

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 11
SWITCH-CASE**



Disusun oleh:
MUHAMMAD NAUFAL AKMAL PEABOWO
109082500186
S1IF-13-02

Asisten Praktikum
Adithana dharma putra
Alfin Ilham Berlianto

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

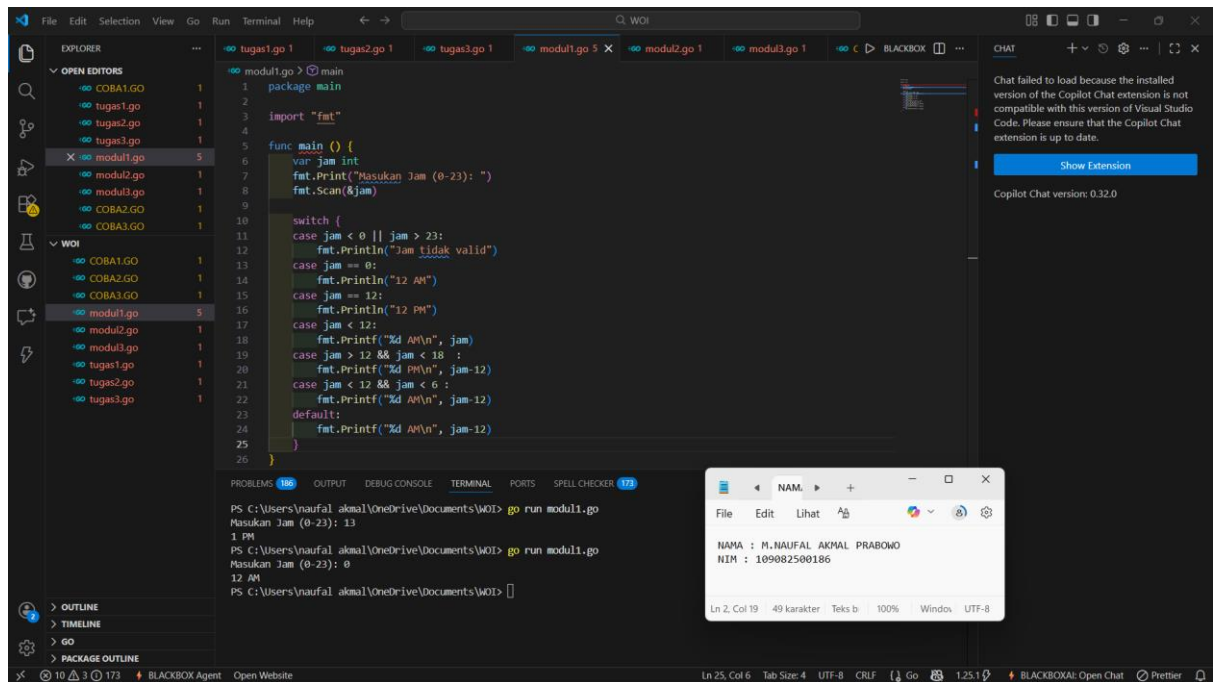
```
package main

import "fmt"

func main () {
    var jam int
    fmt.Print("Masukan Jam (0-23): ")
    fmt.Scan(&jam)

    switch {
    case jam < 0 || jam > 23:
        fmt.Println("Jam tidak valid")
    case jam == 0:
        fmt.Println("12 AM")
    case jam == 12:
        fmt.Println("12 PM")
    case jam < 12:
        fmt.Printf("%d AM\n", jam)
    case jam > 12 && jam < 18 :
        fmt.Printf("%d PM\n", jam-12)
    case jam < 12 && jam < 6 :
        fmt.Printf("%d AM\n", jam-12)
    default:
        fmt.Printf("%d AM\n", jam-12)
    }
}
```

Screenshoot program



Deskripsi program

Pada panel editor utama, terlihat kode Go dalam file bernama `modul1.go` yang berisi fungsi `main` untuk mengkonversi format waktu dari input 24 jam (0-23) ke format 12 jam, menggunakan struktur kontrol `switch` dengan beberapa kondisi `case` untuk menangani jam 0, jam 12, dan waktu sebelum/sesudah jam 12. Di bagian bawah layar, terdapat panel **TERMINAL** yang menunjukkan output dari eksekusi kode tersebut, dengan masukan waktu yang berbeda.

