

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN**

**MODUL 11  
SWITCH-CASE**



**Disusun Oleh :**  
ERIC SETIAWAN  
NIM 109082500197

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## Guided 1

```
package main

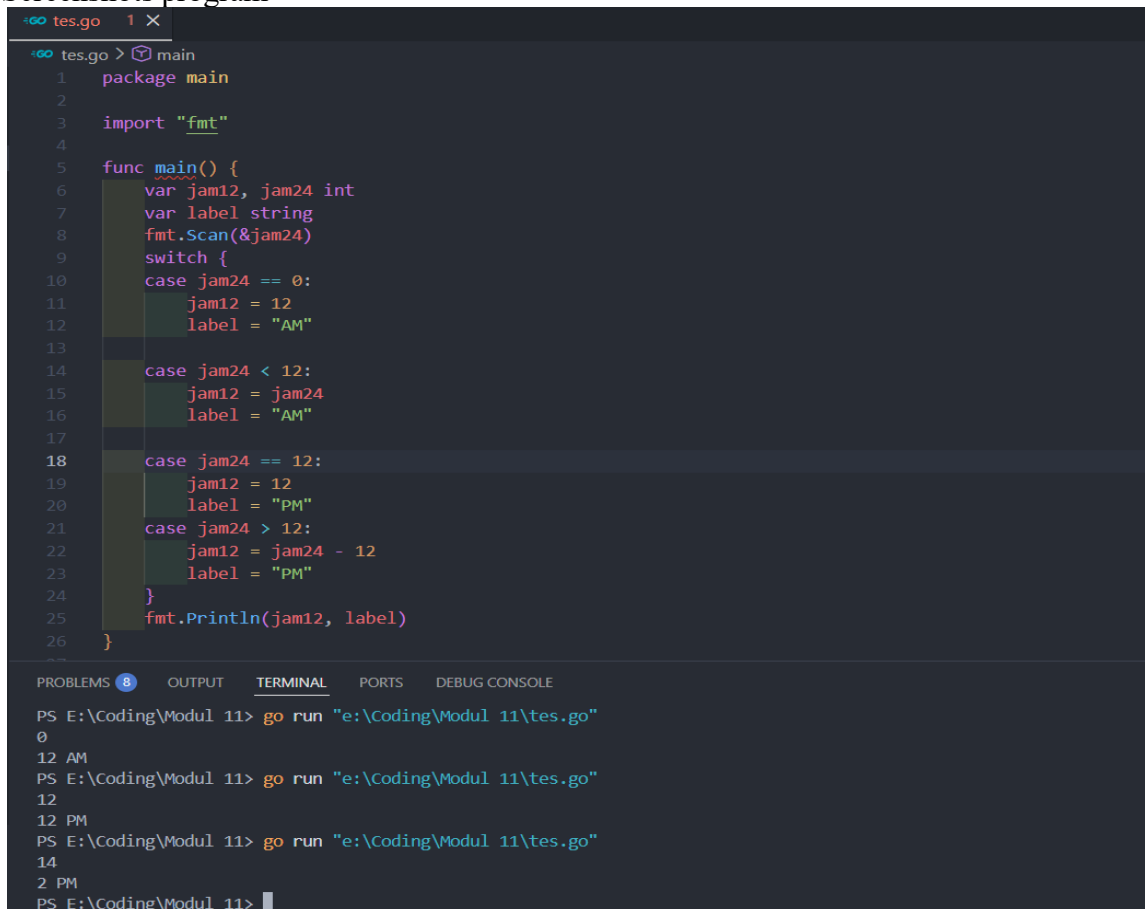
import "fmt"

func main() {
    var jam12, jam24 int
    var label string
    fmt.Scan(&jam24)
    switch {
    case jam24 == 0:
        jam12 = 12
        label = "AM"

    case jam24 < 12:
        jam12 = jam24
        label = "AM"

    case jam24 == 12:
        jam12 = 12
        label = "PM"
    case jam24 > 12:
        jam12 = jam24 - 12
        label = "PM"
    }
    fmt.Println(jam12, label)
```

