LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11

SWITCH-CASE

Disusun oleh:

Andra Dwicki Saputra

109082500206

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

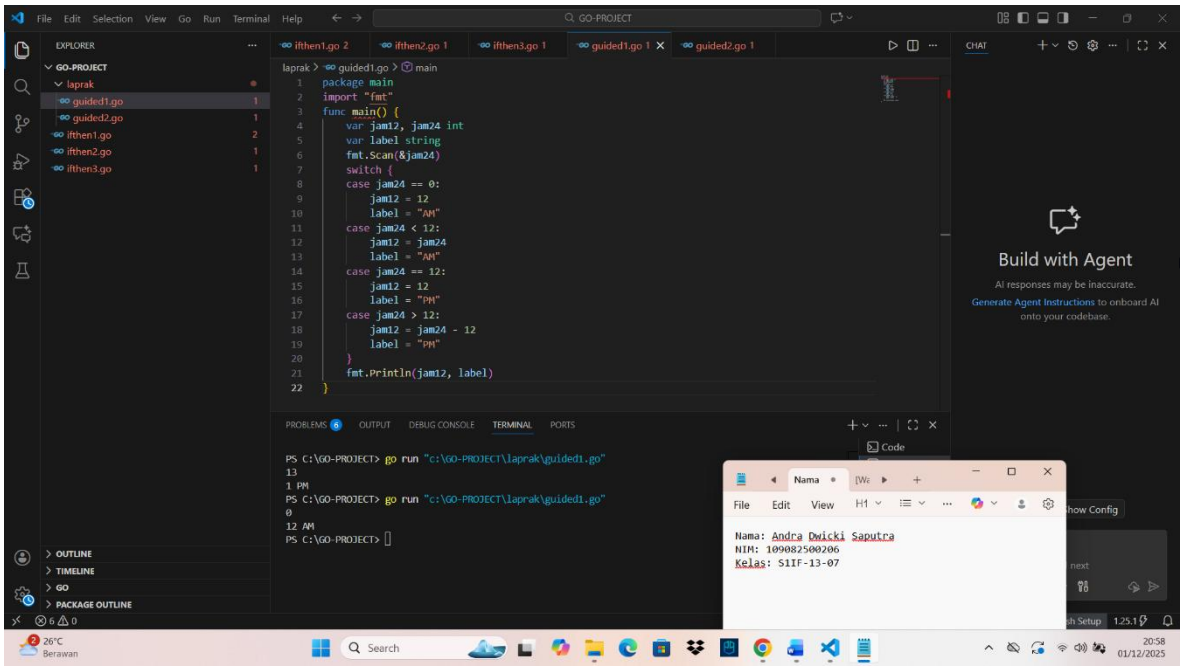
LATIHAN KELAS – GUIDED

1. **Guided 1**

Source Code

```
package main

import "fmt"

func main() {

    var 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)

}
```

