

# **LAPORAN PRA KTIKUM**

**Algoritma Pemrograman**

**MODUL 11**

**Switch case**



**Disusun oleh:**

**MUHAMMAD TETUKO KEMAL PASHA**

**109082500181**

**S1IF-13-04**

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## LATIHAN KELAS – GUIDED

### 1. Guided 1

#### Source Code

```
package main

import "fmt"

func main() {
    var jam int
    var jamFormat int
    var perihai string

    fmt.Print("Masukkan jam (0-23): ")
    fmt.Scan(&jam)

    switch jam {
    case 0:
        jamFormat = 12
        perihai = "AM"
    case 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11:
        jamFormat = jam
        perihai = "AM"
    case 12:
        jamFormat = 12
        perihai = "PM"
    case 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23:
        jamFormat = jam - 12
        perihai = "PM"
    default:
        fmt.Println("Jam tidak valid! Masukkan jam 0-23")
    }
```

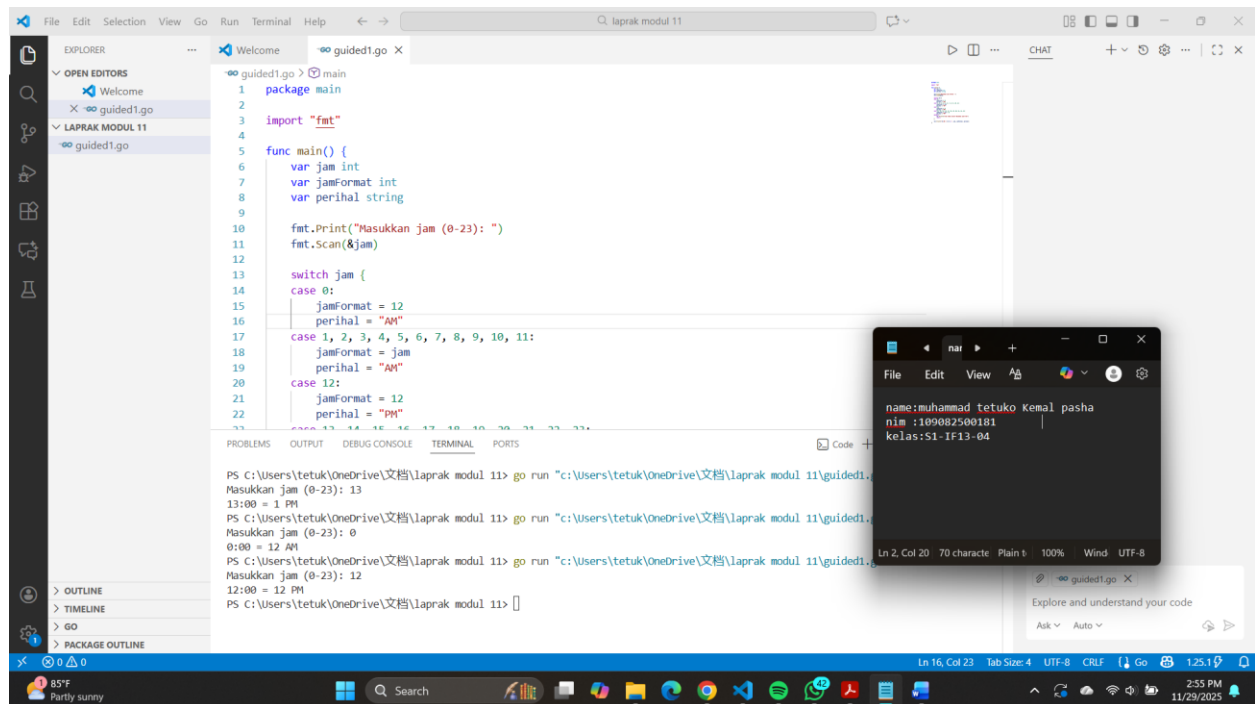
```

        return
    }

    fmt.Printf("%d:00 = %d %s\n", jam, jamFormat, perihal)
}