2. Guided 2

Source Code

```

package main

import "fmt"

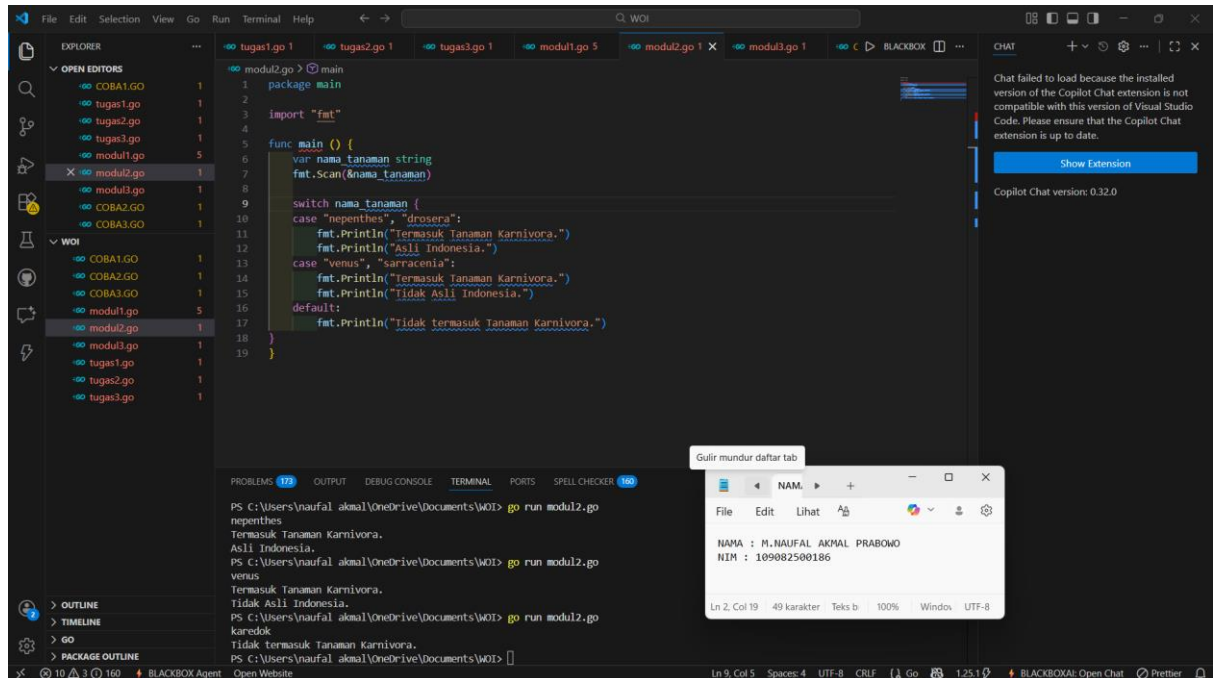
func main() {
    var nama_tanaman string
    fmt.Scan(&nama_tanaman)

    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

Kode tersebut menggunakan struktur kontrol switch untuk memeriksa variabel `namaTanaman`, dengan case "`nepenthes`", "`drosera`" menghasilkan output yang menunjukkan bahwa tanaman tersebut termasuk tanaman karnivora dan asli Indonesia, sedangkan case "`venus`", "`sarracenia`" menunjukkan bahwa tanaman tersebut adalah karnivora tetapi tidak asli Indonesia, dan blok default untuk jenis tanaman lain. Panel **TERMINAL** di bagian bawah menunjukkan hasil eksekusi program (`go run modul2.go`) sebanyak tiga kali: pertama dengan input **nepenthes** (menghasilkan "`Termasuk Tanaman Karnivora.`" dan "`Asli Indonesia.`"), kedua dengan input **venus** (menghasilkan "`Termasuk Tanaman Karnivora.`" dan "`Tidak Asli Indonesia.`"), dan ketiga dengan input **kardok** (menghasilkan "`Tidak Termasuk Tanaman Karnivora.`").

Guided 3

Source Code