Screenshoot program



Deskripsi program

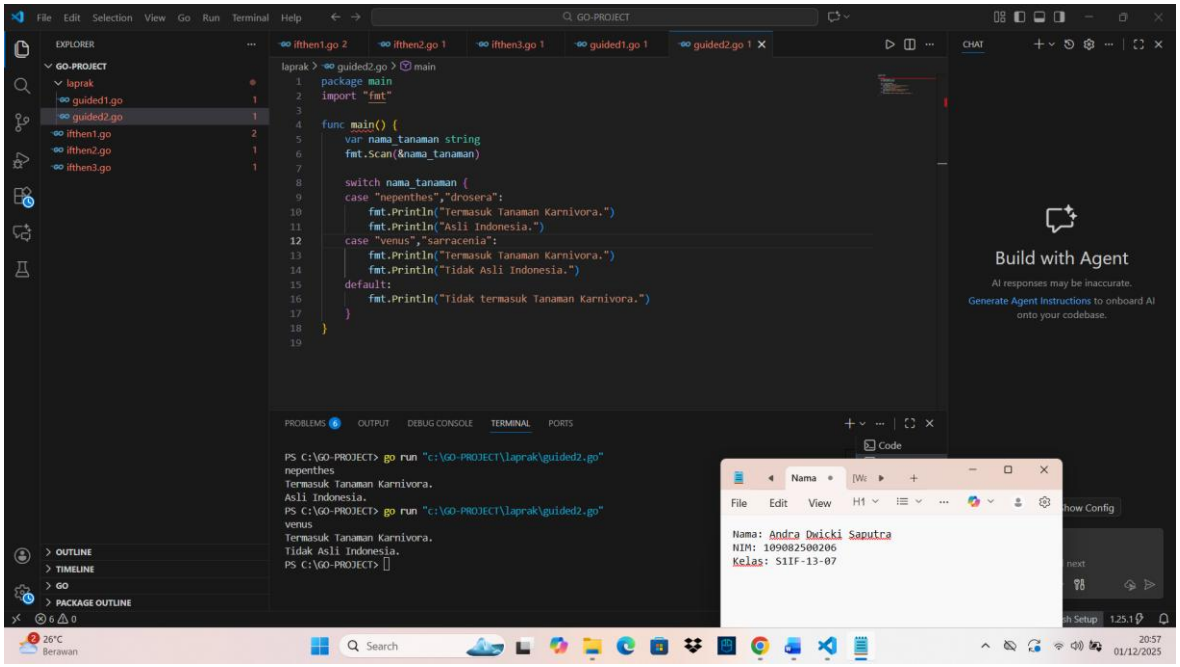
Program tersebut berfungsi untuk mengubah jam dalam format 24 jam menjadi format 12 jam. Program menerima input angka jam (0–23) melalui fmt.Scan. Kemudian, program menggunakan struktur switch untuk menentukan nilai jam dalam format 12 jam dan label waktu AM atau PM. Jika input bernilai 0, maka jam diubah menjadi 12 AM sebagai penanda tengah malam. Jika kurang dari 12, jam tetap sama dengan label AM. Jika bernilai 12, jam menjadi 12 PM sebagai penanda tengah hari. Sedangkan jika lebih dari 12, jam dikurangi 12 dan diberi label PM. Di akhir, program menampilkan hasil konversi jam dalam format “jam label”, misalnya “2 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.")
    }
}
```

Screenshoot program



Deskripsi program

Program ini fokus utamanya adalah mengidentifikasi jenis dan asal usul tanaman berdasarkan input nama yang diberikan oleh kita. Di dalam fungsi main, program mendeklarasikan variabel nama_tanaman bertipe *string*, kemudian menggunakan fungsi fmt.Scan untuk menangkap input teks dari terminal dan menyimpannya ke dalam variabel tersebut.

Logika pengelompokan diimplementasikan menggunakan struktur kontrol switch yang memeriksa nilai variabel nama_tanaman. Keunikan kode ini terletak pada penggunaan multiple values in case, di mana satu blok case dapat menangani beberapa kondisi string sekaligus (seperti "nepenthes" dan "drosera" dalam satu grup). Program membagi output menjadi tiga kemungkinan: tanaman karnivora asli Indonesia, tanaman karnivora non-lokal (seperti "venus" atau "sarracenia"), dan blok default untuk menangani input tanaman lain yang tidak terdaftar sebagai karnivora.

3. Guided 3

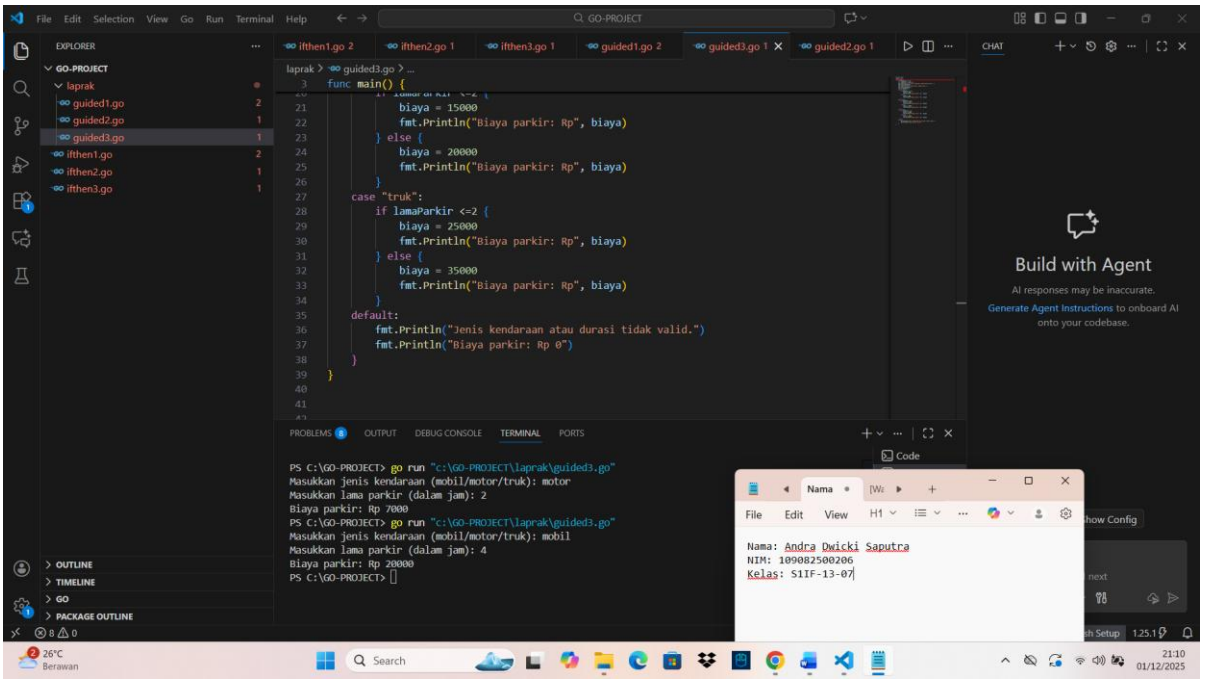
Source Code