```

## Screenshot program



## Deskripsi program

Program pada gambar merupakan aplikasi sederhana dalam bahasa Go yang berfungsi untuk mengonversi format waktu dari 24 jam menjadi format 12 jam (AM/PM). Pengguna diminta memasukkan jam dalam rentang 0–23, kemudian program memproses input tersebut menggunakan struktur `switch` untuk menentukan apakah waktu tersebut termasuk AM atau PM, sekaligus mengubah nilai `jamFormat` agar sesuai dengan format 12 jam; misalnya jam 0 menjadi 12 AM, jam 1–11 tetap AM, jam 12 menjadi 12 PM, dan jam 13–23 dikurangi 12 sehingga berubah menjadi 1–11 PM. Setelah pengolahan selesai, program menampilkan hasil konversi dalam bentuk jam 12 jam lengkap dengan keterangan AM atau PM.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    var nama_tanaman string
    fmt.Scan(&nama_tanaman)
    switch nama_tanaman {
    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.")
    }
}
```

## Screenshot program

```
package main

import "fmt"

func main() {
    var nama_tanaman string
    fmt.Scan(&nama_tanaman)
    switch nama_tanaman {
    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.")
    }
}
```

```
PS C:\Users\tetuk\OneDrive\laprak modul 11> go run "c:\Users\tetuk\OneDrive\laprak modul 11\guided2karnivora.go"
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS C:\Users\tetuk\OneDrive\laprak modul 11> go run "c:\Users\tetuk\OneDrive\laprak modul 11\guided2karnivora.go"
venus
Termasuk Tanaman Karnivora.
Tidak Asli Indonesia.
PS C:\Users\tetuk\OneDrive\laprak modul 11> go run "c:\Users\tetuk\OneDrive\laprak modul 11\guided2karnivora.go"
karedok
Tidak termasuk Tanaman Karnivora.
PS C:\Users\tetuk\OneDrive\laprak modul 11>
```

```
name:muhammad tetuko Kemal pasha
nim :109082500181
kelas:S1-IF13-04
```

## Deskripsi program

Program pada gambar adalah program Go yang berfungsi untuk mengidentifikasi apakah suatu tanaman termasuk golongan tanaman karnivora atau bukan berdasarkan nama yang dimasukkan pengguna. Program meminta input nama tanaman melalui `fmt.Scan(&nama_tanaman)`, kemudian menggunakan struktur `switch` untuk membandingkan input dengan beberapa pilihan, yaitu "nepenthes", "drosera" untuk tanaman asli Indonesia, serta "venus" dan "sarracenia" yang juga termasuk tanaman karnivora. Jika nama tanaman sesuai dengan salah satu kasus tersebut, program menampilkan bahwa tanaman tersebut merupakan tanaman karnivora, disertai keterangan apakah berasal dari Indonesia atau tidak. Jika input tidak cocok dengan semua kasus, maka program menampilkan pesan bahwa tanaman tersebut tidak termasuk tanaman karnivora. Program ini sederhana namun efektif untuk mengecek klasifikasi dasar jenis tanaman.

### 3. Guided 3

#### Source Code

```
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
    }
```

```

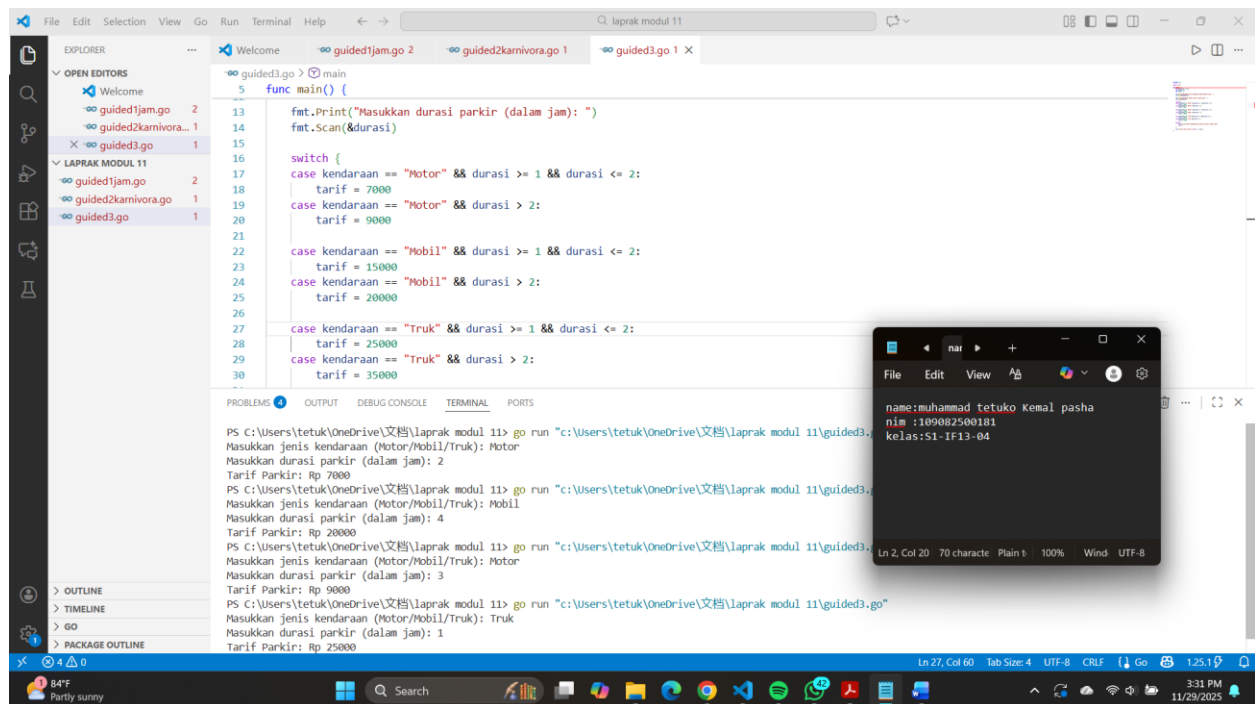
default:

    fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
    return
}

fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
}

```

## Screenshot program



## Deskripsi program

Program pada gambar adalah program Go yang digunakan untuk menghitung tarif parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Program meminta input berupa jenis kendaraan (Motor, Mobil, atau Truk) serta durasi parkir dalam jam, kemudian menggunakan struktur switch tanpa ekspresi untuk mengevaluasi beberapa kondisi sekaligus. Setiap kondisi memeriksa apakah jenis kendaraan sesuai dan durasi berada dalam rentang tertentu, lalu menetapkan tarif yang sesuai: Motor dikenakan Rp7.000 untuk durasi 1–2 jam dan Rp9.000 jika lebih dari 2 jam; Mobil dikenakan Rp15.000 untuk durasi 1–2 jam dan Rp20.000 jika lebih lama; sedangkan Truk dikenakan Rp25.000 untuk durasi 1–2 jam dan Rp35.000 jika lebih dari 2 jam. Setelah kondisi terpenuhi, program menampilkan tarif parkir sesuai aturan yang berlaku. Program ini membantu memberikan perhitungan tarif secara otomatis berdasarkan input yang diberikan.

## 1.soal 1

### Source code

```
package main

import "fmt"

func main() {
    var pH float64

    fmt.Print("Masukkan kadar pH: ")
    fmt.Scan(&pH)

    if pH < 0 || pH > 14 {
        fmt.Println("Input tidak valid, rentang pH 0 - 14")
        return
    }

    var kategori int
    switch {
    case pH < 6.5:
        kategori = 0
    case pH >= 6.5 && pH <= 8.6:
        kategori = 1
    case pH > 8.6:
        kategori = 2
    }

