

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

**MODUL 11**  
**SWITCH-CASE**



**Disusun oleh:**  
**EDWARD ABIMAS SURYA HATTA**  
**109082500171**  
**S1IF-13-04**

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

**2025**

## LATIHAN KELAS – GUIDED

### 1. Guided 1

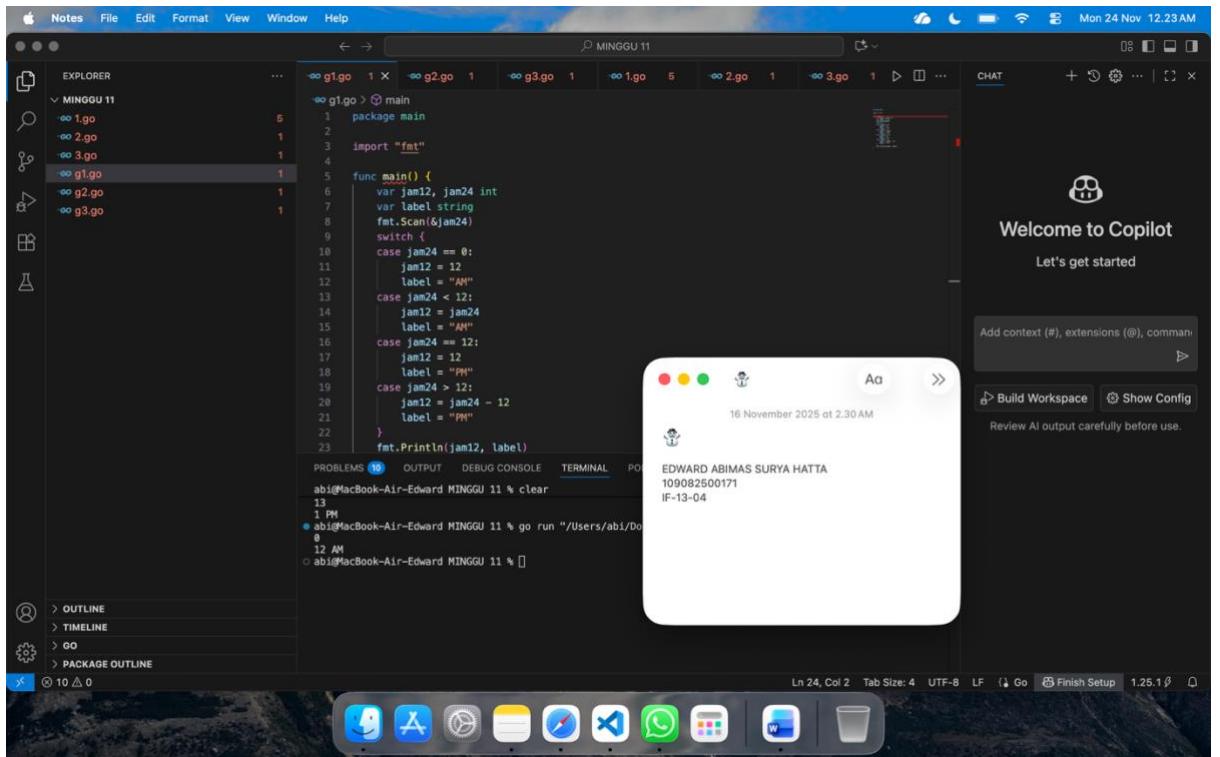
#### Source Code

```
package main

import "fmt"

func main() {
    var 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)
}
```

#### Screenshot program



## Deskripsi program

Program ini dirancang untuk mengonversi format waktu 24 jam menjadi format 12 jam (AM/PM). Setelah pengguna memasukkan angka jam, sistem akan mengevaluasi nilai tersebut menggunakan struktur kendali *switch* tanpa ekspresi yang memungkinkan pengecekan kondisi secara fleksibel. Terdapat empat kondisi utama yang ditangani secara spesifik: jika jam menunjukkan angka 0, sistem akan mengubahnya menjadi jam 12 dengan label "AM"; jika angka berada di bawah 12 namun bukan nol, nilai tetap dipertahankan dengan label "AM"; jika tepat angka 12, label berubah menjadi "PM"; dan jika angka melebihi 12, nilai tersebut akan dikurangi 12 untuk mendapatkan ekuivalen waktu sore atau malam dengan label "PM". Hasil akhir berupa angka jam yang telah disesuaikan beserta labelnya kemudian ditampilkan ke layer.