## Screenshots program



The screenshot shows a Go program in an IDE (likely Visual Studio Code) and its execution output in a terminal. The program is a simple time conversion utility that takes a 24-hour time value and outputs it in 12-hour format with AM/PM.

```
tes.go 1 X
tes.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var jam12, jam24 int
7     var label string
8     fmt.Scan(&jam24)
9     switch {
10    case jam24 == 0:
11        jam12 = 12
12        label = "AM"
13
14    case jam24 < 12:
15        jam12 = jam24
16        label = "AM"
17
18    case jam24 == 12:
19        jam12 = 12
20        label = "PM"
21    case jam24 > 12:
22        jam12 = jam24 - 12
23        label = "PM"
24    }
25    fmt.Println(jam12, label)
26 }
```

The terminal output shows the program being run three times with different inputs:

```
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes.go"
0
12 AM
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes.go"
12
12 PM
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes.go"
14
2 PM
PS E:\Coding\Modul 11>
```

### Deskripsi program:

Program diatas yaitu untuk mengubah format waktu dari sistem 24 jam menjadi 12 jam (AM/PM). Untuk eksekusi awal, program menerima input angka jam dalam format 24 jam. Dengan menggunakan struktur switch tanpa kondisi variabel (switch true), program mengecek nilai input ke dalam beberapa case. Jika inputnya 0, maka diubah jadi 12 AM. Jika kurang dari 12, tetap angka yang sama tapi labelnya AM. Tapi, jika inputnya lebih dari 12, program mengurangi angka tersebut dengan 12 dan memberi label PM.

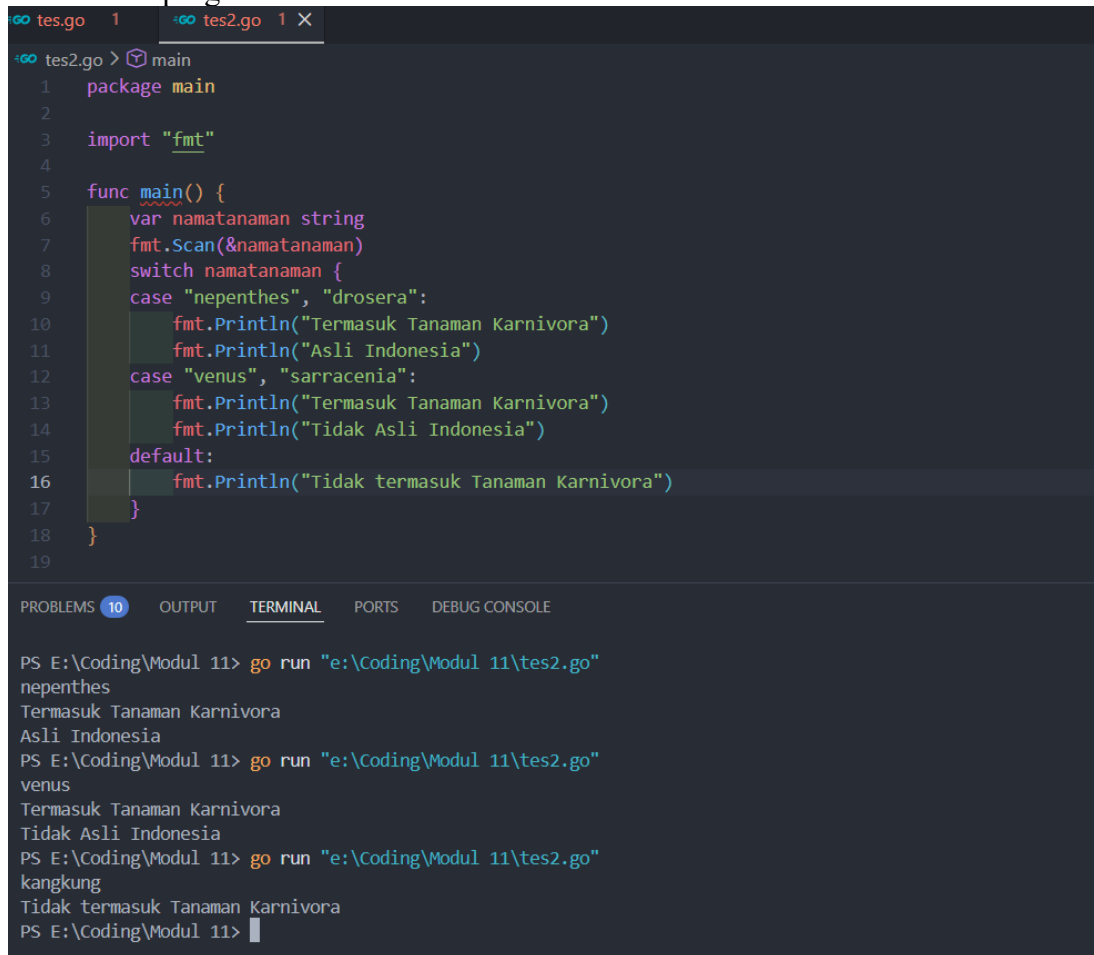
### Guided 2

```
package main

import "fmt"

func main() {
    var namatanaman string
    fmt.Scan(&namatanaman)
    switch namatanaman {
    case "nepenthes", "drosera":
        fmt.Println("Termasuk Tanaman Karnivora")
        fmt.Println("Asli Indonesia")
    case "venus", "sarracenia":
        fmt.Println("Termasuk Tanaman Karnivora")
        fmt.Println("Tidak Asli Indonesia")
    default:
        fmt.Println("Tidak termasuk Tanaman Karnivora")
    }
}
```