```
package main


import "fmt"

func main() {

    var kendaraan string

    var lamaParkir, biaya int
```

```
        fmt.Print("Masukkan jenis kendaraan
    (mobil/motor/truk): ")
        fmt.Scan(&kendaraan)
        fmt.Print("Masukkan lama parkir (dalam jam): ")
        fmt.Scan(&lamaParkir)
        switch kendaraan {
        case "motor":
            if lamaParkir <=2 {
                biaya = 7000
                fmt.Println("Biaya parkir: Rp", biaya)
            } else {
                biaya = 9000
                fmt.Println("Biaya parkir: Rp", biaya)
            }
        case "mobil":
            if lamaParkir <=2 {
                biaya = 15000
                fmt.Println("Biaya parkir: Rp", biaya)
            } else {
                biaya = 20000
                fmt.Println("Biaya parkir: Rp", biaya)
            }
        case "truk":
            if lamaParkir <=2 {
                biaya = 25000
                fmt.Println("Biaya parkir: Rp", biaya)
```

```
            } else {
                biaya = 35000
                fmt.Println("Biaya parkir: Rp", biaya)
            }
        default:
            fmt.Println("Jenis kendaraan atau durasi tidak
    valid.")
            fmt.Println("Biaya parkir: Rp 0")
        }
    }
```

Screenshoot program



Deskripsi program

Program ini meminta kita untuk memasukkan dua data utama: jenis kendaraan (sebagai *string*) dan lama durasi parkir dalam satuan jam (sebagai *integer*). Variabel biaya disiapkan untuk menampung nominal harga yang harus dibayarkan berdasarkan aturan yang berlaku.

Logika perhitungan tarif menggunakan struktur kontrol bertingkat (*nested control structures*). Di bagian luar, program menggunakan switch untuk menyeleksi kategori kendaraan (motor, mobil, atau truk). Kemudian, di dalam setiap blok case, terdapat percabangan if-else untuk membedakan tarif berdasarkan durasi: tarif dasar untuk parkir <= 2 jam, dan tarif berbeda (lebih mahal) jika parkir lebih dari 2 jam. Penting dicatat bahwa skema tarif di sini bersifat *flat* (harga tetap) untuk durasi di atas 2 jam, bukan dihitung secara akumulatif per jam.

TUGAS

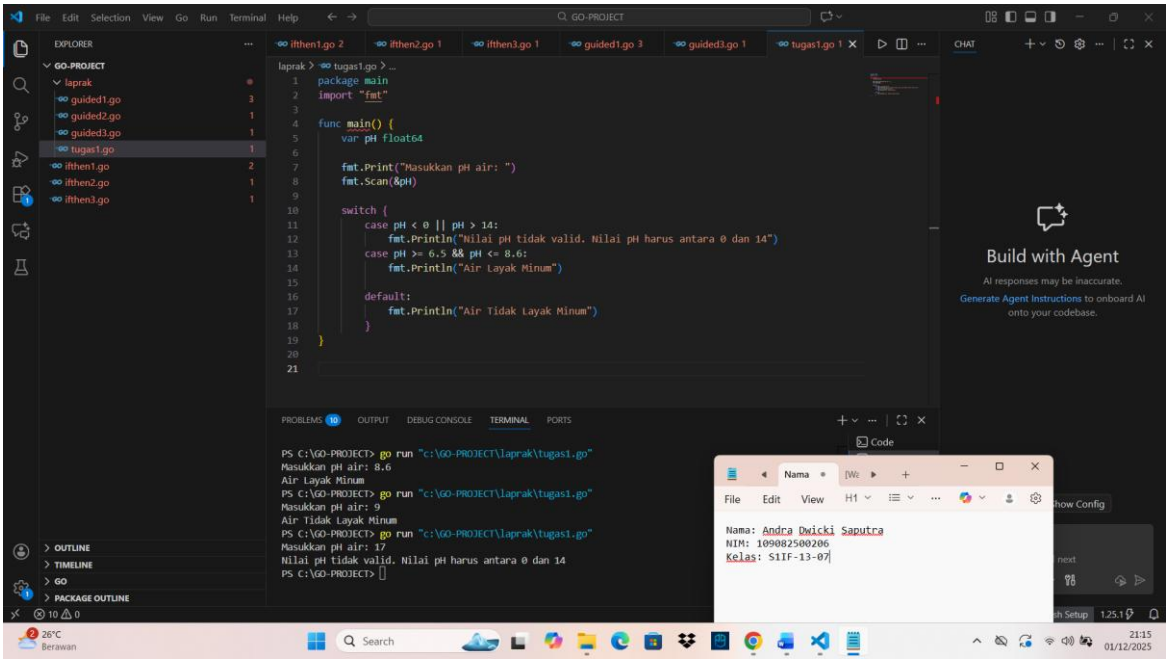
1. **Tugas 1**

   Source code

