

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 11
“SWITCH-CASE”



DISUSUN OLEH:
DAFFA TSAQIFNA FAUZTSANY
103112400032
S1 IF-12-01

DOSEN:
Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

DASAR TEORI:

1. Paradigma Percabangan

Percabangan dalam pemrograman memungkinkan eksekusi baris kode tertentu berdasarkan kondisi atau syarat tertentu. Salah satu struktur kontrol percabangan yang umum digunakan adalah else-if, yang memungkinkan program memilih antara beberapa aksi berdasarkan nilai kebenaran kondisi yang diberikan.

2. Karakteristik Else-If

- Else-if serupa dengan struktur kontrol if-then, tetapi memiliki aksi alternatif jika suatu kondisi bernilai false.
- Struktur penulisan terdiri dari:
 1. **Kondisi:** Syarat atau ketentuan dalam bentuk nilai boolean.
 2. **Aksi:** Instruksi yang dijalankan jika kondisi bernilai true.
 3. **Aksi Lain:** Instruksi yang dijalankan jika kondisi bernilai false.

3. Struktur Penulisan

Dalam pseudocode:

```
if kondisi_1 then
    // aksi 1
else if kondisi_2 then
    // aksi 2
else if kondisi_3 then
    // aksi 3
else
    // aksi lain
endif
```

Contoh dalam pseudocode:

```
if kondisi_1 {
    // aksi 1
} else if kondisi_2 {
    // aksi 2
} else if kondisi_3 {
    // aksi 3
} else {
    // aksi lain
}
```

4. Implementasi dengan Bahasa Go

Contoh implementasi else-if dalam Go melibatkan kasus-kasus seperti:

- Menampilkan hasil pembagian hanya jika pembagi bukan nol.
- Menentukan indeks nilai berdasarkan parameter nilai, kehadiran, dan tugas.

5. Aplikasi dan Contoh Kasus

Beberapa contoh soal yang menggunakan struktur else-if meliputi:

- **Menentukan kelayakan seseorang untuk membuat KTP berdasarkan usia dan kepemilikan kartu keluarga.**
- **Mengidentifikasi apakah suatu huruf adalah vokal, konsonan, atau bukan huruf.**
- **Menentukan apakah digit suatu bilangan empat digit terurut membesar, mengecil, atau tidak terurut.**

6. Signifikansi dan Penerapan

Else-if memungkinkan pengambilan keputusan yang lebih kompleks dalam algoritma, mendukung pengembangan program yang fleksibel dan terstruktur dengan logika bercabang.

CONTOH SOAL

Conso 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)
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-1.go'
13
1 PM
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-1.go'
0
12 AM
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-1.go'
12
12 PM
```

Deskripsi Program:

Program ini mengkonversi waktu dalam format **24 jam** ke format **12 jam AM/PM** dan mencetak hasilnya.

1. Deklarasi variabel:

```
var jam12, jam24 int
var label string
```

- **jam12**: Menyimpan hasil konversi waktu dalam format **12 jam**.
- **jam24**: Menyimpan input waktu dalam format **24 jam**.
- **label**: Menentukan apakah waktu tersebut dalam periode **AM** atau **PM**.

2. Input Waktu:

```
fmt.Scan(&jam24)
```

- Pengguna diminta untuk memasukkan waktu dalam format **24 jam** (nilai antara **0** hingga **23**).

3. Konversi Waktu:

Program menggunakan blok **switch-case** untuk menentukan nilai **jam12** dan **label (AM/PM)** berdasarkan kondisi:

```
switch {
case jam24 == 0:
    jam12 = 12
    label = "AM"
```

- Jika **jam24 = 0** (tengah malam), maka **jam12 = 12** dan label menjadi **AM**.

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

- Jika **jam24 < 12** (antara pukul 1 pagi hingga 11 pagi), maka **jam12 = jam24** dan label menjadi **AM**.

```
case jam24 == 12:
    jam12 = 12
    label = "PM"
```

- Jika **jam24 = 12** (tengah hari), maka **jam12 = 12** dan label menjadi **PM**.