## Screenshots program



```
tes.go 1 tes2.go 1 X
tes2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var namatanaman string
7     fmt.Scan(&namatanaman)
8     switch namatanaman {
9     case "nepenthes", "drosera":
10         fmt.Println("Termasuk Tanaman Karnivora")
11         fmt.Println("Asli Indonesia")
12     case "venus", "sarracenia":
13         fmt.Println("Termasuk Tanaman Karnivora")
14         fmt.Println("Tidak Asli Indonesia")
15     default:
16         fmt.Println("Tidak termasuk Tanaman Karnivora")
17     }
18 }
19

PROBLEMS 10 OUTPUT TERMINAL PORTS DEBUG CONSOLE

PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes2.go"
nepenthes
Termasuk Tanaman Karnivora
Asli Indonesia
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes2.go"
venus
Termasuk Tanaman Karnivora
Tidak Asli Indonesia
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes2.go"
kangkung
Tidak termasuk Tanaman Karnivora
PS E:\Coding\Modul 11>
```

## Deskripsi program:

Program diatas yaitu untuk mengidentifikasi jenis tanaman, apakah termasuk karnivora dan asalnya dari mana. Untuk eksekusi awal, program menerima input nama tanaman dalam bentuk string. Dengan menggunakan struktur switch berdasarkan variabel nama tanaman, program mencocokkan input dengan daftar yang ada. Jika inputnya "nepenthes" atau "drosera", outputnya karnivora asli Indonesia. Sedangkan jika "venus" atau "sarracenia", outputnya karnivora tapi bukan asli Indonesia. Tapi, jika nama tanaman tidak ada di dalam daftar case, program otomatis masuk ke default dan menyatakan bukan tanaman karnivora.

