilai pH air yang dimasukkan. Kemudian, program meminta pengguna untuk memasukkan nilai pH melalui `fmt.Print` dan membaca input dengan `fmt.Scan(&ph)`. Setelah menerima input, program menggunakan struktur `switch` tanpa ekspresi, yang artinya setiap `case` berisi kondisi yang akan diperiksa satu per satu. Program pertama memeriksa apakah nilai pH berada dalam rentang yang dianggap layak untuk air minum, yaitu antara 6.5 hingga 8.6. Jika kondisi ini terpenuhi, maka program akan mencetak "Air layak diminum". Jika nilai pH kurang dari 6.5 tetapi lebih besar dari 0, atau lebih besar dari 8.6 dan kurang dari atau sama dengan 14, program akan mencetak "air tidak layak diminum". Kondisi ini mencakup pH yang berada di luar rentang yang aman untuk konsumsi. Terakhir, jika nilai pH yang dimasukkan berada di luar rentang yang valid (yaitu lebih kecil dari 0 atau lebih besar dari 14), program akan memberikan pesan kesalahan: "inputan harus antara 0-14", karena pH air seharusnya berada di antara 0 dan 14. Secara keseluruhan, kode ini memeriksa apakah kadar pH air sesuai untuk diminum atau tidak, serta memberikan informasi apabila input yang diberikan tidak valid.

## 2. Latihan Modul 2

**Program** Kendaraan

**Kamus**

var kendaraan string

var Waktu int

**Algoritma**

Input (&Kendaraan)

Input (&Waktu)

**switch kendaraan**

case 1 : jika input motor

aksi motor := 2000 \* waktu,

Output (motor)

case 2 : jika input mobil

aksi mobil := 5000 \* waktu,

Output (mobil)

case kondisi\_3 : jika input truk

aksi truk := 8000 \* waktu,

Output (truk)

**end switch**

**end program**

## Source Code

```
package main

import "fmt"

func main() {
    var ph float64
    fmt.Print("kadar pH Air
: ")
    fmt.Scan(&ph)
    switch {
    case ph >= 6.5 && ph <=
8.6 :
        fmt.Println("Air
layak diminum")
    case ph < 6.5 && ph >
0 || ph > 8.6 && ph <= 14:
        fmt.Println("air
tidak layak diminum")
    case ph > 14 || ph < 0:
        fmt.Println("inputan
harus antara 0-14")
    }
}
```



## Screenshoot Program

The screenshot shows a Go IDE with a project named 'Alpro11'. The Explorer panel on the left shows a file structure with 'OPEN EDITORS' containing 'guided3.go', 'unguided1.go', 'unguided2.go', and 'guided1.go'. The 'ALPRO11' folder contains 'guided1.go', 'guided2.go', 'guided3.go', 'unguided1.go', and 'unguided2.go'. The main editor displays the code for 'unguided2.go', which defines a 'main' function. The code prompts the user for vehicle type and parking duration, calculates the fee based on a switch statement, and prints the result. The bottom panel shows the 'TERMINAL' output, which displays the program's execution for three different inputs: 'motor 3', 'mobil 1', and 'truk 5'. A small window titled 'Best' is also visible in the bottom right corner, displaying the text 'Besthian Guido Rafael Simbolon', '103112430258', and 'IF - 12 - 07'.

```
5 func main() {
7     var waktu int
8     fmt.Println("tipe mobil dan waktu parkir : ")
9     fmt.Scan(&kendaraan, &waktu)
10
11     switch kendaraan {
12     case "motor":
13         motor := 2000 * waktu
14         fmt.Println("Tarif parkir : ", motor)
15     case "mobil":
16         mobil := 5000 * waktu
17         fmt.Println("Tarif parkir : ", mobil)
18     case "truk":
19         truk := 8000 * waktu
20         fmt.Println("Tarif parkir : ", truk)
21     default :
22         fmt.Println("tidak valid")
23     }
24 }
```

PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\unguided2.go"  
tipe mobil dan waktu parkir : motor 3  
Tarif parkir : 6000  
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\unguided2.go"  
tipe mobil dan waktu parkir : mobil 1  
Tarif parkir : 5000  
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\unguided2.go"  
tipe mobil dan waktu parkir : truk 5  
Tarif parkir : 40000  
PS C:\Alpro\Alpro11> []

