

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 11**

**SWITCH-CASE**



**Disusun oleh:**

**Anindya Rahadita Yumnaa**

**109082500138**

**S1IF-13-07**

**Asisten Praktikum**

Adithana dharma putra

Apri pandu wicaksono

**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 jam24 int
    fmt.Print("Masukkan jam dalam format 24 jam: ")
    fmt.Scan(&jam24)

    var period string
    var jam12 int

    if jam24 == 0 {
        jam12 = 12
        period = "AM"
    } else if jam24 == 12 {
        jam12 = 12
        period = "PM"
    } else if jam24 > 12 {
        jam12 = jam24 - 12
        period = "PM"
    } else {
        jam12 = jam24
    }
}
```

```

        period = "AM"

    }

    fmt.Printf("%d %s\n", jam12, period)

}

```

## Screenshot program

The screenshot shows a Go program named `guided 1.go` in a VS Code editor. The code reads a 24-hour time input from the user and prints it with its AM/PM period. A terminal window titled `LightClock` shows the execution of the program and its output.

```

package main

import "fmt"

func main() {
    var jam24 int
    fmt.Println("Masukkan jam dalam format 24 jam: ")
    fmt.Scan(&jam24)

    var period string
    var jam12 int

    if jam24 == 0 {
        jam12 = 12
        period = "AM"
    } else if jam24 == 12 {
        jam12 = 12
        period = "PM"
    } else if jam24 > 12 {
        jam12 = jam24 - 12
        period = "PM"
    } else {
        jam12 = jam24
        period = "AM"
    }
    fmt.Printf("%d %s\n", jam12, period)
}

```

`LightClock` terminal output:

```

109082500138
S11F-13-07
ANINDYA RAHADITA YUMNAA

```

The screenshot shows the same Go program in VS Code with a dark theme. The terminal window output is identical to the previous one.

```

package main

import "fmt"

func main() {
    var jam24 int
    fmt.Println("Masukkan jam dalam format 24 jam: ")
    fmt.Scan(&jam24)

    var period string
    var jam12 int

    if jam24 == 0 {
        jam12 = 12
        period = "AM"
    } else if jam24 == 12 {
        jam12 = 12
        period = "PM"
    } else if jam24 > 12 {
        jam12 = jam24 - 12
        period = "PM"
    } else {
        jam12 = jam24
        period = "AM"
    }
    fmt.Printf("%d %s\n", jam12, period)
}

PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/guided 1/guided 1.go"
Masukkan jam dalam format 24 jam: 13
1 PM
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/guided 1/guided 1.go"
Masukkan jam dalam format 24 jam: 0
12 AM
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/guided 1/guided 1.go"
Masukkan jam dalam format 24 jam: 12
12 PM
PS C:\Users\MSI>

```

The screenshot shows the Visual Studio Code interface. In the top left, there are tabs for 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', etc. Below the tabs, the file path is shown: C:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 1\guided 1.go. The main editor area contains the following Go code:

```

5 func main() {
15     period = "AM"
16 } else if jam24 == 12 {
17     jam12 = 12
18     period = "PM"
19 } else if jam24 > 12 {
20     jam12 = jam24 - 12
21     period = "PM"
22 } else {
23     jam12 = jam24
24     period = "AM"
25 }
26 fmt.Printf("%d %s\n", jam12, period)
27 }

```

In the bottom right corner of the editor, there is a small terminal window titled 'LightClock' showing the output of the program:

```

109082500138
S11F-13-07
ANINDYA RAHADITA YUMNA

```

Below the editor, the terminal tab is active, displaying the command PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 1\guided 1.go" followed by user input prompts and the program's output.

### Deskripsi program

Program Go ini memiliki fungsi utama untuk **mengubah format waktu dari 24 jam menjadi format 12 jam yang dilengkapi dengan keterangan AM atau PM**.

Pada awalnya, program akan meminta pengguna untuk memasukkan angka jam dalam format 24 jam (mulai dari 0 hingga 23). Setelah menerima input, program akan memproses angka tersebut menggunakan serangkaian logika kondisional.

Inti dari program ini adalah proses konversi:

- Jika jam yang dimasukkan adalah **0 (tengah malam)**, program akan mengubahnya menjadi **12 AM**.
- Jika jam adalah **12 (tengah hari)**, program akan menampilkannya sebagai **12 PM**.
- Untuk jam yang **lebih dari 12 (misalnya jam 13 hingga 23)**, program akan menguranginya dengan 12 dan memberikan keterangan **PM**. Contohnya, jam 17 akan dikonversi menjadi 5 PM.
- Sedangkan jam dari **1 hingga 11** akan tetap sama dan langsung diberikan keterangan **AM**.

## 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