```
package main


import "fmt"


func main() {
    var pH float64


    fmt.Print("Masukkan pH air: ")
    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")

            }

        }
```

Screenshoot program



Deskripsi program

Program ini fungsinya adalah mengevaluasi kelayakan konsumsi air berdasarkan input tingkat keasaman (pH). Variabel pH dideklarasikan dengan tipe data float64 agar mampu menerima angka desimal yang presisi (seperti 6.5), di mana inputnya diambil dari kita menggunakan fungsi fmt.Scan.

Logika pengambilan keputusan menggunakan struktur switch tanpa ekspresi (*tagless switch*) yang fleksibel. Blok pertama (case) bertugas memvalidasi angka menggunakan operator logika OR (||) untuk menolak nilai yang tidak masuk akal (di bawah 0 atau di atas 14). Blok kedua menggunakan operator AND (&&) untuk menetapkan rentang aman (antara 6.5 sampai 8.6) sebagai "Air Layak Minum". Jika input valid namun tidak masuk dalam rentang aman tersebut (misalnya terlalu asam atau terlalu basa), program akan menjalankan blok default dan menyatakan air tidak layak minum.

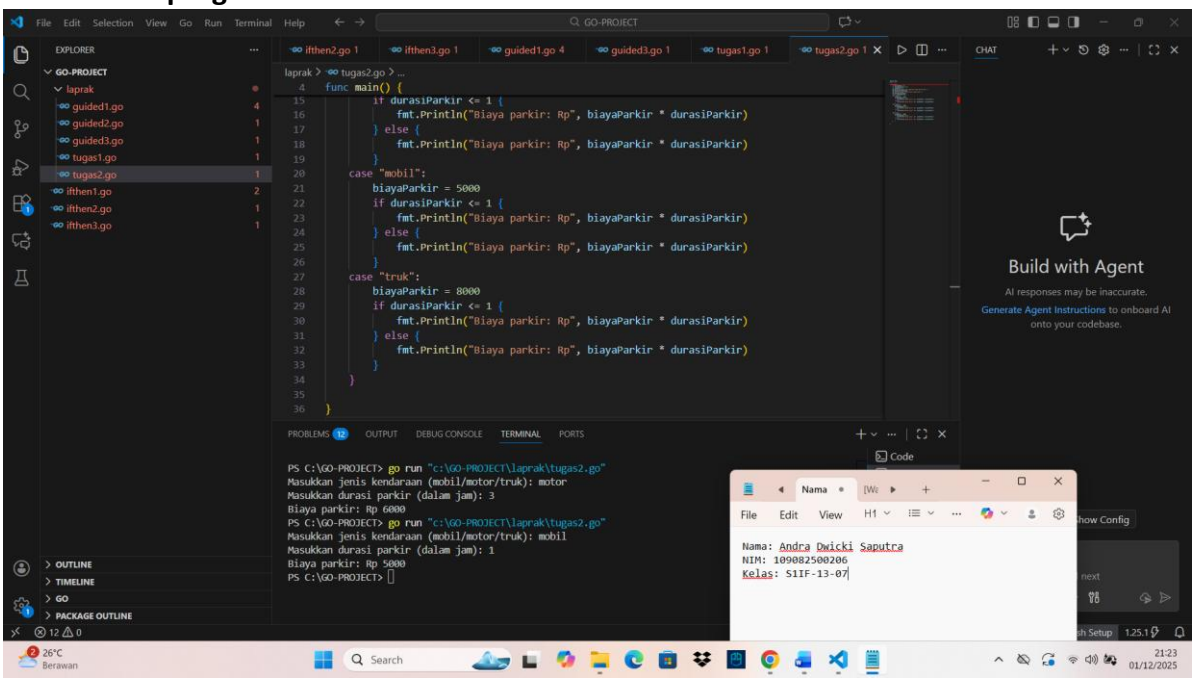
2. Tugas 2

Source code