Besthian Guido Rafael Simbolon  
103112430258  
IF - 12 - 07

## Deskripsi Program

Program di atas berjalan dimulai Ketika program meminta input dari pengguna berupa jenis kendaraan (seperti motor, mobil, atau truk) dan waktu parkir dalam jam. Program ini dimulai dengan mendeklarasikan dua variabel, yaitu kendaraan bertipe string untuk menyimpan jenis kendaraan dan waktu bertipe integer untuk menyimpan durasi parkir. Setelah itu, program meminta pengguna untuk memasukkan kedua nilai tersebut menggunakan `fmt.Scan`. Kemudian, program menggunakan struktur `switch` untuk memeriksa jenis kendaraan yang dimasukkan. Jika jenis kendaraan yang dimasukkan adalah "motor", program akan menghitung tarif parkir dengan mengalikan tarif per jam (2000) dengan waktu parkir. Jika kendaraan yang dimasukkan adalah "mobil", tarif dihitung dengan tarif per jam 5000, dan untuk "truk" tarifnya adalah 8000 per jam. Program kemudian menampilkan tarif parkir yang sesuai berdasarkan jenis kendaraan dan waktu parkir yang dimasukkan. Jika jenis kendaraan yang dimasukkan tidak valid, program akan menampilkan pesan "tidak valid". Dengan demikian, kode ini berfungsi untuk menghitung tarif parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan, serta memberikan informasi jika input tidak valid.

### 3. Latihan Modul 3

#### **Program Aritmatika**

#### **Kamus**

var angka string

#### **Algoritma**

input (&angka)

bagi := angka % 10 == 0 && angka >  
10

kuadrat := angka % 5 == 0 && angka  
> 5

ganjil := angka % 2 != 0

genap := angka % 2 == 0

#### **switch**

case 1 bagi

aksi 1 :

hasil := angka / 10

Output("Kategori = Bilangan  
kelipatan 10")

Output("Hasil penjumlahan  
bilangan berikutnya %d / 10 =  
%d", angka, hasil)

case 2 bagi

aksi 2 :

hasil := angka \* angka

Output("Kategori = Bilangan  
kelipatan 5")

Output("Hasil penjumlahan  
bilangan berikutnya %d^2 =  
%d", angka, hasil)

case 3 bagi

aksi 3 :

angka2 := angka + 1

hasil := angka + angka2

Output("Kategori = Bilangan  
ganjil")

Output("Hasil penjumlahan  
bilangan berikutnya %d + %d =  
%d", angka, angka2, hasil)

case 4 bagi

aksi 4 :

```
angka2 := angka + 1
hasil := angka * angka2
Output ("Bilangan genap")
Output("Hasil penjumlahan
bilangan berikutnya %d * %d =
%d", angka,angka2,hasil)
```

**end switch**

**end program**

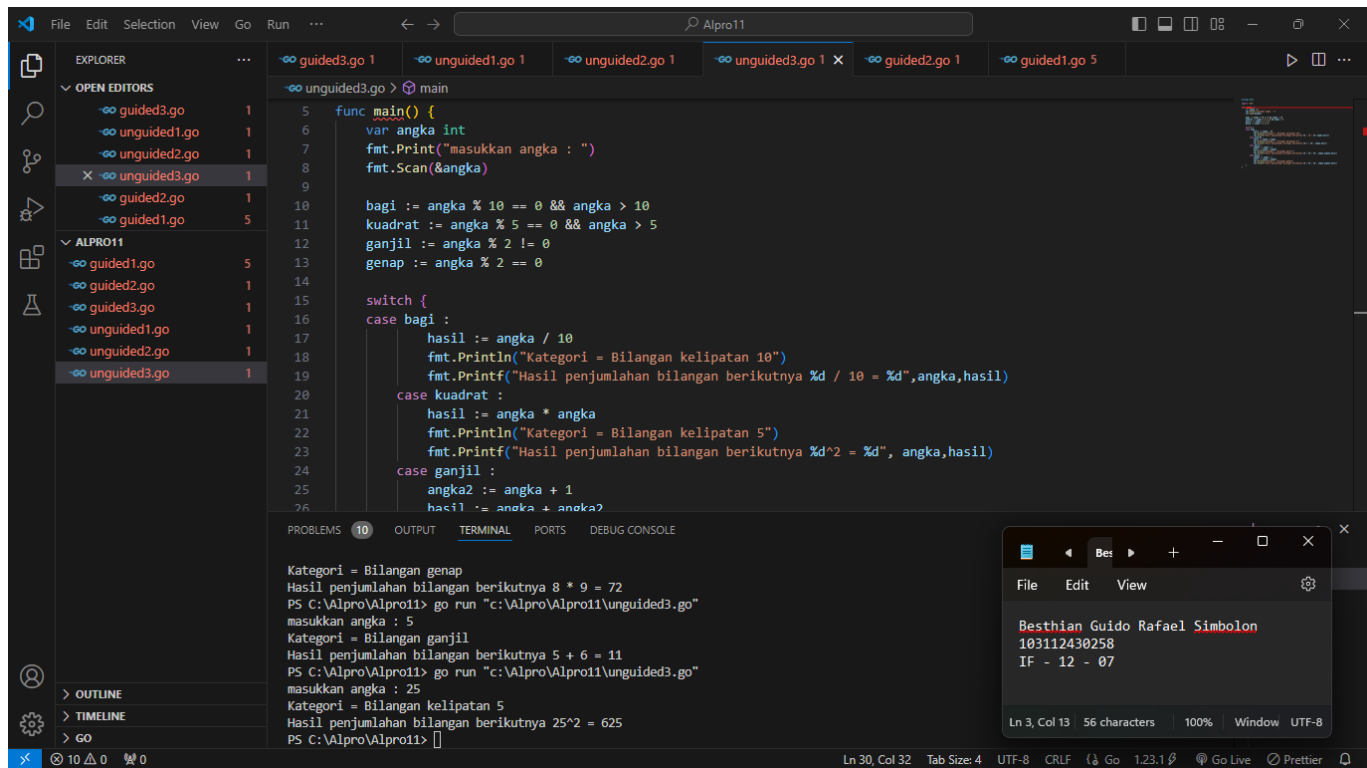
**Source Code**

```
package main

import "fmt"

func main() {
    var ph float64
    fmt.Print("kadar pH Air
: ")
    fmt.Scan(&ph)
    switch {
    case ph >= 6.5 && ph <=
8.6 :
        fmt.Println("Air
layak diminum")
    case ph < 6.5 && ph >
0 || ph > 8.6 && ph <= 14:
        fmt.Println("air
tidak layak diminum")
    case ph > 14 || ph < 0:
        fmt.Println("inputan
harus antara 0-14")
    }
}
```