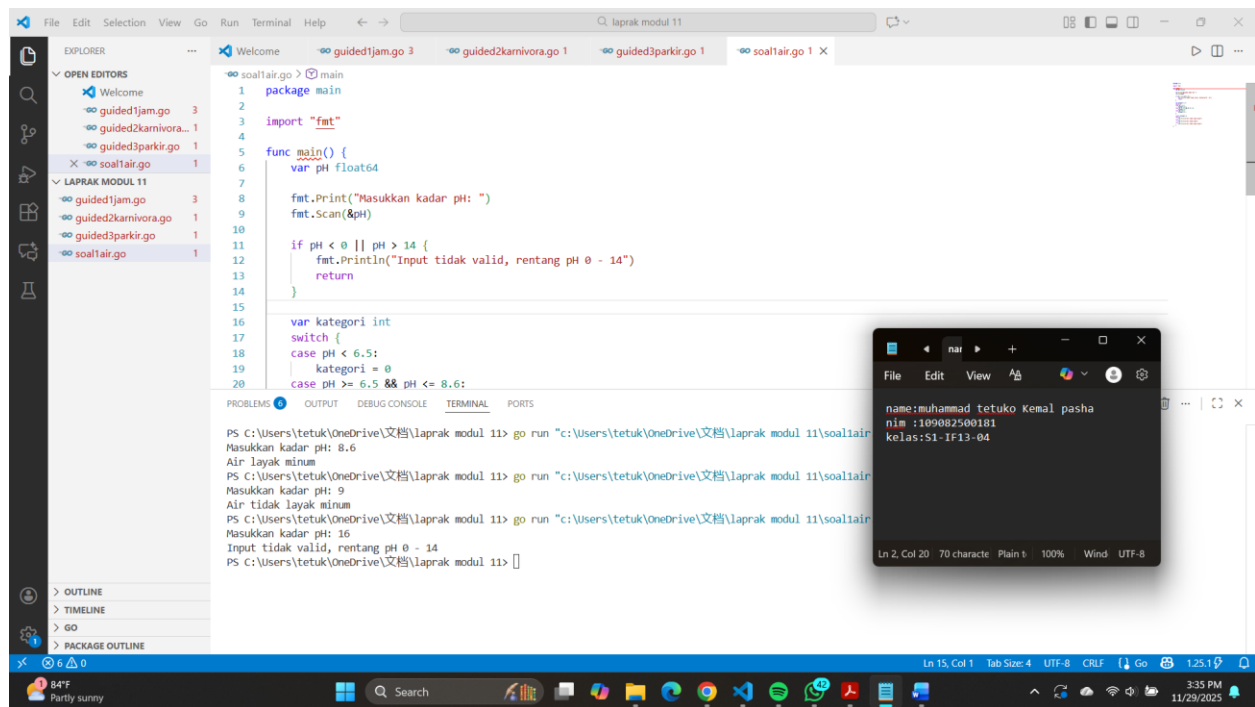
    switch kategori {
    case 0:
        fmt.Println("Air tidak layak minum")
```

```

case 1:
    fmt.Println("Air layak minum")
case 2:
    fmt.Println("Air tidak layak minum")
}
}

```

## Screenshot program



### **Deskripsi program**

Program pada gambar adalah program Go yang berfungsi untuk menentukan kategori kualitas air berdasarkan nilai pH yang dimasukkan pengguna. Setelah pengguna memasukkan nilai pH, program terlebih dahulu memeriksa apakah input berada dalam rentang valid 0 hingga 14; jika nilai berada di luar rentang tersebut, program langsung menampilkan pesan bahwa input tidak valid dan menghentikan eksekusi. Jika nilai pH valid, program melanjutkan dengan struktur switch untuk menentukan kategori air: pH < 6.5 dikategorikan sebagai air terlalu asam, pH antara 6.5 hingga 8.5 sebagai air layak minum, dan pH > 8.5 hingga 14 sebagai air terlalu basa. Berdasarkan kategori yang sesuai, program kemudian menampilkan hasil evaluasi kualitas air kepada pengguna.

## 2.soal 2

### Source code

```
package main

import "fmt"

func main() {
    var jenisKendaraan string
    var durasi int
    var tarifPerJam, totalBiaya int

    fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk): ")
    fmt.Scan(&jenisKendaraan)

    fmt.Print("Masukkan durasi parkir (jam): ")
    fmt.Scan(&durasi)

    if durasi < 1 {
        durasi = 1
    }

    switch jenisKendaraan {
    case "motor":
        tarifPerJam = 2000
    case "mobil":
        tarifPerJam = 5000
    case "truk":
        tarifPerJam = 8000
    default:
```

```

        fmt.Println("Jenis kendaraan tidak valid")

        return
    }

    totalBiaya = tarifPerJam * durasi

    fmt.Printf("Total biaya parkir: Rp %d\n", totalBiaya)
}

```

## Screenshot program

The screenshot shows a Go program in VS Code. The Explorer pane on the left shows the project structure with files like `guided1jam.go`, `guided2karnivora.go`, `guided3parkir.go`, `soal1air.go`, `kendaraan[1].go`, and `soal2kendaraan.go`. The main editor shows the source code for `soal2kendaraan.go`, which defines a `main` function. The function prompts the user to enter the vehicle type and duration, then calculates the parking fee based on the vehicle type and duration. The terminal output shows the program being run with inputs for vehicle type and duration, resulting in calculated parking fees.

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var jenisKendaraan string
7     var durasi int
8     var tarifPerJam, totalBiaya int
9
10    fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk): ")
11    fmt.Scan(&jenisKendaraan)
12
13    fmt.Print("Masukkan durasi parkir (jam): ")
14    fmt.Scan(&durasi)
15
16
17    if durasi < 1 {
18        durasi = 1
19    }
20