### Guided 3

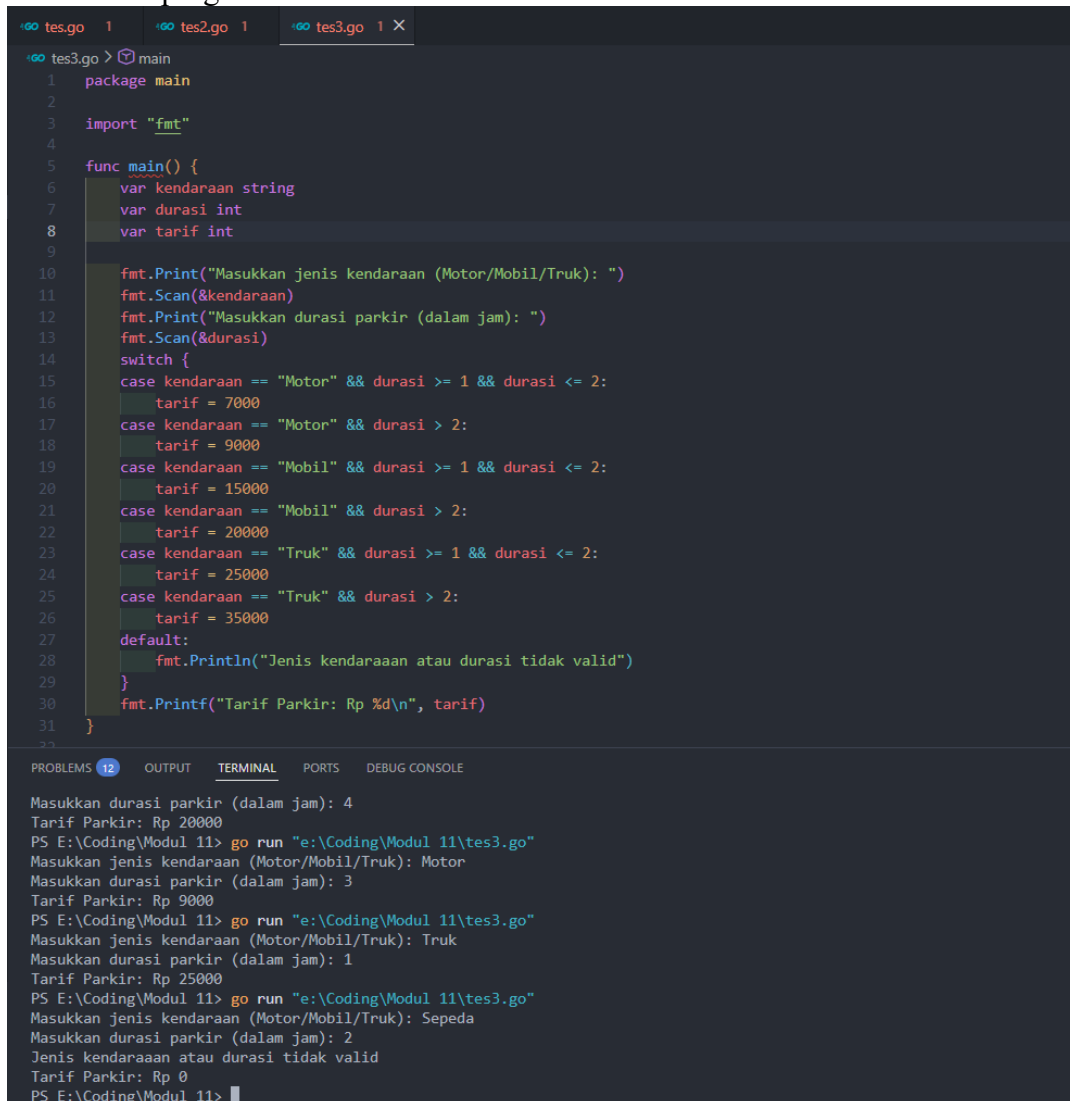
```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi int
    var tarif int

    fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
    fmt.Scan(&kendaraan)
    fmt.Print("Masukkan durasi parkir (dalam jam): ")
    fmt.Scan(&durasi)
    switch {
    case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
        tarif = 7000
    case kendaraan == "Motor" && durasi > 2:
        tarif = 9000
    case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
        tarif = 15000
    case kendaraan == "Mobil" && durasi > 2:
        tarif = 20000
    case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
        tarif = 25000
    case kendaraan == "Truk" && durasi > 2:
        tarif = 35000
    default:
        fmt.Println("Jenis kendaraan atau durasi tidak valid")
    }
    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
}
```

## Screenshots program



```
tes3.go 1 tes2.go 1 tes3.go 1 X
tes3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var durasi int
8     var tarif int
9
10    fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
11    fmt.Scan(&kendaraan)
12    fmt.Print("Masukkan durasi parkir (dalam jam): ")
13    fmt.Scan(&durasi)
14    switch {
15    case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
16        tarif = 7000
17    case kendaraan == "Motor" && durasi > 2:
18        tarif = 9000
19    case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
20        tarif = 15000
21    case kendaraan == "Mobil" && durasi > 2:
22        tarif = 20000
23    case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
24        tarif = 25000
25    case kendaraan == "Truk" && durasi > 2:
26        tarif = 35000
27    default:
28        fmt.Println("Jenis kendaraan atau durasi tidak valid")
29    }
30    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
31 }
32
```