```
case jam24 > 12:
    jam12 = jam24 - 12
    label = "PM"
```

- Jika **jam24 > 12** (antara pukul 13 hingga 23), maka **jam12 = jam24 - 12** dan label menjadi **PM**.

4. Output Hasil:

```
fmt.Println(jam12, label)
```

- Program mencetak waktu dalam format **12 jam** disertai label **AM** atau **PM**.

Contoh: Jika pengguna memasukkan

```
0 (input)
```

```
12 AM (output)
```

2.cosol2

```
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.")
    }
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-2.go'
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-2.go'
venus
Termasuk Tanaman Karnivora.
Tidak Asli Indonesia.
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-2.go'
keredok
Tidak termasuk Tanaman Karnivora.
```

Deskripsi Program:

Program ini menentukan apakah nama tanaman yang diinputkan termasuk tanaman karnivora, serta asal-usulnya (asli Indonesia atau tidak).

1. Deklarasi variabel:

```
var nama_tanaman string
```

- **nama_tanaman:** Menyimpan nama tanaman yang diinputkan oleh pengguna.

2. Input Nama Tanaman:

```
fmt.Scan(&nama_tanaman)
```

- Program meminta pengguna memasukkan nama tanaman.

3. Pengelompokan Tanaman dengan switch-case:

Blok **switch-case** digunakan untuk memeriksa nama tanaman yang diinputkan:

```
case "nepenthes", "drosera":  
    fmt.Println("Termasuk Tanaman Karnivora.")  
    fmt.Println("Asli Indonesia.")
```

- Jika tanaman adalah "**nepenthes**" atau "**drosera**", maka tanaman termasuk **karnivora** dan **asli Indonesia**.

```
case "venus", "sarracenia":  
    fmt.Println("Termasuk Tanaman Karnivora.")  
    fmt.Println("Tidak Asli Indonesia.")
```

- Jika tanaman adalah "**venus**" atau "**sarracenia**", maka tanaman termasuk **karnivora** tetapi **tidak asli Indonesia**.

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

- Jika nama tanaman tidak cocok dengan kasus di atas, maka tanaman dianggap **bukan tanaman karnivora**.

```
case jam24 > 12:  
    jam12 = jam24 - 12  
    label = "PM"
```

- Jika **jam24 > 12** (antara pukul 13 hingga 23), maka **jam12 = jam24 - 12** dan label menjadi **PM**.

Contoh: Jika pengguna memasukkan

```
nepenthes (input)  
Termasuk Tanaman Karnivora. (output)  
Asli Indonesia. (output)
```

3.cosol3

```
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)
}
```

Output:


```

PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-3.go'
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-3.go'
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-3.go'
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir: Rp 9000
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-3.go'
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\conso8\conso8-3.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

```

Deskripsi Program:

Program ini menghitung tarif parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna.

1. Deklarasi variabel:

```

var kendaraan string
var durasi int
var tarif int

```

- **kendaraan:** Menyimpan jenis kendaraan yang diinputkan (Motor, Mobil, atau Truk).
- **durasi:** Menyimpan lama waktu parkir (dalam jam).
- **tarif:** Menyimpan tarif parkir berdasarkan jenis kendaraan dan durasi parkir.

2. Input Jenis Kendaraan dan Durasi Parkir:

```

fmt.Scan(&kendaraan)
fmt.Scan(&durasi)

```

- Program meminta pengguna untuk memasukkan jenis kendaraan dan durasi parkir.

3. Logika switch-case:

Blok **switch-case** menentukan tarif parkir berdasarkan:

- **Jenis kendaraan** (Motor, Mobil, Truk).
- **Durasi parkir** (1–2 jam atau lebih dari 2 jam).

Kasus untuk Motor

```
case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:  
    tarif = 7000  
case kendaraan == "Motor" && durasi > 2:  
    tarif = 9000
```

- Motor parkir selama 1–2 jam: **Rp 7000.**
- Motor parkir lebih dari 2 jam: **Rp 9000.**

Kasus untuk Mobil

```
case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:  
    tarif = 15000  
case kendaraan == "Mobil" && durasi > 2:  
    tarif = 20000
```

- Mobil parkir selama 1–2 jam: **Rp 15000.**
- Mobil parkir lebih dari 2 jam: **Rp 20000.**

Kasus untuk Truk

```
case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:  
    tarif = 25000  
case kendaraan == "Truk" && durasi > 2:  
    tarif = 35000
```

- Truk parkir selama 1–2 jam: **Rp 25000.**
- Truk parkir lebih dari 2 jam: **Rp 35000.**

Kasus Default