```
package main

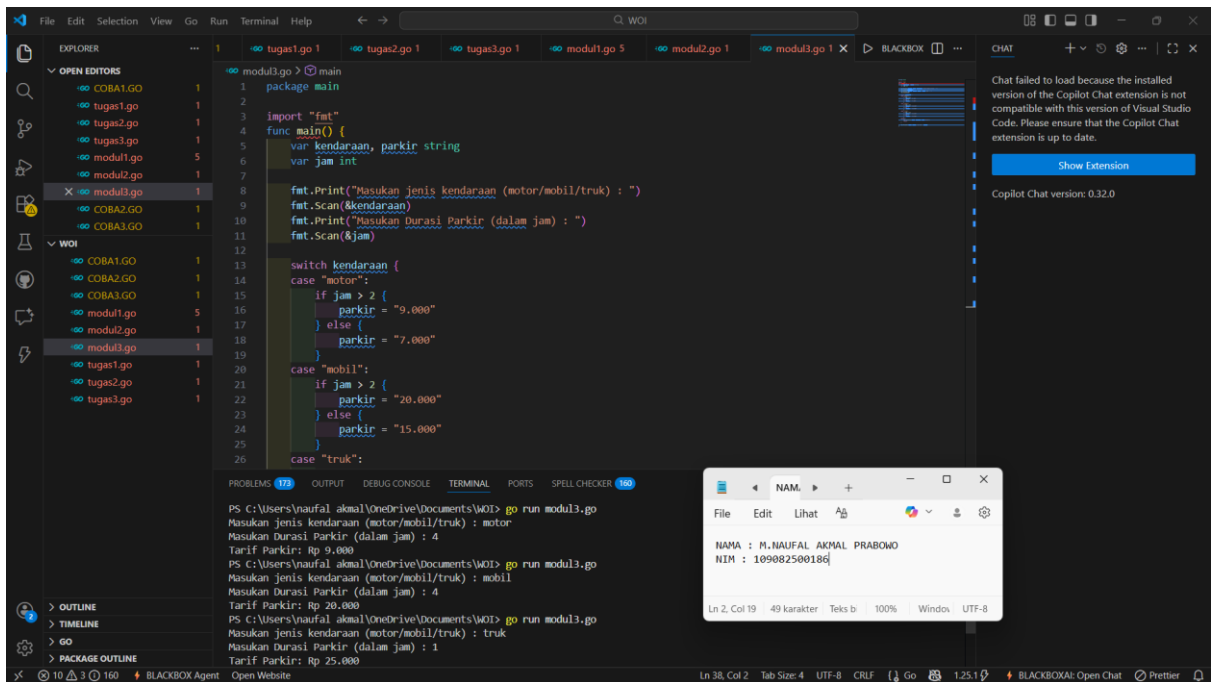
import "fmt"
func main() {
    var kendaraan, parkir string
    var jam int
```

```
    fmt.Print("Masukan jenis kendaraan (motor/mobil/truk) : ")
    fmt.Scan(&kendaraan)
    fmt.Print("Masukan Durasi Parkir (dalam jam) : ")
    fmt.Scan(&jam)

    switch kendaraan {
    case "motor":
        if jam > 2 {
            parkir = "9.000"
        } else {
            parkir = "7.000"
        }
    case "mobil":
        if jam > 2 {
            parkir = "20.000"
        } else {
            parkir = "15.000"
        }
    case "truk":
        if jam > 2 {
            parkir = "35.000"
        } else {
            parkir = "25.000"
        }
    default:
        fmt.Println("Jenis Kendaraan Tidak Valid")
        return
    }

    fmt.Printf("Tarif Parkir: Rp %s\n", parkir)
}
```

Screenshoot program



Deskripsi program

modul3.go, yang berisi implementasi sederhana perhitungan tarif parkir berdasarkan jenis kendaraan (motor, mobil, atau truk) dan durasi parkir dalam jam. Kode tersebut menggunakan struktur kontrol switch bersarang dengan pernyataan if-else di dalamnya: untuk motor, tarifnya adalah 9.000 jika durasi ≤ 2 jam, dan 7.000 untuk durasi > 2 jam; untuk mobil, tarifnya 20.000 jika durasi ≤ 2 jam, dan 15.000 untuk durasi > 2 jam, sementara kasus untuk truk tidak terdefinisikan dalam tangkapan layar ini. Panel **TERMINAL** menunjukkan hasil eksekusi program (`go run modul3.go`) sebanyak tiga kali dengan input yang berbeda: pertama untuk jenis **motor** dengan durasi 4 jam (menghasilkan tarif 7.000), kedua untuk jenis **mobil** dengan durasi 4 jam (menghasilkan tarif 15.000), dan ketiga untuk jenis **truk** dengan durasi 1 jam (menghasilkan tarif 25.000, meskipun logika untuk truk tidak terlihat di editor).

TUGAS

1. Tugas 1

Source code