## 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.")

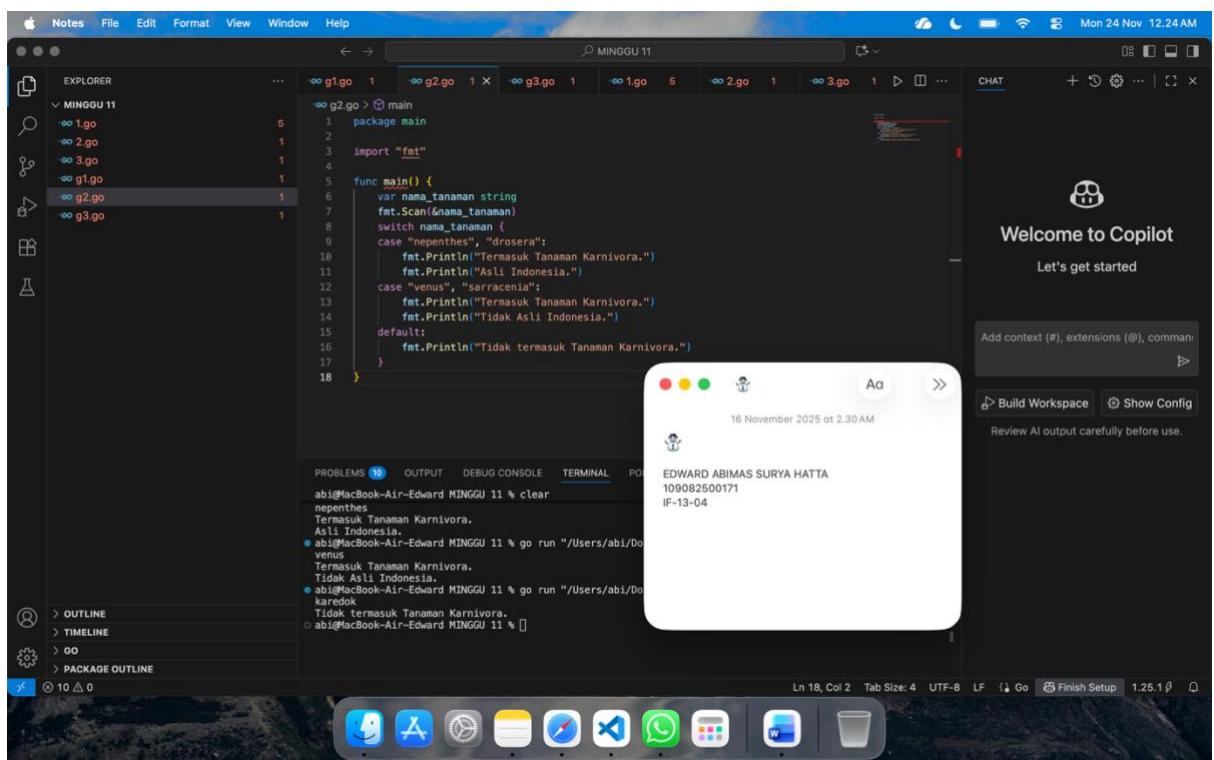
case "venus", "sarracenia":
    fmt.Println("Termasuk Tanaman Karnivora.")

    fmt.Println("Tidak Asli Indonesia.")

