

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 10
SWITCH-CASE



Disusun oleh:

LEONARDO FARRIZ GARCYA

109082530036

S1IF-13-04

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,waktu  int

    fmt.Scan(&jam)

    switch{

    case jam == 0 :

        waktu = 12

        fmt.Printf("%d am", waktu)

    case jam < 12:

        fmt.Printf("%d am", jam)

    case jam == 12:

        fmt.Printf("%d pm", jam)

    case jam > 12 && jam < 24:

        waktu = jam % 12

        fmt.Printf("%d pm", waktu)
```

```

default:

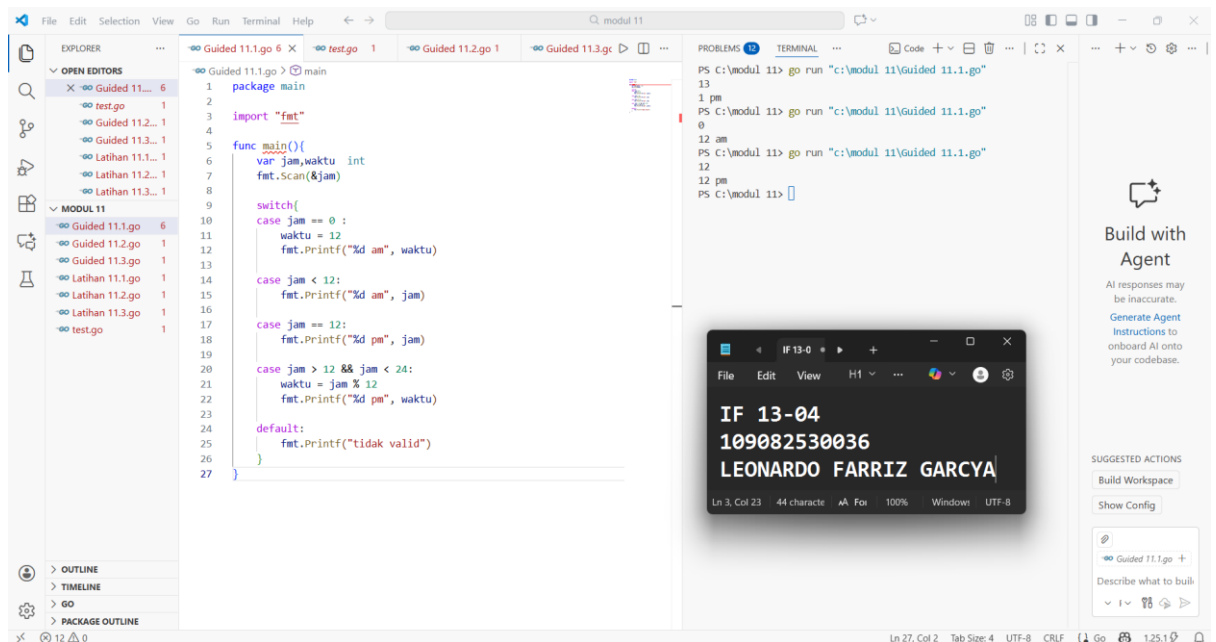
    fmt.Printf("tidak valid")

