

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>

```

```

func main() {
    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

File Edit View

109082500138
S11F-13-07
ANINDYA RAHADITA YUMNA

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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>

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 }
```

LightClock

File Edit View

109082500138
S11F-13-07
ANINDYA RAHADITA YUMNAA

Ln 3, Col 24 | 48 character | Plain text | 100% | Wind: UTF-8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 2\guided 2.go"
nepenthes
Termasuk Tanaman Karnivora.

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 "PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 11\LAPRAK\guided 2\guided 2.go"" followed by the output:

```
nepenthes
Terkait Tanaman Karnivora.
Asli Indonesia.
```
- Output Panel:** Shows the output of the Go program:

```
109082500138
S1F-13-07
ANINDYA RAHADITA YUMNA
```
- Status Bar:** Shows "Ln 24, Col 1 Tab Size: 4 UTF-8 CR LF ⌘L Go 1251 ⌘P".

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 code editor interface with a file named `main.go` open. The code implements a switch statement to calculate a parking fee based on the vehicle type ('Motor', 'Mobil', or 'Truk') and the duration of parking (1 to 2 hours). The output window shows the program's response to a sample input.

```

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)
}

```

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

- Title Bar:** File, Edit, Selection, View, Go, Run, ..., Search.
- 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.
- Editor:** Displays a Go code snippet for calculating parking fees based on vehicle type and duration.
- Terminal:** Shows the execution of the program multiple times, accepting input for vehicle type and duration, and outputting the calculated fee.
- Output:** Shows the results of the program runs.
- LightClock Editor:** A floating window showing the date and time (109082500138, S11F-13-07) and the author's name (ANINDYA RAHADITA YUMNAA).

```
func main() {
    tarif = 35000
    default:
        fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
    }
    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
}
```

```
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/guided 3/guided 3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/guided 3/guided 3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 28000
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/guided 3/guided 3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/guided 3/guided 3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Sepeda
Masukkan durasi parkir (dalam jam): 2
Jenis kendaraan atau durasi parkir tidak valid
Tarif Parkir: Rp 0
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/guided 3/guided 3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): []
```

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 window displays the following text:

```

109082500138
S1IF-13-07
ANINDYA RAHADITA YUMNAA

```

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 VS Code interface with the guided 1.go file open. The code defines a main function that prompts the user for the type of vehicle (motor, mobil, or truk) and the parking duration in hours. It then calculates the total parking fee based on a minimum rate of Rp 2,000 per hour for motorbikes, Rp 5,000 for cars, and Rp 8,000 for trucks. If the input is invalid, it prints an error message. A terminal window titled 'LightClock' is visible, showing the output of the program for a motorcycle parker for 1 hour.

```
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 VS Code interface with the guided 1.go file open. The code has been modified to include an else if block for 'mobil'. The rest of the logic remains the same, calculating the total parking fee based on the input vehicle type and duration. A terminal window titled 'LightClock' is visible, showing the output of the program for a car parker for 1 hour.

```
func main() {
    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)
}
```

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 left pane displays the Go code for 'soal 3.go'. The right pane shows a terminal window titled 'LightClock' with the following output:

```

109082500138
S1IF-13-07
ANINDYA RAHADITA YUMNA

```

The screenshot shows the Visual Studio Code interface. The left pane displays the completed Go code for 'soal 3.go'. The right pane shows a terminal window titled 'LightClock' with the following output, demonstrating the program's functionality with multiple inputs:

```

PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 3/soal 3.go"
Masukkan bilangan: 5
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 5 ^2 = 25
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 3/soal 3.go"
Masukkan bilangan: 8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 3/soal 3.go"
Masukkan bilangan: 25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^2 = 625
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 3/soal 3.go"
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 3/soal 3.go"
Masukkan bilangan: 20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 11/LAPRAK/soal 3/soal 3.go"
Masukkan bilangan: []

```

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 $7 + 8$).