```

```

PS C:\Users\tetuk\OneDrive\文档\laprak modul 11> go run "c:\Users\tetuk\OneDrive\文档\laprak modul 11\soal2kendaraan.go"
Masukkan jenis kendaraan (motor/mobil/truk): motor
Masukkan durasi parkir (jam): 3
Total biaya parkir: Rp 6000
PS C:\Users\tetuk\OneDrive\文档\laprak modul 11> go run "c:\Users\tetuk\OneDrive\文档\laprak modul 11\soal2kendaraan.go"
Masukkan jenis kendaraan (motor/mobil/truk): mobil
Masukkan durasi parkir (jam): 1
Total biaya parkir: Rp 5000
PS C:\Users\tetuk\OneDrive\文档\laprak modul 11> go run "c:\Users\tetuk\OneDrive\文档\laprak modul 11\soal2kendaraan.go"
Masukkan jenis kendaraan (motor/mobil/truk): truk
Masukkan durasi parkir (jam): 5
Total biaya parkir: Rp 40000
PS C:\Users\tetuk\OneDrive\文档\laprak modul 11>

```

## Deskripsi program

Kode program Go, yang tersimpan dalam file soal2kendaraan.go dan berada di package main, tampaknya berfungsi untuk **menghitung biaya parkir** berdasarkan jenis kendaraan (motor/mobil/truk) dan durasi parkir dalam jam, meskipun logika perhitungannya tidak terlihat secara lengkap. Bagian *terminal* atau *debug console* di bawah menunjukkan tiga kali eksekusi program dengan input yang berbeda dan menampilkan **"Total biaya parkir"** yang dihasilkan (Rp 6000, Rp 5000, dan Rp 80000), mengindikasikan bahwa program tersebut telah diuji.**3.soal 3**

## Source code

```
package main

import "fmt"

func main() {
    var bilangan int

    fmt.Print("Masukkan bilangan bulat: ")
    fmt.Scan(&bilangan)

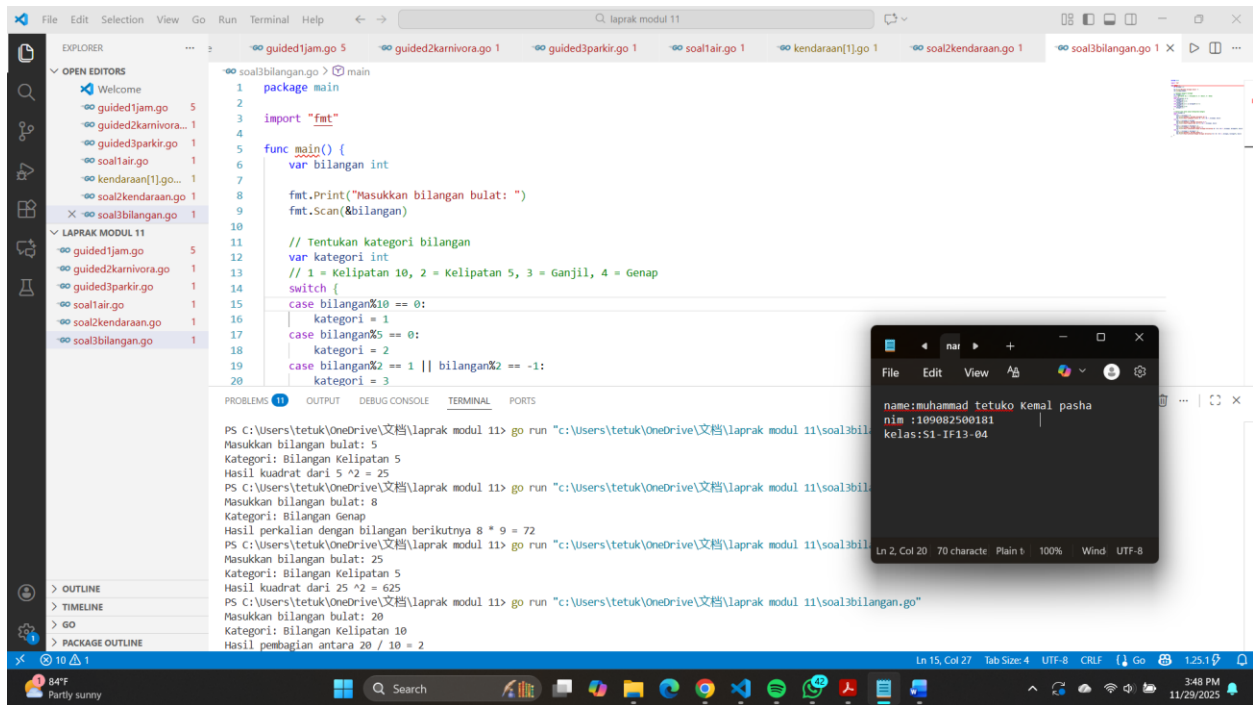
    // Tentukan kategori bilangan
    var kategori int
    // 1 = Kelipatan 10, 2 = Kelipatan 5, 3 = Ganjil, 4 = Genap
    switch {
    case bilangan%10 == 0:
        kategori = 1
    case bilangan%5 == 0:
        kategori = 2
    case bilangan%2 == 1 || bilangan%2 == -1:
        kategori = 3
    case bilangan%2 == 0:
        kategori = 4
    }
}
```

```

// Switch case untuk output berdasarkan kategori
switch kategori {
case 1:
    hasil := bilangan / 10
    fmt.Printf("Kategori: Bilangan Kelipatan 10\n")
    fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", bilangan,
hasil)
case 2:
    hasil := bilangan * bilangan
    fmt.Printf("Kategori: Bilangan Kelipatan 5\n")
    fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", bilangan, hasil)
case 3:
    hasil := bilangan + (bilangan + 1)
    fmt.Printf("Kategori: Bilangan Ganjil\n")
    fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d +
%d = %d\n", bilangan, bilangan+1, hasil)
case 4:
    hasil := bilangan * (bilangan + 1)
    fmt.Printf("Kategori: Bilangan Genap\n")
    fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d
= %d\n", bilangan, bilangan+1, hasil)
}
}