PROBLEMS 12 OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir: Rp 9000
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\tes3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Sepeda
Masukkan durasi parkir (dalam jam): 2
Jenis kendaraan atau durasi tidak valid
Tarif Parkir: Rp 0
PS E:\Coding\Modul 11>
```

## Deskripsi program:

Program diatas yaitu untuk menghitung biaya parkir bertingkat berdasarkan jenis kendaraan dan lamanya parkir. Untuk eksekusi awal, program menerima dua input: jenis kendaraan dan durasi jam. Dengan menggunakan struktur switch, program mengecek kombinasi antara jenis kendaraan dan durasi secara spesifik. Semisal disini kendaraannya Motor dan durasinya di antara 1-2 jam, tarifnya beda dengan yang durasinya lebih dari 2 jam. Program akan mencocokkan kondisi yang paling sesuai dan menyimpan harga ke variabel tarif. Tapi, jika inputnya ngawur atau tidak sesuai kriteria manapun, program akan menjalankan default untuk memberi pesan error.

## Tugas 1

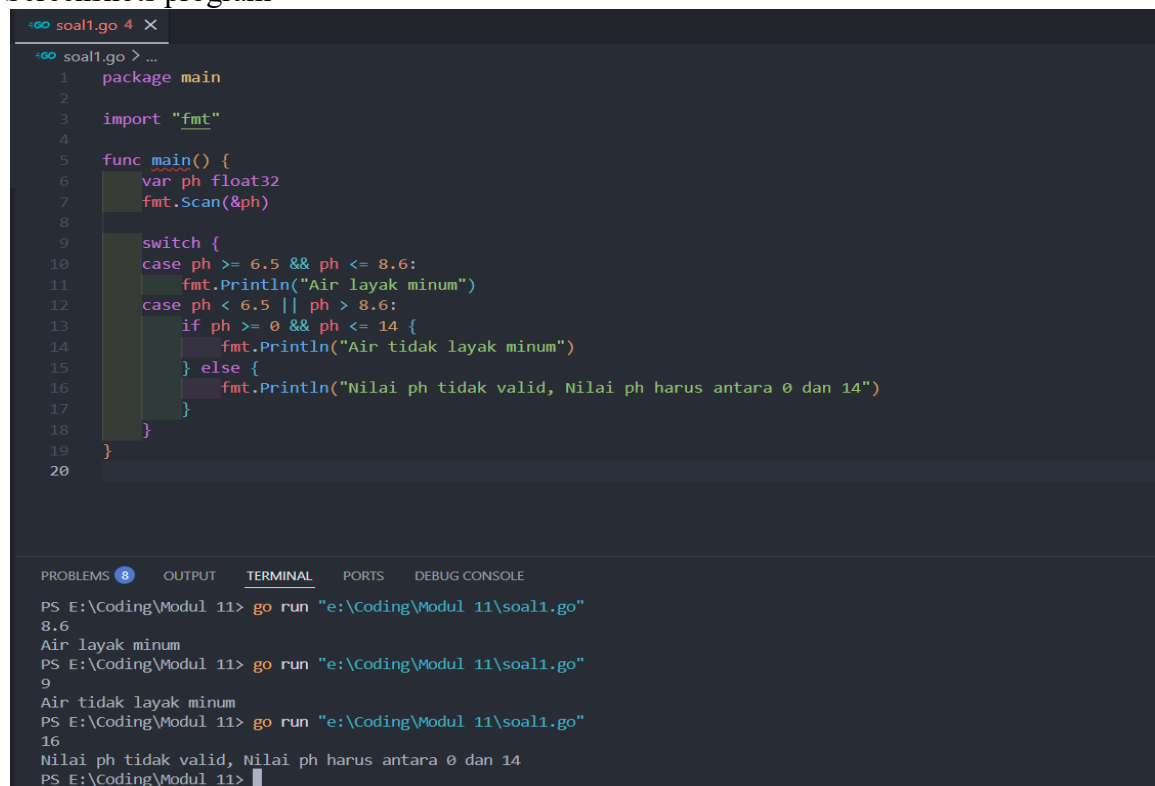
```
package main

import "fmt"

func main() {
    var ph float32
    fmt.Scan(&ph)