default:
    fmt.Println("Tidak termasuk Tanaman Karnivora.")
}
}

```

## Screenshot program



## Deskripsi program

Program ini bertujuan untuk mengidentifikasi status tanaman, apakah termasuk tanaman karnivora dan apakah tanaman tersebut asli Indonesia atau bukan, berdasarkan nama yang diinputkan. Melalui struktur *switch* yang memeriksa variabel nama tanaman, program mengelompokkan input ke dalam beberapa kategori kasus

(case). Apabila nama tanaman yang dimasukkan adalah "nepenthes" atau "drosera", program akan mencetak informasi bahwa tanaman tersebut adalah karnivora dan asli Indonesia. Sementara itu, untuk masukan "venus" atau "sarracenia", sistem akan mengidentifikasinya sebagai tanaman karnivora namun bukan asli Indonesia. Jika nama tanaman yang dimasukkan tidak terdapat dalam daftar yang telah ditentukan, blok *default* akan dieksekusi untuk memberikan informasi bahwa tanaman tersebut tidak termasuk kategori karnivora.

### 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

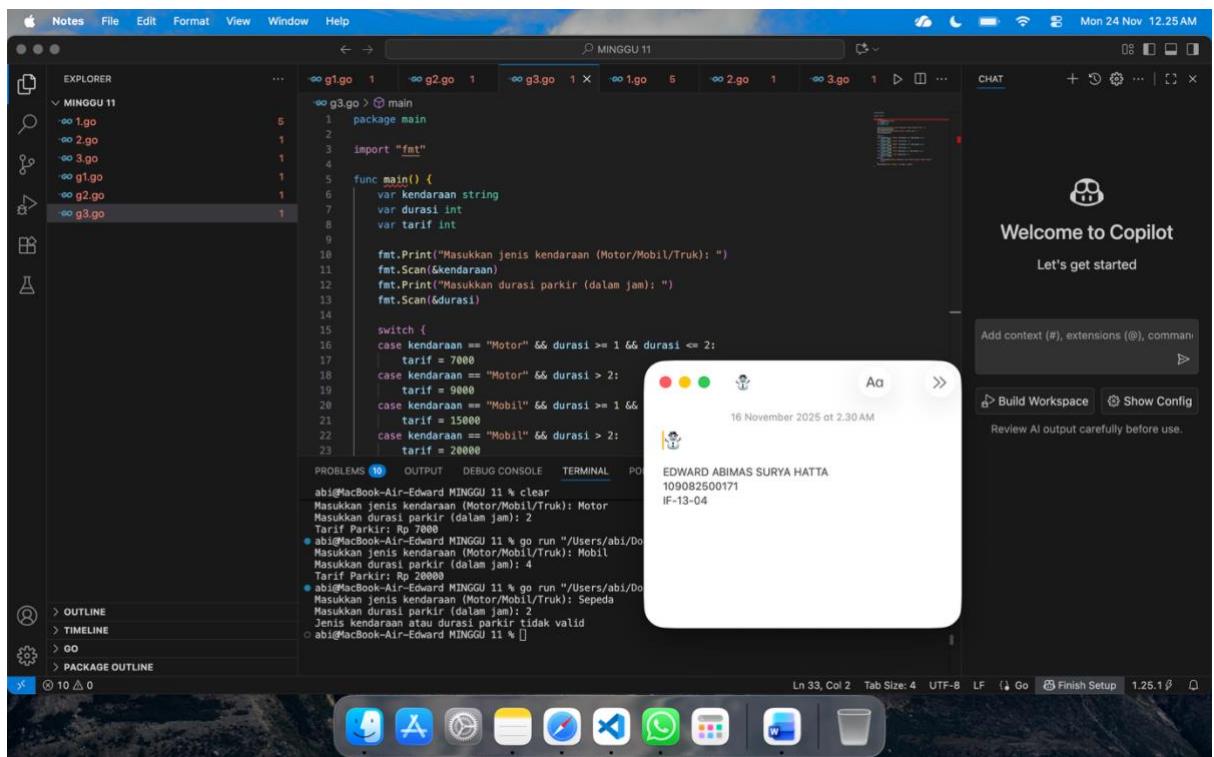
    }

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

}

```

## Screenshot program



## Deskripsi program

Program ini menangani perhitungan tarif parkir yang lebih kompleks karena melibatkan dua variabel masukan sekaligus, yaitu jenis kendaraan dan durasi parkir.