```
package main

import "fmt"


func main() {
    var durasiParkir int
    var jenisKendaraan string
    var biayaParkir int
    fmt.Print("Masukkan jenis kendaraan
(mobil/motor/truk): ")
    fmt.Scan(&jenisKendaraan)
    fmt.Print("Masukkan durasi parkir (dalam jam): ")
    fmt.Scan(&durasiParkir)
    switch jenisKendaraan {
    case "motor":
        biayaParkir = 2000
        if durasiParkir <= 1 {
            fmt.Println("Biaya parkir: Rp", biayaParkir
* durasiParkir)
        } else {
            fmt.Println("Biaya parkir: Rp", biayaParkir
* durasiParkir)
        }
    case "mobil":
        biayaParkir = 5000
        if durasiParkir <= 1 {
            fmt.Println("Biaya parkir: Rp", biayaParkir
* durasiParkir)
```

```
            } else {
                fmt.Println("Biaya parkir: Rp", biayaParkir
    * durasiParkir)
            }
        case "truk":
            biayaParkir = 8000
            if durasiParkir <= 1 {
                fmt.Println("Biaya parkir: Rp", biayaParkir
    * durasiParkir)
            } else {
                fmt.Println("Biaya parkir: Rp", biayaParkir
    * durasiParkir)
            }
        }
    }
```

Screenshoot program



Deskripsi program

Program ini merupakan kalkulator biaya parkir berbasis tarif per jam, program meminta input dari kita berupa jenis kendaraan (string) dan durasi parkir dalam jam (integer). Variabel biayaParkir digunakan untuk menyimpan harga dasar per jam yang berbeda-beda tergantung jenis kendaraannya.

Logika program menggunakan struktur switch untuk memisahkan kategori kendaraan. Untuk "motor" tarifnya 2.000, "mobil" 5.000, dan "truk" 8.000. Di dalam setiap blok case, terdapat percabangan if-else yang mengecek apakah durasi parkir kurang dari atau sama dengan 1 jam. Namun, jika diperhatikan dengan saksama, rumus perhitungan di dalam blok if maupun else adalah identik (biayaParkir * durasiParkir). Artinya, program ini menghitung biaya secara linier (tarif dikali jam) tanpa membedakan apakah parkir itu 1 jam pertama atau jam-jam berikutnya.

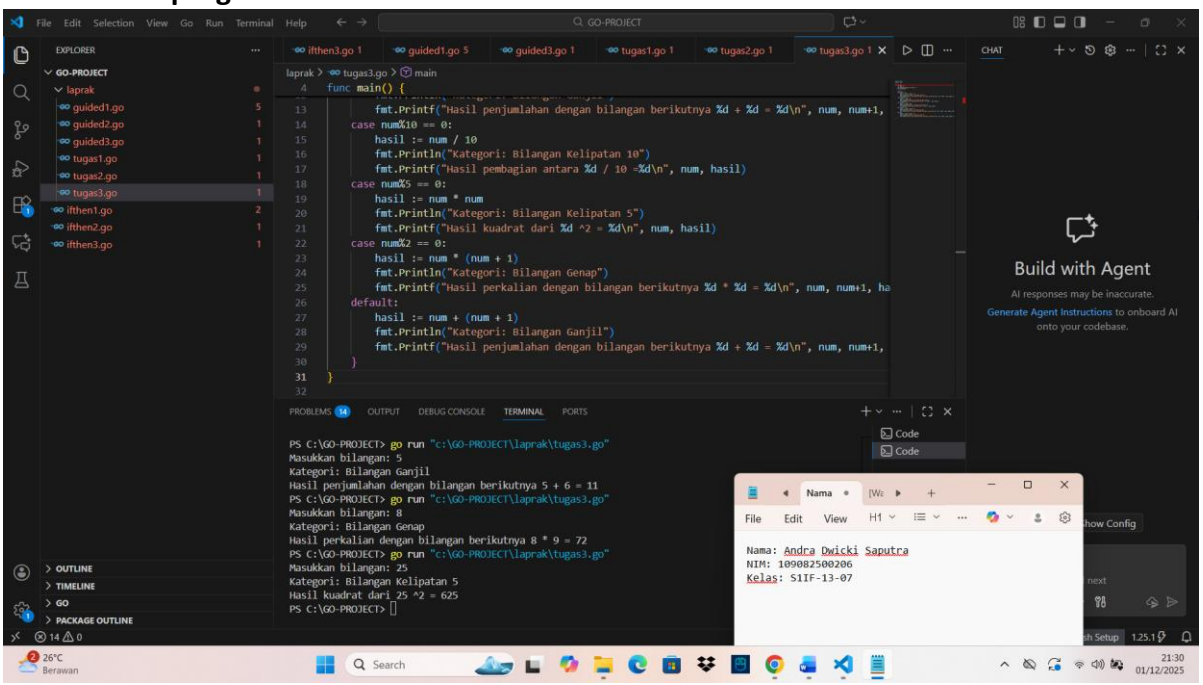
3. Tugas 3

Source code