```
File Edit Selection View Go Run ... ← → Search C: > Users > MSI > Documents > VSCode > MINGGU 11 > LAPRAK > guided 2 > guided 2.go > main  
1 package main  
2  
3 import "fmt"  
4  
5 func main() {  
6  
7     var nama_tanaman string  
8  
9     fmt.Scan(&nama_tanaman)  
10    switch nama_tanaman {  
11  
12        case "nepenthes", "drosera":  
13            fmt.Println("Termasuk Tanaman Karnivora.")  
14            fmt.Println("Asli Indonesia.")  
15  
16        case "venus", "sarracenia":  
17            fmt.Println("Termasuk Tanaman Karnivora.")  
18            fmt.Println("Tidak Asli Indonesia.")  
19  
20        default:  
21            fmt.Println("Tidak termasuk Tanaman Karnivora.")  
22        }  
23    }  
24  
109082500138  
S11F-13-07  
ANINDYA RAHADITA YUMNAA  
Ln 3, Col 24 | 48 characters | Plain text | 100% | Wind: UTF-8
```

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows files like "guided 1.go", "guided 2.go" (the current tab), "guided 3.go", and "soal 1.go" through "soal 3.go".
- Search Bar:** Contains the word "Search".
- Terminal:** Displays the command "go run" followed by the path to the Go file "guided 2.go". The output shows the program's logic for identifying plants based on their names ("nepenthes", "drosera", "venus", "sarracenia") and their carnivorous nature.
- Code Editor:** Shows a Go program that defines a main function with a switch statement. It checks if the input name matches "nepenthes" or "drosera", and prints "Termasuk Tanaman Karnivora." for both cases. It also handles other cases like "venus" and "sarracenia".
- Output Panel:** Shows the execution results from the terminal, including the case labels and the output string "ANINDYA RAHADITA YUMNA".

## Deskripsi program

Program Go ini berfungsi untuk **mengklasifikasikan nama-nama tanaman** yang dimasukkan oleh pengguna untuk mengetahui apakah tanaman tersebut termasuk **karnivora** dan apakah ia **berasal dari Indonesia**. Saat program dijalankan, ia akan menunggu input dari pengguna, yaitu nama tanaman. Setelah nama diterima, program akan membandingkannya dengan daftar yang sudah ditentukan menggunakan mekanisme switch. Ada dua kategori utama tanaman karnivora yang dikenali oleh program: pertama, jika nama yang dimasukkan adalah "**nepenthes**" atau "**drosera**", program akan mengidentifikasi mereka sebagai **Tanaman Karnivora** yang **Asli Indonesia**. Kedua, jika nama yang dimasukkan adalah "**venus**" (seperti Venus flytrap) atau "**sarracenia**", program akan tetap menggolongkannya sebagai **Tanaman Karnivora**.

**Karnivora**, tetapi dengan keterangan bahwa ia **Tidak Asli Indonesia**. Jika nama tanaman yang dimasukkan tidak ditemukan dalam daftar nama-nama karnivora tersebut, program akan secara otomatis menampilkan pesan bahwa tanaman tersebut **Tidak termasuk Tanaman 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")

    }

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

}