Logika program disusun menggunakan *switch* tanpa ekspresi untuk mengevaluasi kombinasi kondisi logis antara tipe kendaraan (Motor, Mobil, atau Truk) dan rentang waktu parkir secara bersamaan. Setiap *case* mendefinisikan tarif spesifik; misalnya, motor yang parkir antara 1 hingga 2 jam memiliki tarif berbeda dibandingkan jika parkir lebih dari 2 jam, dan pola perhitungan bertingkat serupa diterapkan pula untuk mobil serta truk dengan nominal harga yang lebih tinggi. Apabila input pengguna tidak memenuhi salah satu dari kombinasi aturan yang valid, mekanisme *default* akan memberikan notifikasi bahwa jenis kendaraan atau durasi parkir tidak valid, sehingga mencegah kesalahan perhitungan biaya.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

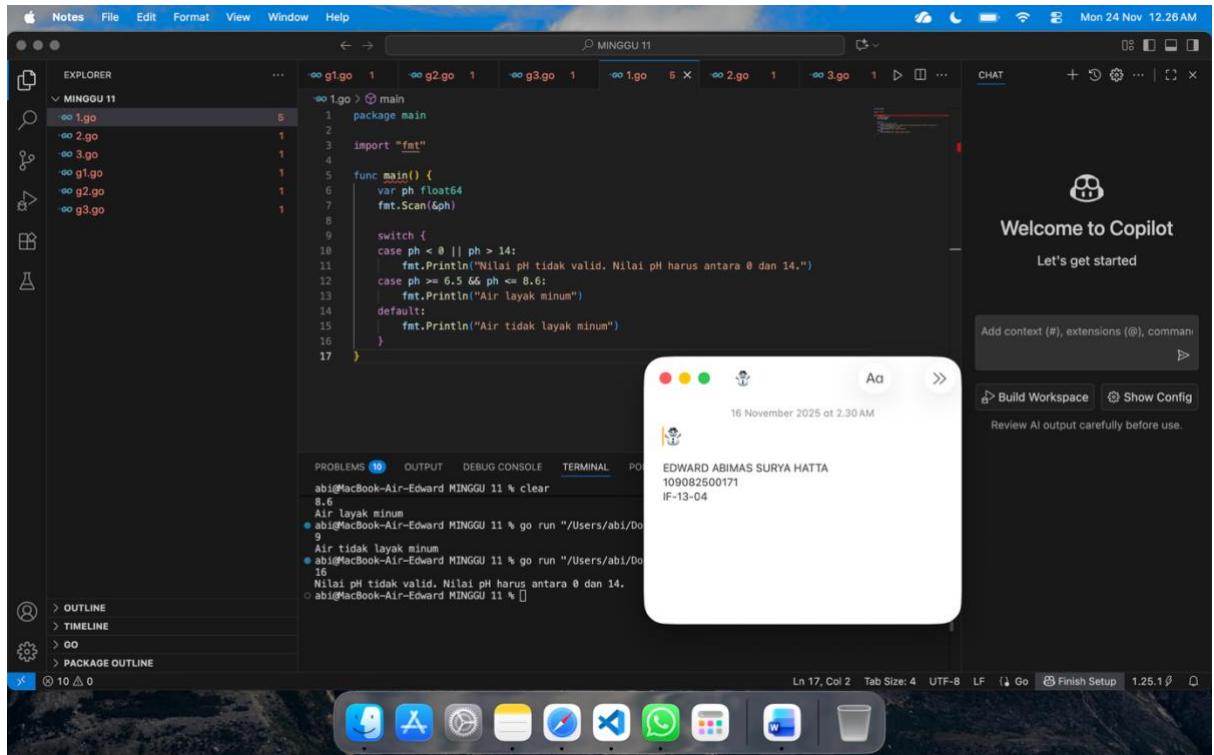
import "fmt"