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

## Screenshots program



```
soal1.go 4 X
soal1.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float32
7     fmt.Scan(&ph)
8
9     switch {
10    case ph >= 6.5 && ph <= 8.6:
11        fmt.Println("Air layak minum")
12    case ph < 6.5 || ph > 8.6:
13        if ph >= 0 && ph <= 14 {
14            fmt.Println("Air tidak layak minum")
15        } else {
16            fmt.Println("Nilai ph tidak valid, Nilai ph harus antara 0 dan 14")
17        }
18    }
19 }
20

PROBLEMS 0 OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal1.go"
8.6
Air layak minum
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal1.go"
9
Air tidak layak minum
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal1.go"
16
Nilai ph tidak valid, Nilai ph harus antara 0 dan 14
PS E:\Coding\Modul 11> 
```

## Deskripsi:

Program diatas yaitu untuk mengecek kualitas air minum berdasarkan kadar phnya. Untuk eksekusi awal, program menerima input angka desimal (float) sebagai nilai ph. Dengan menggunakan struktur switch yang mengecek kondisi batasan nilai, program memfilter inputnya. Jika nilai ph di luar rentang 0 sampai 14, program akan memberi peringatan input tidak valid. Jika ph berada di angka aman (antara 6.5 sampai 8.6), outputnya "Air layak minum". Tapi, jika ph nya terlalu asam atau terlalu basa (di luar

rentang aman tapi masih 0-14), program menyatakan air tidak layak minum.

## Tugas 2

```
package main

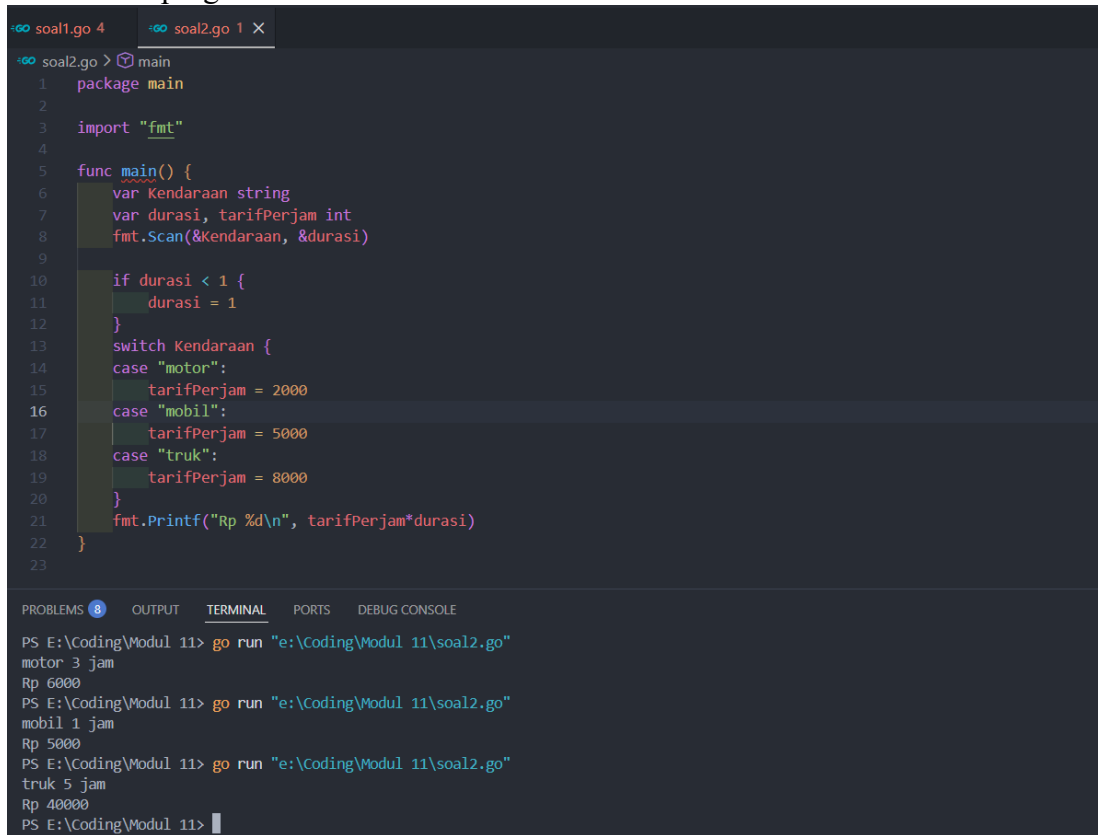
import "fmt"

func main() {
    var Kendaraan string
    var durasi, tarifPerjam int
    fmt.Scan(&Kendaraan, &durasi)

    if durasi < 1 {
        durasi = 1
    }
    switch Kendaraan {
    case "motor":
        tarifPerjam = 2000
    case "mobil":
        tarifPerjam = 5000
    case "truk":
        tarifPerjam = 8000
    }
    fmt.Printf("Rp %d\n", tarifPerjam*durasi)
}
```