```

## Screenshot program

The screenshot shows a Go development environment with two windows. The left window is a code editor displaying a Go program named 'main'. The right window is a terminal window titled 'LightClock' showing the output of the program.

```

File Edit Selection View Go Run ... ← → Q Search
C: > Users > MSI > Documents > VSCode > MINGGU 11 > LAPRAK > guided 3 > guided 3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var kendaraan string
8     var durasi int
9     var tarif int
10
11    fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
12    fmt.Scan(&kendaraan)
13    fmt.Print("Masukkan durasi parkir (dalam jam): ")
14    fmt.Scan(&durasi)
15    switch {
16        case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
17            tarif = 7000
18        case kendaraan == "Motor" && durasi > 2:
19            tarif = 9000
20        case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
21            tarif = 15000
22        case kendaraan == "Mobil" && durasi > 2:
23            tarif = 20000
24        case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
25            tarif = 25000
26        case kendaraan == "Truk" && durasi > 2:
27            tarif = 35000
28        default:
29            fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
30    }
31
32    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
33
34
35
36
37
38
39
39

```

LightClock

109082500138  
S11F-13-07  
ANINDYA RAHADITA YUMNAA

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows files like "guided 1.go", "guided 2.go", "guided 3.go" (marked as modified), "soal 1.go", "soal 2.go", and "soal 3.go".
- Search Bar:** Contains the text "Q Search".
- Terminal:** Displays the output of running the "guided 3.go" file multiple times with different inputs (Motor, Mobil, Truk) and parking durations (7000, 20000, 25000). The terminal also shows the input prompt "Masukkan jenis kendaraan (Motor/Mobil/Truk):" and "Masukkan durasi parkir (dalam jam):".
- Code Editor:** Shows the source code for "guided 3.go". The code defines a function "main" that takes no arguments. It contains a switch statement based on the value of "tarif". If "tarif" is 35000, it prints "Jenis kendaraaan atau durasi parkir tidak valid". Otherwise, it prints "Tarif Parkir: Rp %d\n" followed by the value of "tarif".
- Bottom Status Bar:** Shows the current file is "Ln 9, Col 18", "Tab Size: 4", "UTF-8", "CRLF", and the date "12.5.19".

## Deskripsi program

Program Go ini bertugas menghitung berapa total biaya yang harus dibayar pengguna. Program akan bertanya dulu **jenis kendaraan** apa yang parkir (Motor, Mobil, atau Truk) dan **berapa lama** durasinya dalam jam.

Inti dari program ini ada pada bagaimana ia menetapkan tarif. Biaya parkir di sini tidak sama rata, program bergantung pada dua hal sekaligus: jenis kendaraan, dan apakah durasinya **pendek (1 sampai 2 jam)** atau **panjang (lebih dari 2 jam)**. Jadi, di balik layar, program membandingkan semua kemungkinan itu:

- Misalnya, jika parkir **Motor** selama **satu jam**, biayanya Rp 7.000. Tapi jika parkir **lebih dari 2 jam**, biayanya naik jadi Rp 9.000.
  - Hal yang sama berlaku untuk **Mobil** dan **Truk**, di mana masing-masing memiliki dua tarif yang berbeda untuk durasi pendek dan durasi panjang.

Jika pengguna memasukkan jenis kendaraan yang tidak dikenal oleh sistem, program akan menampilkan pesan peringatan dan tidak melanjutkan perhitungan. Setelah semua kondisi dicocokkan, program akan menemukan tarif yang tepat dan langsung mencetaknya di layar.

## **TUGAS**

## 1. Tugas 1

## Source code

```
package main

import "fmt"
```

```

func main() {

    var ph float64


    fmt.Print("Masukkan nilai pH: ")

    fmt.Scan(&ph)


    if ph < 0 || ph > 14 {

        fmt.Println("Nilai pH tidak valid. Nilai pH harus
        antara 0 dan 14.")

    } else if ph >= 6.5 && ph <= 8.6 {

        fmt.Println("Air layak minum")

    } else {

        fmt.Println("Air tidak layak minum")

    }

}

```

## Screenshot program

The screenshot shows a Visual Studio Code interface with a Go file named `soal 1.go` open. The code prompts the user for a pH value, checks if it's between 0 and 14, and prints a message based on the result. A terminal window shows the execution of the program and its output. The output includes the user input, the validation message, and the final classification of the water as drinkable or not.

```

package main

import "fmt"

func main() {
    var ph float64

    fmt.Print("Masukkan nilai pH: ")
    fmt.Scan(&ph)

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

PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 1/soal 1.go"
Masukkan nilai pH: 8.6
Air layak minum
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 1/soal 1.go"
Masukkan nilai pH: 9
Air tidak layak minum
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 1/soal 1.go"
Masukkan nilai pH: 16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.
PS C:\Users\MSI>