}

}

```

Screenshoot program



Deskripsi program

1.package main

- Bahwa program ini adalah program utama yang bisa langsung dijalankan.
- Tanpa main, kode tidak bisa dieksekusi sebagai aplikasi.

2.import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
- "fmt.Println" : menampilkan output ke layar.

3.func main () { ... }

- Fungsi main adalah untuk titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini

4. var jam,waktu int

- Menyatakan variabel jam dan waktu bertipe integer
- Jam untuk format 24 jam

- Waktu untuk menghitung jam hasil konversi 12 jam.

5. `fmt.Scan(&jam)`

- Membaca input dari pengguna dan menyimpannya ke variabel `jam`

6. `case jam == 0:`

`waktu = 12`

`fmt.Printf("%d am", waktu)`

- Jika `jam = 0`, maka dalam format 12 jam, menjadi 12 AM.

7. `case jam < 12:`

`fmt.Printf("%d am", jam)`

-Jika `jam < 12`, maka jam 01–11 tetap sama, tapi dengan keterangan AM.

8. `case jam == 12:`

`fmt.Printf("%d pm", jam)`

- Untuk jam 12 siang = jam 12pm

9. `case jam > 12 && jam < 24:`

`waktu = jam % 12`

`fmt.Printf("%d pm", waktu)`

- Kondisi ini berlaku untuk jam 13–23.

- untuk jam 13 = 1PM dan seterusnya

10. `default:`

`fmt.Printf("tidak valid")`

- Jadi ini untuk kondisi jika case di atas tidak terpenuhi

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var tanaman string

    fmt.Scan(&tanaman)

    switch 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:

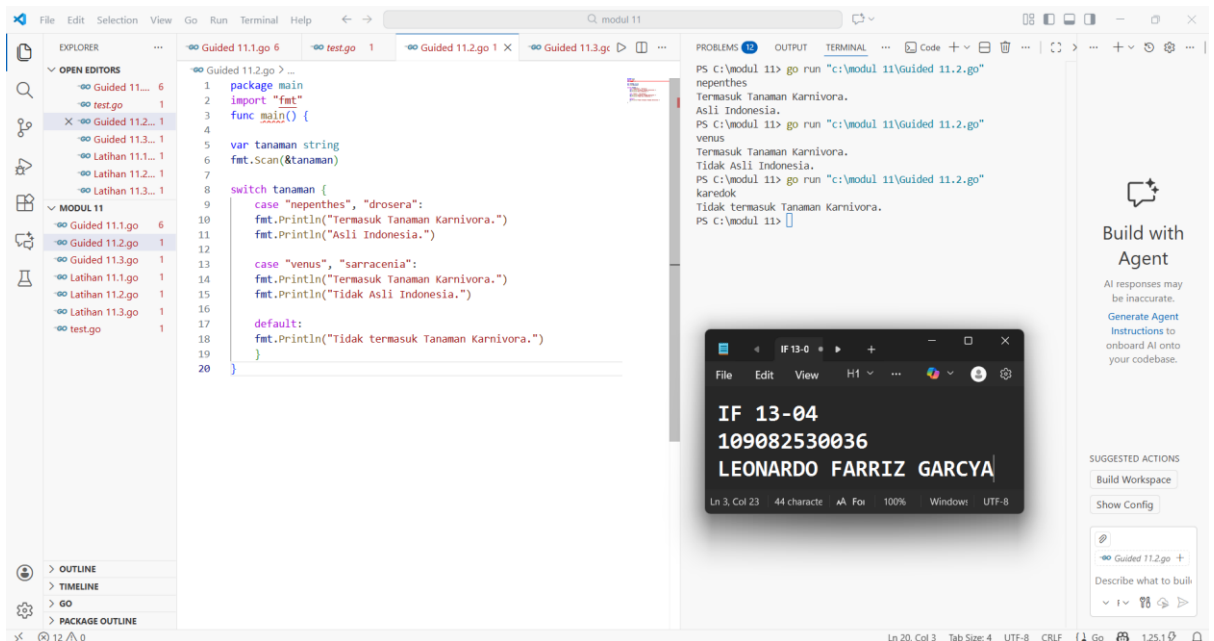
    fmt.Println("Tidak termasuk Tanaman Karnivora.")

}

}

```

Screenshoot program



Deskripsi program

1.package main

- Bahwa program ini adalah program utama yang bisa langsung dijalankan.
- Tanpa main, kode tidak bisa dieksekusi sebagai aplikasi.

2. import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
"fmt.Println" : menampilkan output ke layar.

3. func main () { ... }

- Fungsi main adalah untuk titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini

4. var tanaman string

- Menyatakan variabel tanaman bertipe string

5. fmt.Scan(&tanaman)

- Membaca input dari pengguna dan menyimpannya ke variabel tanaman

6. switch tanaman {

- Jadi switch di sini membandingkan nilai string dari inputan user.

7. case "nepenthes", "drosera":

```
    fmt.Println("Termasuk Tanaman Karnivora.")  
    fmt.Println("Asli Indonesia.")
```

- Jadi disini jika user mengetik nepenthes , maka outputnya: karnivora + asli Indonesia.
- Dan jika user mengetik drosera, maka outputnya: karnivora + asli Indonesia.

8. case "venus", "sarracenia":

```
    fmt.Println("Termasuk Tanaman Karnivora.")  
    fmt.Println("Tidak Asli Indonesia.")
```

- Ini persis seperti case di atas

9. case "venus", "sarracenia":

```
    fmt.Println("Termasuk Tanaman Karnivora.")  
    fmt.Println("Tidak Asli Indonesia.")
```

- Jika user memasukkan tanaman selain empat di atas, maka dianggap bukan tanaman karnivora.