```
package main

import "fmt"

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

    switch {
    case num == 5:
        hasil := num + (num + 1)
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan
berikutnya %d + %d = %d\n", num, num+1, hasil)
    case num%10 == 0:
        hasil := num / 10
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        fmt.Printf("Hasil pembagian antara %d / 10 =
%d\n", num, hasil)
    case num%5 == 0:
```

```
            hasil := num * num
            fmt.Println("Kategori: Bilangan Kelipatan 5")
            fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n",
    num, hasil)
        case num%2 == 0:
            hasil := num * (num + 1)
            fmt.Println("Kategori: Bilangan Genap")
            fmt.Printf("Hasil perkalian dengan bilangan
    berikutnya %d * %d = %d\n", num, num+1, hasil)
        default:
            hasil := num + (num + 1)
            fmt.Println("Kategori: Bilangan Ganjil")
            fmt.Printf("Hasil penjumlahan dengan bilangan
    berikutnya %d + %d = %d\n", num, num+1, hasil)
        }
    }
```

Screenshoot program



Deskripsi program

Program ini adalah aplikasi pengolah bilangan bulat yang melakukan operasi aritmatika berbeda tergantung pada kategori angka yang diinputkan. Program meminta satu input *integer* num, lalu mengevaluasinya menggunakan struktur kontrol switch tanpa kondisi (*tagless switch*). Setiap kategori memiliki rumus

perhitungan unik, seperti penjumlahan, pembagian, pengkuadratan, atau perkalian, yang hasilnya kemudian diformat dan ditampilkan menggunakan fmt.Printf.

Logika program ini sangat bergantung pada urutan prioritas pengecekan di dalam blok switch. Program mengecek kondisi dari atas ke bawah: pertama apakah angka tersebut spesifik 5, lalu apakah kelipatan 10, kemudian kelipatan 5, dan terakhir apakah angka tersebut genap (kelipatan 2). Karena sifat switch di Go yang otomatis berhenti setelah menemukan kondisi yang benar, angka seperti 20 (yang sebenarnya memenuhi syarat kelipatan 10, kelipatan 5, dan genap) hanya akan dieksekusi oleh *case* "Kelipatan 10" saja, sedangkan kondisi di bawahnya akan diabaikan. Angka yang tidak memenuhi semua kriteria tersebut akan masuk ke blok default dan dikategorikan sebagai bilangan ganjil.