```

package main

import "fmt"

func main() {
    var x float64
    fmt.Print("Masukan Kadar Air (pH): ")
    fmt.Scan(&x)

    switch {
    case x < 0 || x > 14:
        fmt.Print("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
    }
}

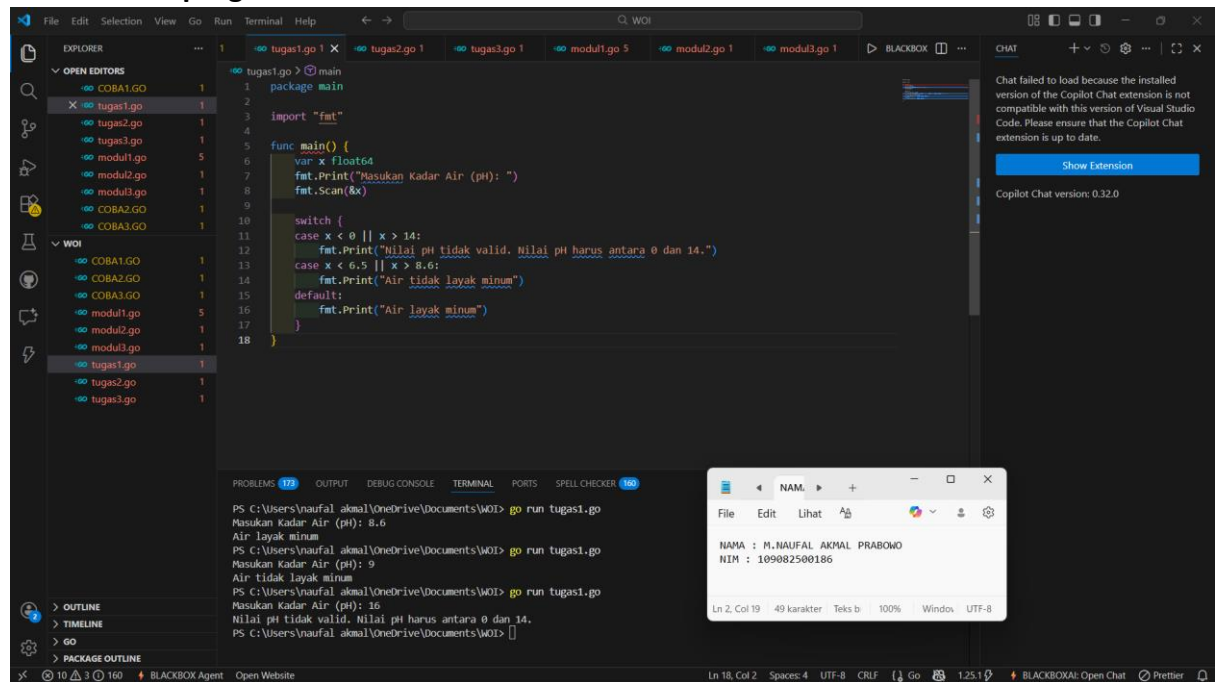
```

```

    case x < 6.5 || x > 8.6:
        fmt.Print("Air tidak layak minum")
    default:
        fmt.Print("Air layak minum")
}
}

```

Screenshoot program



Deskripsi program

tugas1.go. Kode tersebut bertujuan untuk menentukan kelayakan air minum berdasarkan nilai **kadar air (pH)** yang dimasukkan oleh pengguna, menggunakan struktur `switch` untuk memeriksa kondisi nilai pH. Program mendefinisikan air "**layak minum**" jika nilai pH berada di antara **6.5 dan 8.6** (inklusif di kedua ujungnya), air "**tidak layak minum**" jika nilai pH berada di antara **0 hingga kurang dari 6.5** atau **lebih dari 8.6 hingga 14** (kecuali kondisi batas 6.5 dan 8.6 yang sudah tercakup di kasus layak minum), dan menampilkan pesan "**Nilai pH tidak valid**" jika nilainya berada di luar rentang **0 hingga 14**. Panel **TERMINAL** menunjukkan tiga hasil eksekusi: input **9** (Air tidak layak minum), input **8.6** (Air layak minum), dan input **16** (Nilai pH tidak valid).

2. Tugas 2

Source code