## Screenshots program



```
soal2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var Kendaraan string
7     var durasi, tarifPerjam int
8     fmt.Scan(&Kendaraan, &durasi)
9
10    if durasi < 1 {
11        durasi = 1
12    }
13    switch Kendaraan {
14    case "motor":
15        tarifPerjam = 2000
16    case "mobil":
17        tarifPerjam = 5000
18    case "truk":
19        tarifPerjam = 8000
20    }
21    fmt.Printf("Rp %d\n", tarifPerjam*durasi)
22 }
23
```

PROBLEMS 8 OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal2.go"
motor 3 jam
Rp 6000
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal2.go"
mobil 1 jam
Rp 5000
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal2.go"
truk 5 jam
Rp 40000
PS E:\Coding\Modul 11>
```

## Deskripsi:

Program diatas yaitu untuk menghitung total tarif parkir dengan skema harga per jam. Untuk eksekusi awal, program menerima input jenis kendaraan dan durasi. Program memvalidasi durasi dulu. Jika durasi kurang dari 1 jam, otomatis dianggap 1 jam. Selanjutnya, dengan menggunakan struktur switch pada variabel kendaraan, program menentukan harga dasar per jamnya (motor 2000, mobil 5000, truk 8000). Setelah harga ketemu, program tinggal mengalikan harga tersebut dengan durasi untuk mendapatkan total biaya.

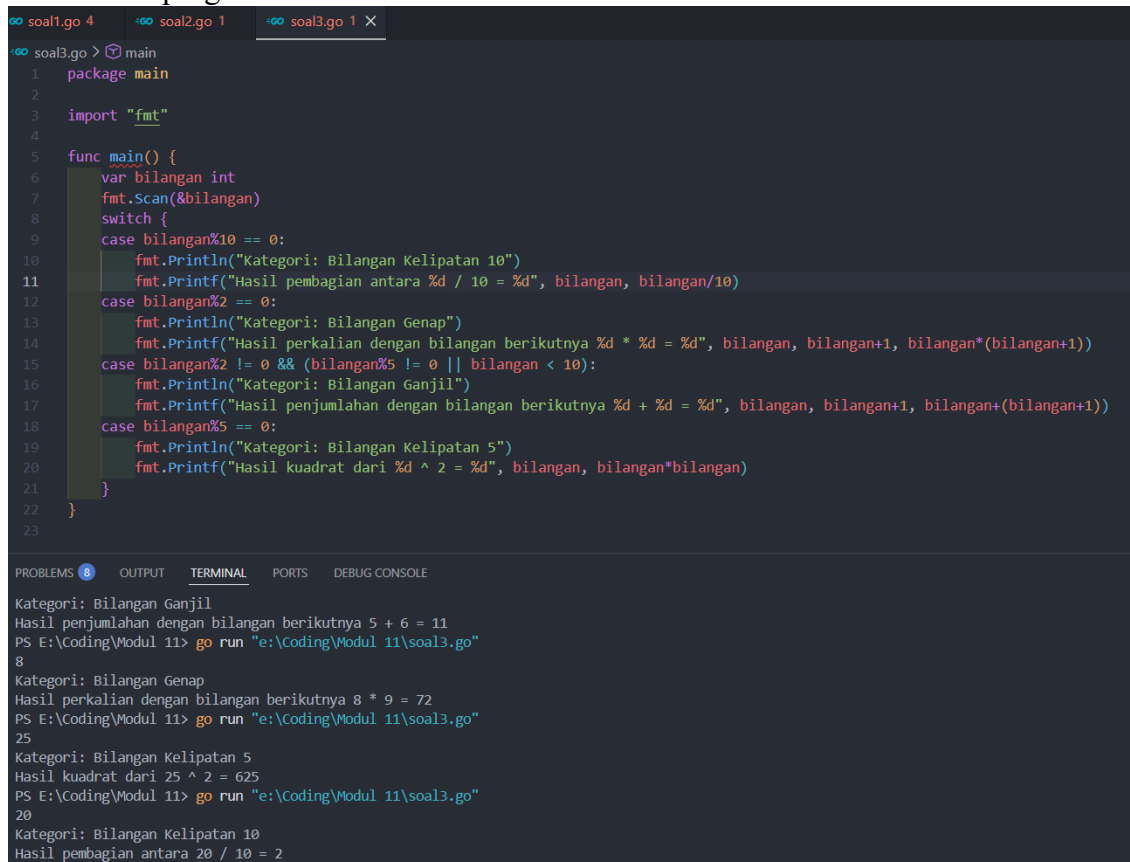
## Tugas