## Screenshoot Program



```
5 func main() {
6     var angka int
7     fmt.Print("masukkan angka : ")
8     fmt.Scan(&angka)
9
10    bagi := angka % 10 == 0 && angka > 10
11    kuadrat := angka % 5 == 0 && angka > 5
12    ganjil := angka % 2 != 0
13    genap := angka % 2 == 0
14
15    switch {
16    case bagi :
17        hasil := angka / 10
18        fmt.Println("Kategori = Bilangan kelipatan 10")
19        fmt.Printf("Hasil penjumlahan bilangan berikutnya %d / 10 = %d", angka, hasil)
20    case kuadrat :
21        hasil := angka * angka
22        fmt.Println("Kategori = Bilangan kelipatan 5")
23        fmt.Printf("Hasil penjumlahan bilangan berikutnya %d^2 = %d", angka, hasil)
24    case ganjil :
25        angka2 := angka + 1
26        hasil := angka + angka2
```

```
Kategori = Bilangan genap
Hasil penjumlahan bilangan berikutnya 8 * 9 = 72
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\unguided3.go"
masukkan angka : 5
Kategori = Bilangan ganjil
Hasil penjumlahan bilangan berikutnya 5 + 6 = 11
PS C:\Alpro\Alpro11> go run "c:\Alpro\Alpro11\unguided3.go"
masukkan angka : 25
Kategori = Bilangan kelipatan 5
Hasil penjumlahan bilangan berikutnya 25^2 = 625
PS C:\Alpro\Alpro11>
```

## Deskripsi Program

Program di atas berjalan dimulai Ketika program meminta pengguna untuk memasukkan sebuah angka dan kemudian melakukan beberapa pengecekan untuk menentukan kategori angka tersebut, seperti apakah angka tersebut merupakan kelipatan 10, kelipatan 5, angka ganjil, atau angka genap. Program dimulai dengan mendeklarasikan variabel angka untuk menyimpan input dari pengguna, yang akan dibaca menggunakan `fmt.Scan`. Selanjutnya, program mendefinisikan beberapa kondisi untuk kategori angka yang berbeda, yaitu: Kelipatan 10: Jika angka habis dibagi 10 dan lebih besar dari 10, maka program mengategorikan angka sebagai "Bilangan kelipatan 10" dan menghitung hasil pembagian angka dengan 10. Kelipatan 5: Jika angka habis dibagi 5 dan lebih besar dari 5, maka program mengategorikan angka sebagai "Bilangan kelipatan 5" dan menghitung hasil kuadrat dari angka tersebut. Ganjil: Jika angka tidak habis dibagi 2 (angka ganjil), maka program mengategorikan angka sebagai "Bilangan ganjil" dan menghitung hasil penjumlahan antara angka tersebut dengan angka berikutnya ( $\text{angka} + 1$ ). Genap: Jika angka habis dibagi 2 (angka genap), maka program mengategorikan angka sebagai "Bilangan genap" dan menghitung hasil perkalian angka tersebut dengan angka berikutnya ( $\text{angka} * 1$ ). Setelah menentukan kategori angka, program kemudian mencetak kategori tersebut dan hasil perhitungan sesuai dengan kategori yang dipilih. Misalnya, jika angka yang dimasukkan adalah kelipatan 10, program akan mencetak hasil pembagian angka dengan 10, dan jika angka ganjil, program akan mencetak hasil penjumlahan angka dengan angka berikutnya. Secara keseluruhan, program ini bertujuan untuk mengklasifikasikan angka yang dimasukkan ke dalam kategori tertentu dan melakukan operasi matematika berdasarkan kategori tersebut.