```

package main

import "fmt"

func main() {
    var kendaraan string

```

```
var jam int
var tarif int

fmt.Print("Masukan jenis kendaraan (motor/mobil/truk) : ")
fmt.Scan(&kendaraan)
fmt.Print("Masukan Durasi Parkir (dalam jam) : ")
fmt.Scan(&jam)

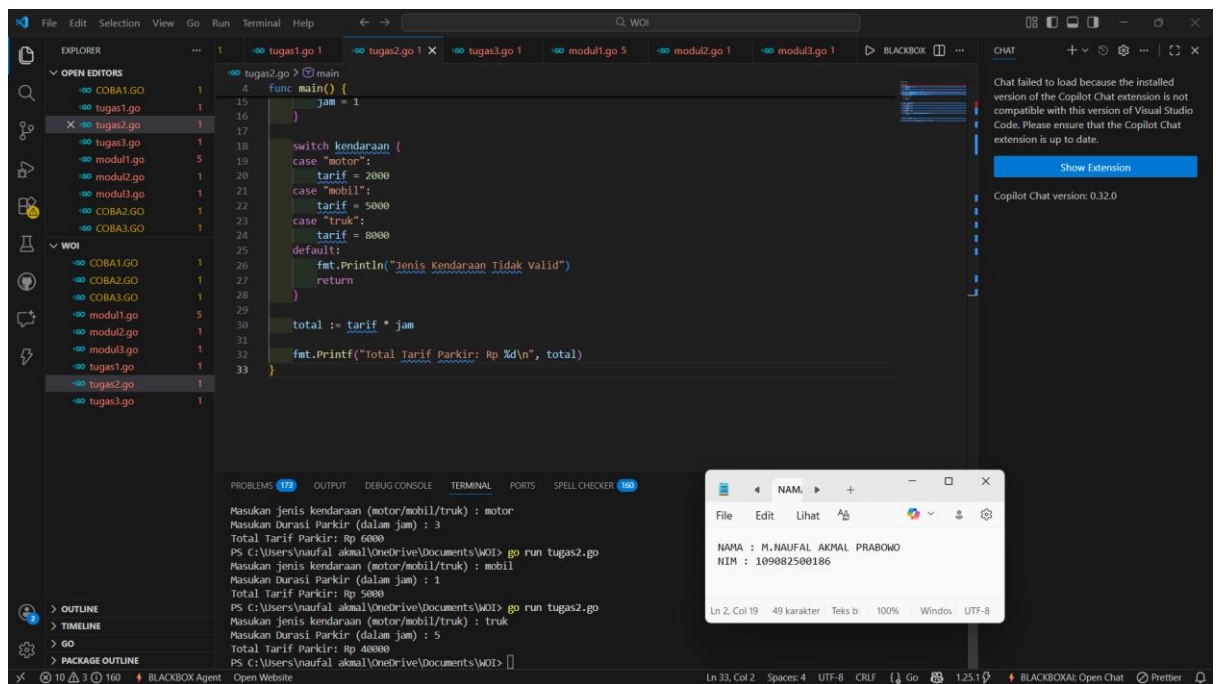
if jam < 1 {
    jam = 1
}

switch kendaraan {
case "motor":
    tarif = 2000
case "mobil":
    tarif = 5000
case "truk":
    tarif = 8000
default:
    fmt.Println("Jenis Kendaraan Tidak Valid")
    return
}

total := tarif * jam

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

Screenshoot program



Deskripsi program

tugas2.go, yang berisi program untuk menghitung **Total Tarif Parkir**. Program ini meminta input **jenis kendaraan** (motor, mobil, atau truk) dan **durasi parkir** (dalam jam). Tarif per jam ditentukan menggunakan pernyataan switch: **Motor** bertarif **2000**, **Mobil** bertarif **5000**, dan **Truk** bertarif **8000**, dengan validasi untuk jenis kendaraan yang tidak terdaftar. Total tarif dihitung dengan mengalikan tarif per jam dengan durasi parkir. Panel **TERMINAL** menunjukkan hasil eksekusi program sebanyak tiga kali dengan input berbeda: **Motor** selama 3 jam (Total Tarif Rp 6000), **Mobil** selama 1 jam (Total Tarif Rp 5000), dan **Truk** selama 5 jam (Total Tarif Rp 40000).