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

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

```
}
```

## Screenshot program



## Deskripsi program

Program ini bertujuan untuk mengklasifikasikan kelayakan air minum berdasarkan kadar pH yang diinputkan oleh pengguna. Proses dimulai dengan mendeklarasikan variabel bertipe bilangan desimal untuk menampung nilai pH, kemudian sistem membaca masukan dari pengguna. Struktur kontrol yang digunakan adalah *switch-case* tanpa ekspresi (yang berfungsi mirip dengan *if-else* bertingkat) untuk mengevaluasi kondisi secara berurutan. Prioritas pengecekan pertama adalah memvalidasi rentang nilai input; jika nilai berada di bawah 0 atau di atas 14, program akan langsung menampilkan pesan kesalahan bahwa input tidak valid. Apabila nilai tersebut lolos validasi, program melanjutkan pemeriksaan apakah pH berada dalam rentang aman, yaitu antara 6.5 hingga 8.6 inklusif, yang akan menghasilkan keluaran bahwa air layak minum. Jika kedua kondisi spesifik tersebut tidak terpenuhi, maka secara otomatis kondisi *default* akan dieksekusi untuk menyatakan bahwa air tersebut tidak layak minum.

## 2. Tugas 2

### Source code

```
package main
```

```
import "fmt"

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

    fmt.Scan(&kendaraan, &durasi)

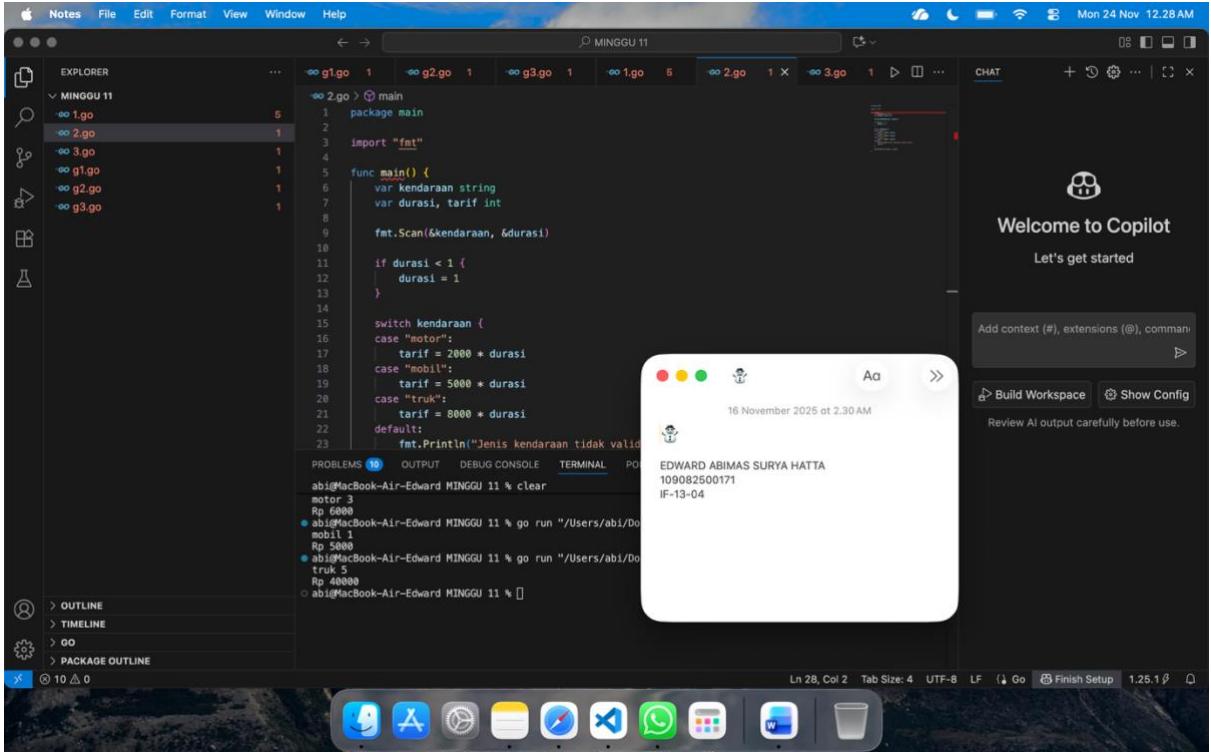
    if durasi < 1 {
        durasi = 1
    }

    switch kendaraan {
    case "motor":
        tarif = 2000 * durasi
    case "mobil":
        tarif = 5000 * durasi
    case "truk":
        tarif = 8000 * durasi
    default:
        fmt.Println("Jenis kendaraan tidak valid")
    }

    fmt.Printf("Rp %d\n", tarif)
```

```
}
```