```
package main

import "fmt"

func main() {
    var bilangan int
    fmt.Scan(&bilangan)
    switch {
    case bilangan%10 == 0:
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        fmt.Printf("Hasil pembagian antara %d / 10 = %d", bilangan,
bilangan/10)
    case bilangan%2 == 0:
        fmt.Println("Kategori: Bilangan Genap")
        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d",
bilangan, bilangan+1, bilangan*(bilangan+1))
    case bilangan%2 != 0 && (bilangan%5 != 0 || bilangan < 10):
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d =
%d", bilangan, bilangan+1, bilangan+(bilangan+1))
    case bilangan%5 == 0:
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d", bilangan,
bilangan*bilangan)
    }
}
```

## Screenshots program



The screenshot shows a Go program in a text editor with three tabs: `soal1.go 4`, `soal2.go 1`, and `soal3.go 1 X`. The active tab is `soal3.go`, which contains the following code:

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

Below the editor is a terminal window with tabs: **PROBLEMS**, **OUTPUT**, **TERMINAL**, **PORTS**, and **DEBUG CONSOLE**. The **TERMINAL** tab is active, showing the execution of the program with the following output:

```
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal3.go"
8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal3.go"
25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^ 2 = 625
PS E:\Coding\Modul 11> go run "e:\Coding\Modul 11\soal3.go"
20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
```

### Deskripsi:

Program diatas yaitu untuk mengkategorikan bilangan bulat ke dalam pola aritmatika tertentu sesuai prioritasnya. Untuk eksekusi awal, program meminta masukan berupa bilangan bulat yang disimpan dalam variabel `bilangan`. Dengan menggunakan struktur `switch` tanpa kondisi variabel, program memilah angka tersebut berdasarkan urutan logika. Pertama, dicek apakah angka itu kelipatan 10. Kedua, dicek apakah termasuk bilangan genap. Ketiga, untuk kategori Ganjil, program menggunakan logika spesifik (operator `&&` dan `||`). Tujuannya supaya angka ganjil kecil seperti 5 tetap dianggap Ganjil, tapi angka ganjil besar yang kelipatan 5, semisal 25, akan dilewatkan. Nah, jika angka tersebut lolos dari filter ganjil tadi, maka dia akan masuk ke case terakhir dan dianggap sebagai kategori Kelipatan 5.