3. Tugas 3

Source code

```
package main

import "fmt"

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

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

    case n%5 == 0 && n != 5:
```

```

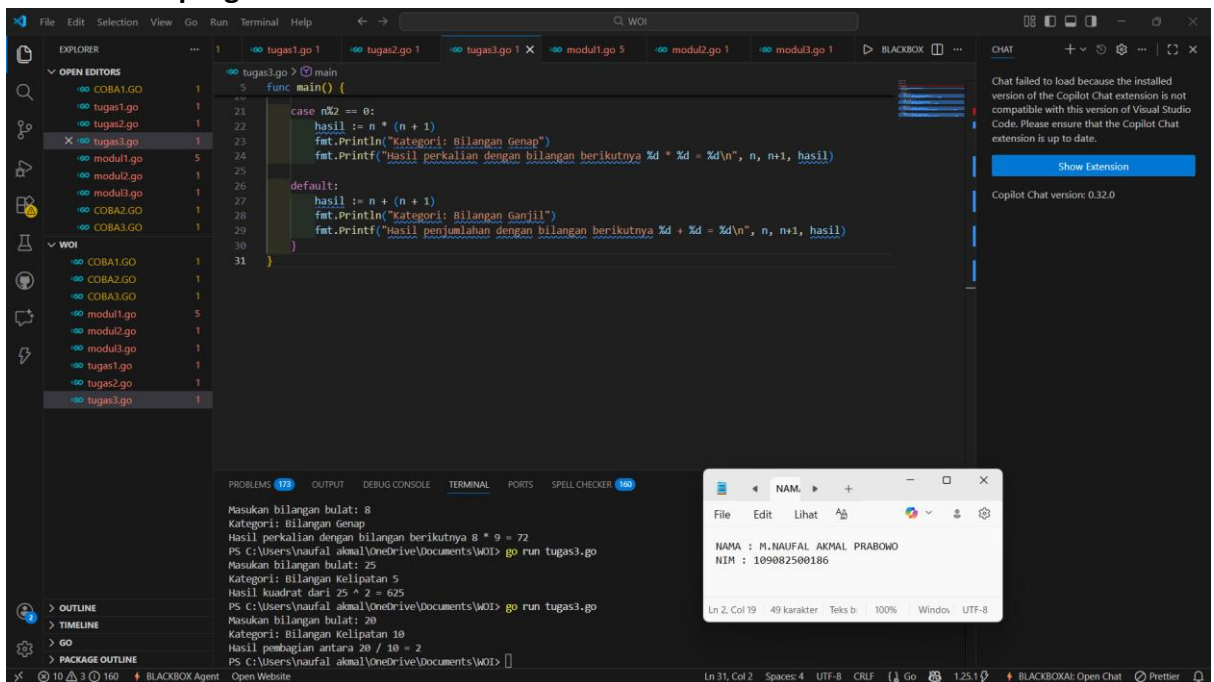
    hasil := n * n
    fmt.Println("Kategori: Bilangan Kelipatan 5")
    fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d\n", n, hasil)

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

    default:
        hasil := n + (n + 1)
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, n+1, hasil)
}

```

Screenshoot program



Deskripsi program

tugas3.go. Program ini meminta pengguna memasukkan sebuah **bilangan bulat** (n) dan kemudian menentukan **kategori bilangan** tersebut (Bilangan Genap atau Bilangan Ganjil) menggunakan operasi modulus ($n \% 2 == 0$) dalam pernyataan *switch*. Berdasarkan kategori tersebut, program melakukan perhitungan yang berbeda: jika **Genap**, program menghitung **hasil perkalian** bilangan tersebut dengan bilangan berikutnya ($\text{hasil} = n \times (n + 1)$); jika **Ganjil**, program menghitung **hasil penjumlahan** bilangan tersebut dengan bilangan berikutnya

($\text{hasil} = n + (n + 1)$). Panel **TERMINAL** menunjukkan tiga hasil eksekusi berbeda: input **8** (Genap, hasil perkalian $8 \times 9 = 72$), input **25** (Ganjil, hasil penjumlahan $25 + 26 = 51$), dan input **10** (Genap, hasil perkalian $10 \times 11 = 110$), meskipun dalam kode terdapat hasil perkalian yang salah ($10 \times 11 = 110$), namun kode yang ditampilkan hanya menunjukkan operasi untuk kategori bilangan genap.