## Screenshot program



## Deskripsi program

Pada ini, fokus utamanya adalah perhitungan tarif parkir yang dinamis bergantung pada jenis kendaraan dan durasi parkir. Program diawali dengan deklarasi variabel untuk menyimpan jenis kendaraan dalam bentuk teks dan durasi dalam satuan jam sebagai bilangan bulat. Setelah input diterima dari pengguna, terdapat logika penyesuaian durasi minimum, di mana durasi parkir yang kurang dari satu jam akan tetap dihitung sebagai satu jam penuh sesuai aturan umum. Penentuan tarif dilakukan menggunakan struktur *switch* berdasarkan jenis kendaraan yang diinputkan, baik itu motor, mobil, atau truk, dengan pengali harga per jam yang berbeda untuk setiap jenisnya. Hasil perkalian antara tarif dasar per jam dan durasi parkir kemudian ditampilkan ke layar dalam format mata uang, namun jika jenis kendaraan yang dimasukkan tidak sesuai dengan daftar yang ada, program akan memberikan notifikasi melalui blok *default* bahwa jenis kendaraan tidak valid.

### 3. Tugas 3

#### Source code

```
package main
```

```
import "fmt"

func main() {
    var input, hasil int
    fmt.Scan(&input)

    switch {
    case input%10 == 0:
        hasil = input / 10
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        fmt.Printf("Hasil pembagian antara %d / 10 =
%d\n", input, hasil)
    case input%5 == 0 && input > 5:
        hasil = input * input
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d\n",
input, hasil)
    case input%2 == 0:
        hasil = input * (input + 1)
        fmt.Println("Kategori: Bilangan Genap")
        fmt.Printf("Hasil perkalian dengan bilangan
berikutnya %d * %d = %d\n", input, input+1, hasil)
    default:
        hasil = input + (input + 1)
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n", input, input+1, hasil)
    }
}
```

## Screenshot program

```
package main
import "fmt"
func main() {
    var input, hasil int
    fmt.Scan(&input)
    switch {
    case input%10 == 0:
        hasil = input / 10
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", input, hasil)
    case input%5 == 0 && input > 5:
        hasil = input * input
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d\n", input, hasil)
    case input%2 == 0:
        hasil = input * (input + 1)
        fmt.Println("Kategori: Bilangan Genap")
        fmt.Printf("Hasil perkalian dengan bilangan berurutan %d * %d = %d\n", input, input+1, hasil)
    default:
        hasil = input + (input + 1)
    }
}
```

## Deskripsi program

Program terakhir dirancang untuk mengidentifikasi pola bilangan bulat dan melakukan operasi matematika yang spesifik sesuai dengan kategori bilangan tersebut. Setelah pengguna memasukkan sebuah bilangan bulat, program menggunakan struktur *switch* untuk memeriksa karakteristik bilangan mulai dari kondisi yang paling spesifik hingga yang umum. Pemeriksaan pertama dilakukan untuk melihat apakah bilangan tersebut merupakan kelipatan 10, di mana jika benar, program akan membaginya dengan 10. Jika bukan, pemeriksaan berlanjut untuk melihat apakah bilangan tersebut adalah kelipatan 5 (dengan nilai lebih besar dari 5), yang mana jika terpenuhi akan dilakukan operasi pengkuadratan. Apabila kedua kondisi tersebut tidak terpenuhi, program mengecek apakah bilangan tersebut genap untuk kemudian dikalikan dengan bilangan urutan berikutnya. Terakhir, jika semua kondisi sebelumnya tidak cocok, bilangan dianggap sebagai ganjil dan akan dijumlahkan dengan bilangan setelahnya sesuai instruksi pada kondisi *default*.