```
default:  
    fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
```

- Jika jenis kendaraan atau durasi tidak valid, program menampilkan pesan error.

4. Output Tarif Parkir:

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

- Program mencetak tarif parkir berdasarkan hasil perhitungan.

Contoh: Jika pengguna memasukkan

```
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor (input)  
Masukkan durasi parkir (dalam jam): 3 (input)  
Tarif Parkir: Rp 9000 (output)
```

SOAL LATIHAN

1. Latihan1

Source Code:

```
package main

import "fmt"

func main() {
    var x float64
    fmt.Scan(&x)
    switch {
    case x >= 0 && x <= 14:
        switch {
        case x >= 6.5 && x <= 8.6:
            fmt.Print("Air layak minum")
        case x < 6.5 || x > 8.6:
            fmt.Print("Air tidak layak minum ")
        }
    default:
        fmt.Print("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
    }
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-1.go'
8.6
Air layak minum
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-1.go'
9
Air tidak layak minum
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-1.go'
16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.
```

Deskripsi Program:

Program ini digunakan untuk menentukan kelayakan air untuk diminum berdasarkan nilai pH yang dimasukkan oleh pengguna. Nilai pH yang valid berada dalam rentang 0 hingga 14, dan air dianggap layak minum jika nilai pH-nya berada di antara 6.5 dan 8.6.

1. Deklarasi variabel:

```
var x float64
```

- **x:** Menyimpan nilai pH air yang diinputkan oleh pengguna.

2. Input Nilai pH:

```
fmt.Scan(&x)
```

- Program meminta pengguna untuk memasukkan nilai pH.

3. Logika Utama dengan switch-case:

Logika utama menggunakan dua tingkat **switch**, yaitu:

- **Tingkat pertama:** Memastikan nilai pH berada di antara 0 dan 14.
- **Tingkat kedua:** Memeriksa kelayakan pH air untuk diminum.

Logika Tingkat Pertama

```
switch {
case x >= 0 && x <= 14:
    // Melanjutkan pemeriksaan ke tingkat kedua
default:
    fmt.Print("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
```

- Jika nilai pH berada di rentang **0 hingga 14**, program melanjutkan pemeriksaan ke tingkat kedua.
- Jika nilai pH berada di luar rentang tersebut, program mencetak pesan kesalahan.

Logika Tingkat Kedua

```
switch {
case x >= 6.5 && x <= 8.6:
    fmt.Print("Air layak minum")
case x < 6.5 || x > 8.6:
    fmt.Print("Air tidak layak minum ")
}
```

- **pH antara 6.5 dan 8.6:** Air dianggap layak minum.
- **pH di luar 6.5 hingga 8.6:** Air dianggap tidak layak minum.

Contoh: Jika pengguna memasukkan

7.5 (input)

Air layak minum (output)

2. Latihan2

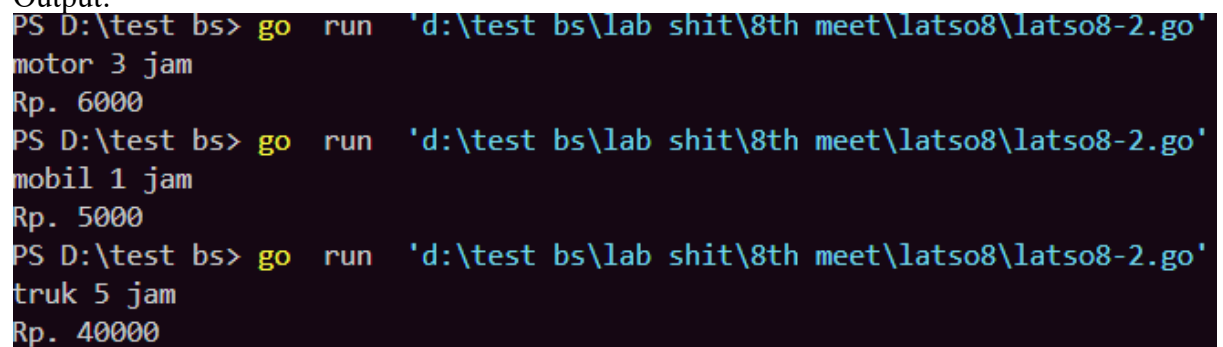
Source Code:

```
package main

import "fmt"

func main() {
    var x int
    var y string
    fmt.Scan(&y)
    switch {
    case y == "motor" || y == "Motor" || y == "MOTOR":
        fmt.Scan(&x)
        x = x * 2000
        fmt.Print("Rp. ", x)
    case y == "mobil" || y == "Mobil" || y == "MOBIL":
        fmt.Scan(&x)
        x = x * 5000
        fmt.Print("Rp. ", x)
    case y == "truk" || y == "Truk" || y == "TRUK":
        fmt.Scan(&x)
        x = x * 8000
        fmt.Print("Rp. ", x)
    default:
        fmt.Print("invalid input")
    }
}
```

Output:



```
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-2.go'
motor 3 jam
Rp. 6000
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-2.go'
mobil 1 jam
Rp. 5000
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-2.go'
truk 5 jam
Rp. 40000
```

Deskripsi Program:

Program ini digunakan untuk menghitung biaya parkir berdasarkan jenis kendaraan dan durasi parkir (dalam jam). Program mendukung tiga jenis kendaraan: **motor**, **mobil**, dan **truk**, dengan tarif per jam berbeda. Jika jenis kendaraan yang dimasukkan tidak valid, program akan mencetak pesan kesalahan.

1. Deklarasi variabel:

```
var x int
var y string
```

- **x**: Menyimpan durasi parkir dalam jam.
- **y**: Menyimpan jenis kendaraan yang dimasukkan pengguna.

2. Input Jenis Kendaraan:

```
fmt.Scan(&y)
```

- Program meminta pengguna untuk memasukkan jenis kendaraan dalam bentuk string (contoh: *motor*, *mobil*, *truk*).

3. Struktur switch-case:

Program menggunakan **switch** untuk menentukan tindakan berdasarkan nilai dari **y**.

Kasus Jenis Kendaraan Motor

```
case y == "motor" || y == "Motor" || y == "MOTOR":
    fmt.Scan(&x)
    x = x * 2000
    fmt.Print("Rp. ", x)
```

Jika pengguna memasukkan *motor* (dengan berbagai kapitalisasi), program:

- Meminta durasi parkir (**x**) dalam jam.
- Menghitung tarif parkir dengan rumus: **tarif = x * 2000**.
- Mencetak hasil biaya parkir.

Kasus Jenis Kendaraan Mobil

```
case y == "mobil" || y == "Mobil" || y == "MOBIL":
    fmt.Scan(&x)
    x = x * 5000
    fmt.Print("Rp. ", x)
```

- Jika pengguna memasukkan *mobil* (dengan berbagai kapitalisasi), program:
 - Meminta durasi parkir (**x**) dalam jam.
 - Menghitung tarif parkir dengan rumus: **tarif = x * 5000**.
 - Mencetak hasil biaya parkir.

Kasus untuk Truk

```
case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
    tarif = 25000
case kendaraan == "Truk" && durasi > 2:
    tarif = 35000
```

- Truk parkir selama 1–2 jam: **Rp 25000**.
- Truk parkir lebih dari 2 jam: **Rp 35000**.

Kasus Default

```
default:  
    fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
```

- Jika jenis kendaraan atau durasi tidak valid, program menampilkan pesan error.

4. Output Tarif Parkir:

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

- Program mencetak tarif parkir berdasarkan hasil perhitungan.

Contoh: Jika pengguna memasukkan

```
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor (input)  
Masukkan durasi parkir (dalam jam): 3 (input)  
Tarif Parkir: Rp 9000 (output)
```

3. Latihan3

Source Code:

```
package main

import "fmt"

func main() {
    var x int
    fmt.Scan(&x)
    switch {
    case x%10 == 0:
        fmt.Print("Kategori: Bilangan Kelipatan 10\nHasil pembagian antara", x, " / 10 = ", x/10)
    default:
        switch {
        case x%5 == 0 && x != 5:
            fmt.Print("Kategori: Bilangan Kelipatan 5 \nHasil kuadrat dari ", x, "^2 = ", x*x)
        default:
            switch {
            case x%2 == 0:
                fmt.Print("Kategori: Bilangan Genap \nHasil perkalian dengan bilangan berikutnya ", x, " * ", x+1, " = ", x*(x+1))
            case x%2 != 0:
                fmt.Print("Kategori: Bilangan Ganjil \nHasil penjumlahan dengan bilangan berikutnya ", x, " + ", x+1, " = ", x+(x+1))
            default:
                fmt.Print("invalid")
            }
        }
    }
}
```

Output:


```

PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-3.go'
5
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 5 + 6 = 11
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-3.go'
8
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-3.go'
25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25^2 = 625
PS D:\test bs> go run 'd:\test bs\lab shit\8th meet\latso8\latso8-3.go'
20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2

```

Deskripsi Program:

Program ini membaca sebuah bilangan bulat (x) yang dimasukkan oleh pengguna dan mengkategorikan bilangan tersebut berdasarkan kondisi berikut:

1. **Bilangan Kelipatan 10**
2. **Bilangan Kelipatan 5 (bukan 10)**
3. **Bilangan Genap (selain kelipatan 10 atau 5)**
4. **Bilangan Ganjil (selain kelipatan 10 atau 5)**

Setelah menentukan kategori, program juga menampilkan hasil perhitungan yang sesuai dengan kategori bilangan tersebut.

1. Deklarasi variabel:

```
var x int
```

- **x**: Menyimpan bilangan bulat yang dimasukkan oleh pengguna.

2. Input dari Pengguna:

```
fmt.Scan(&x)
```

- Membaca bilangan **x** dari input.
3. Penggunaan Struktur switch Bersarang:
 - Program menggunakan struktur **switch-case** untuk memeriksa kondisi bilangan.
 - Kondisi diperiksa dalam urutan tertentu: **kelipatan 10**, **kelipatan 5**, **bilangan genap**, dan **bilangan ganjil**.

Bilangan Kelipatan 10

```
case x%10 == 0:
```

```
fmt.Print("Kategori: Bilangan Kelipatan 10\nHasil pembagian antara", x, " / 10 = ", x/10)
```

- Jika $x \% 10 == 0$ (habis dibagi 10), bilangan dikategorikan sebagai *Kelipatan 10*.
- Program menghitung hasil pembagian $x / 10$ dan menampilkan hasilnya.

Bilangan Kelipatan 5 (Bukan Kelipatan 10)

```
case x%5 == 0 && x != 5:  
    fmt.Print("Kategori: Bilangan Kelipatan 5 \nHasil kuadrat dari ", x, "^2 = ", x*x)
```

- Jika $x \% 5 == 0$ dan x bukan kelipatan 10 ($x \neq 10$), bilangan dikategorikan sebagai *Kelipatan 5*.
- Program menghitung kuadrat x^2 dan menampilkan hasilnya.

Bilangan Genap

```
case x%2 == 0:  
    fmt.Print("Kategori: Bilangan Genap \nHasil perkalian dengan bilangan berikutnya", x, " * ", x+1, " = ", x*(x+1))
```

- Jika $x \% 2 == 0$ (habis dibagi 2) tetapi tidak memenuhi syarat sebelumnya, bilangan dikategorikan sebagai *Genap*.
- Program menghitung perkalian $x * (x+1)$ (dengan bilangan berikutnya) dan menampilkan hasilnya.

Bilangan Ganjil

```
case x%2 != 0:  
    fmt.Print("Kategori: Bilangan Ganjil \nHasil penjumlahan dengan bilangan berikutnya", x, " + ", x+1, " = ", x+(x+1))
```

- Jika $x \% 2 \neq 0$ (tidak habis dibagi 2), bilangan dikategorikan sebagai *Ganjil*.
- Program menghitung penjumlahan $x + (x+1)$ (dengan bilangan berikutnya) dan menampilkan hasilnya.

Kasus Default

```
default:  
    fmt.Print("invalid")
```

- Kondisi `default` hanya akan digunakan jika bilangan tidak memenuhi salah satu kategori sebelumnya, meskipun kasus ini jarang terjadi karena semua bilangan bulat pasti memenuhi salah satu kategori.

Contoh: Jika pengguna memasukkan

20 (input)

Kategori: Bilangan Kelipatan 10 (output)

Hasil pembagian antara 20 / 10 = 2 (output)