```

## Screenshot program



The screenshot shows a Go program in a VS Code editor. The program is located in `soal3bilangan.go` and uses a `switch-case` statement to categorize a number based on its divisibility by 10, 5, and 2. A terminal window is overlaid on the editor, showing the program's execution with sample inputs and their corresponding outputs.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7
8     fmt.Print("Masukkan bilangan bulat: ")
9     fmt.Scan(&bilangan)
10
11     // Tentukan kategori bilangan
12     var kategori int
13     // 1 = Kelipatan 10, 2 = Kelipatan 5, 3 = Ganjil, 4 = Genap
14     switch {
15     case bilangan%10 == 0:
16         kategori = 1
17     case bilangan%5 == 0:
18         kategori = 2
19     case bilangan%2 == 1 || bilangan%2 == -1:
20         kategori = 3
```

Terminal Output:

```
PS C:\Users\tetuk\OneDrive\lengkap modul 11> go run "c:\Users\tetuk\OneDrive\lengkap modul 11\soal3bilangan.go"
Masukkan bilangan bulat: 5
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 5 * 2 = 25
PS C:\Users\tetuk\OneDrive\lengkap modul 11> go run "c:\Users\tetuk\OneDrive\lengkap modul 11\soal3bilangan.go"
Masukkan bilangan bulat: 8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Users\tetuk\OneDrive\lengkap modul 11> go run "c:\Users\tetuk\OneDrive\lengkap modul 11\soal3bilangan.go"
Masukkan bilangan bulat: 25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 * 2 = 625
PS C:\Users\tetuk\OneDrive\lengkap modul 11> go run "c:\Users\tetuk\OneDrive\lengkap modul 11\soal3bilangan.go"
Masukkan bilangan bulat: 20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
```

## Deskripsi program

Tentu, berikut adalah penjelasan dari kode program Go tersebut dalam satu paragraf:

Kode program Go yang terdapat dalam file `soalbilangan.go` ini berfungsi untuk mengategorikan bilangan bulat yang dimasukkan oleh pengguna menggunakan *statement* `switch-case`. Setelah meminta dan menerima input bilangan bulat, program ini menentukan kategorinya menjadi: Kelipatan 10 (jika habis dibagi 10), Kelipatan 5 atau 3 (jika habis dibagi 5 atau 3), Ganjil (jika tidak habis dibagi 2), atau Genap (sebagai *default*). Bagian konsol menunjukkan bahwa program telah diuji dengan input seperti 15 (dikategorikan Kelipatan 5), 72 (dikategorikan Genap), dan 20 (dikategorikan Kelipatan 10), membuktikan fungsionalitas kategorisasi tersebut, sekaligus menampilkan hasil operasi aritmatika tambahan yang menyertai setiap hasil.