```

## Deskripsi program

Program Go ini dirancang untuk **mengecek kelayakan air minum** berdasarkan nilai **pH** yang dimasukkan oleh pengguna, sebuah ukuran tingkat keasaman atau kebasaan air. Pertama, program akan meminta pengguna memasukkan nilai pH. Selanjutnya, ia akan memvalidasi input tersebut, yaitu memastikan bahwa nilai pH berada dalam rentang standar ilmiah dari **0 hingga 14**, jika di luar rentang ini, program akan menampilkan pesan kesalahan. Apabila nilai pH valid, program akan menentukan kelayakan air. Air dianggap **layak minum** jika nilai pH-nya berada dalam kisaran ideal **6.5 hingga 8.6**. Di luar rentang **6.5 sampai 8.6**, air akan diklasifikasikan sebagai tidak layak minum.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var jenis string
    var durasi int

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

    fmt.Scan(&jenis)

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

    if durasi < 1 {
        durasi = 1
    }
}
```

```
var tarif int

if jenis == "motor" {
    tarif = 2000
} else if jenis == "mobil" {
    tarif = 5000
} else if jenis == "truk" {
    tarif = 8000
} else {
    fmt.Println("Jenis kendaraan tidak valid.")
    return
}

total := tarif * durasi
fmt.Printf("Total biaya parkir: Rp %d\n", total)
```

### Screenshot program

The screenshot shows the Visual Studio Code interface with a dark theme. A Go file named 'soal 2.go' is open in the editor. The code prompts the user for the type of vehicle ('motor/mobil/truk') and the duration of parking ('jam'). It then calculates the total parking fee based on the vehicle type and duration. A terminal window titled 'LightClock' is visible, showing the output of the program. The terminal output includes the timestamp '109082500138', the date 'SIIF-13-07', and the name 'ANINDYA RAHADITA YUMNA'A. The status bar at the bottom indicates the file is 1.25.1.

```
package main

import "fmt"

func main() {
    var jenis string
    var durasi int

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

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

    if durasi < 1 {
        durasi = 1
    }

    var tarif int

    if jenis == "motor" {
        tarif = 2000
    } else if jenis == "mobil" {
        tarif = 5000
    } else if jenis == "truk" {
        tarif = 8000
    } else {
        fmt.Println("Jenis kendaraan tidak valid.")
        return
    }

    total := tarif * durasi
    fmt.Printf("Total biaya parkir: Rp %d\n", total)
}
```

The screenshot shows the Visual Studio Code interface with a dark theme. A terminal window titled 'LightClock' is visible, showing the output of the program. The terminal output includes the timestamp '109082500138', the date 'SIIF-13-07', and the name 'ANINDYA RAHADITA YUMNA'A. The status bar at the bottom indicates the file is 1.25.1.

```
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 2/soal 2.go"
Masukkan jenis kendaraan (motor/mobil/truk): motor
Masukkan durasi parkir (jam): 3
Total biaya parkir: Rp 6000
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 2/soal 2.go"
Masukkan jenis kendaraan (motor/mobil/truk): mobil
Masukkan durasi parkir (jam): 1
Total biaya parkir: Rp 5000
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 2/soal 2.go"
Masukkan jenis kendaraan (motor/mobil/truk): truk
Masukkan durasi parkir (jam): 5
Total biaya parkir: Rp 40000
PS C:\Users\MSI>
```

## Deskripsi program

Program Go ini berfungsi sebagai **kalkulator sederhana untuk menghitung total biaya parkir** berdasarkan dua data input dari pengguna, **jenis kendaraan** (motor, mobil, atau truk) dan **lama durasi parkir** dalam satuan jam. Setelah menerima input, program akan segera memastikan bahwa durasi parkir minimum dihitung sebagai **satu jam**, bahkan jika pengguna memasukkan angka nol atau kurang, untuk menjamin adanya biaya parkir minimum. Selanjutnya, menggunakan struktur if-else if-else, program menetapkan tarif dasar per jam untuk kendaraan yaitu Rp 2.000 untuk motor, Rp 5.000 untuk mobil, dan Rp 8.000 untuk truk. Jika jenis kendaraan yang dimasukkan tidak sesuai, program akan menampilkan pesan kesalahan dan berhenti. Terakhir, program mengalikan tarif dasar yang sudah ditentukan dengan durasi parkir (yang

sudah dimodifikasi) untuk mendapatkan dan menampilkan **total biaya parkir** yang harus dibayar oleh pengguna.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)

    if n%10 == 0 {
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        hasil := n / 10
        fmt.Printf("Hasil pembagian antara %d / 10 =
%d\n", n, hasil)

    } else if n%5 == 0 {
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        hasil := n * n
        fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n",
n, hasil)

    } else if n%2 == 0 {
        fmt.Println("Kategori: Bilangan Genap")
        next := n + 1
        hasil := n * next
        fmt.Printf("Hasil perkalian dengan bilangan
berikutnya %d * %d = %d\n", n, next, hasil)

    } else {
        fmt.Println("Kategori: Bilangan Ganjil")
        next := n + 1
    }
}
```

```

        hasil := n + next
        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n", n, next, hasil)
    }
}

```

## Screenshot program

The screenshot shows the Visual Studio Code interface. The code editor displays a Go file named `main.go` with the following content:

```

package main
import "fmt"

func main() {
    var n int
    fmt.Println("Masukkan bilangan: ")
    fmt.Scan(&n)

    if n%10 == 0 {
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        hasil := n / 10
        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, hasil)
    } else if n%5 == 0 {
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        hasil := n * n
        fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", n, hasil)
    } else if n%2 == 0 {
        fmt.Println("Kategori: Bilangan Genap")
        next := n + 1
        hasil := n * next
        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", n, next, hasil)
    } else {
        fmt.Println("Kategori: Bilangan Ganjil")
        next := n + 1
        hasil := n + next
    }
}

```

The terminal window to the right shows the execution of the program:

```

10
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 10 * 11 = 110
110

```

The screenshot shows the Visual Studio Code interface. The code editor displays the same `main.go` file as before. The terminal window shows the execution of the program with different inputs:

```

8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
9
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^2 = 625
25
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
20

```

## Deskripsi program

Program Go ini berfungsi untuk **menganalisis sebuah angka** yang dimasukkan oleh pengguna, **mengelompokkannya** ke dalam salah satu dari empat kategori, dan kemudian **melakukan perhitungan** yang berbeda-beda untuk setiap kategori.

### **Cara Kerja dan Ketentuan Klasifikasi (Prioritas Berjenjang):**

Program ini menggunakan sistem pengecekan berurutan, artinya jika satu kondisi sudah terpenuhi, kondisi di bawahnya tidak akan dicek lagi.

#### **1. Kategori: Kelipatan 10**

- **Ketentuan:** Angka yang dimasukkan **habis dibagi 10** (contoh: 20, 30, 100).
- **Tindakan:** Program akan mencetak bahwa angka tersebut adalah "Bilangan Kelipatan 10" dan kemudian **membagi angka tersebut dengan 10**.

#### **2. Kategori: Kelipatan 5**

- **Ketentuan:** Angka **tidak habis dibagi 10**, tetapi **habis dibagi 5** (contoh: 5, 15, 25).
- **Tindakan:** Program akan mencetak "Bilangan Kelipatan 5" dan kemudian **mengkuadratkan** angka tersebut.

#### **3. Kategori: Bilangan Genap**

- **Ketentuan:** Angka **tidak termasuk** kelipatan 10 atau kelipatan 5, tetapi **habis dibagi 2** (contoh: 2, 4, 6, 8).
- **Tindakan:** Program akan mencetak "Bilangan Genap" dan kemudian **mengalikannya dengan bilangan bulat berikutnya** (misalnya, jika input 8, dihitung  $8+9$ ).

#### **4. Kategori: Bilangan Ganjil**

- **Ketentuan:** Angka **tidak memenuhi** semua kondisi di atas.
- **Tindakan:** Program akan mencetak "Bilangan Ganjil" dan kemudian **menjumlahkan** angka tersebut dengan bilangan bulat berikutnya (misalnya, jika input 7, dihitung  $7 + 8$ ).