3. Guided 3

Source Code

```
package main  
  
import "fmt"
```

```
func main(){

    var jenis string

    var waktu,harga int

    fmt.Scan(&jenis, &waktu)


    switch jenis {

    case "motor":

        harga = 7000

        if waktu > 2{

            harga += 2000

        }

        fmt.Printf("tarif parkir: %d", harga)


    case "mobil":

        harga = 15000

        if waktu > 2{

            harga += 5000

        }

        fmt.Printf("tarif parkir: %d", harga)


    case "truk":

        harga = 25000

        if waktu > 2{

            harga += 10000

        }

    }
```

```

        fmt.Printf("tarif parkir: %d", harga)

    default:

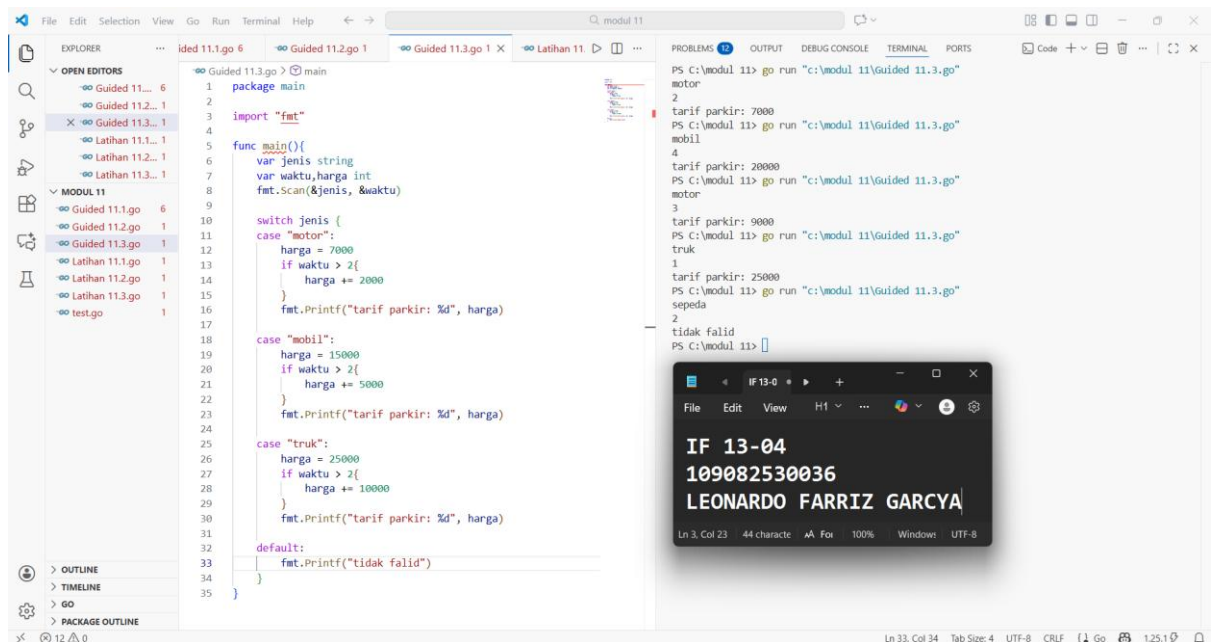
        fmt.Printf("tidak falid")

    }

}

```

Screenshoot program



Deskripsi program

1.package main

- Bahwa program ini adalah program utama yang bisa langsung dijalankan.
- Tanpa main, kode tidak bisa dieksekusi sebagai aplikasi.

2.import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
- "fmt.Println" : menampilkan output ke layar.

3.func main () { ... }

- Fungsi main adalah untuk titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini

4. var jenis string

- Menyatakan variabel jenis bertipe string

5. var waktu,harga int

- Menyatakan variabel waktu dan harga bertipe integer

6. `fmt.Scan(&jenis, &waktu)`

- Membaca input dari pengguna dan menyimpannya ke variabel jenis dan waktu.

7. `switch jenis`

- Jadi switch di sini membandingkan nilai string dari inputan user.

8. `case "motor":`

`harga = 7000`

`if waktu > 2 { harga += 2000 }`

`fmt.Printf("tarif parkir: %d", harga)`

- Jadi ini untuk case motor.

- Dengan tarif dasar = 7000. Dan jika waktu lebih dari 2 jam, maka tambah 2000.

9. `case "mobil":`

`harga = 15000`

`if waktu > 2 { += 500 }`

`fmt.Printf("tarif parkir: %d", harga)`

- Jadi ini untuk case mobil.

- Dengan tarif dasar = 15000. Dan jika waktu lebih dari 2 jam, maka tambah 5000.

10. `case "truk":`

`harga = 25000`

`if waktu > 2 { harga += 10000 }`

`fmt.Printf("tarif parkir: %d", harga)`

- Jadi ini untuk case truk.

- Dengan tarif dasar = 25000. Dan jika waktu lebih 2 jam, maka tambah 10000.

11. `default:`

`fmt.Printf("tidak valid")`

- Jadi ini jika user memasukan inputan selain 3 di atas, maka "tidak valid"

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main(){

    var pH float64

    fmt.Scan(&pH)

    switch {

    case pH < 0 || pH > 14:

        fmt.Println("Tidak Valid")

    case pH >= 6.5 && pH <= 8.6:

        fmt.Println("Air Layak Minum")

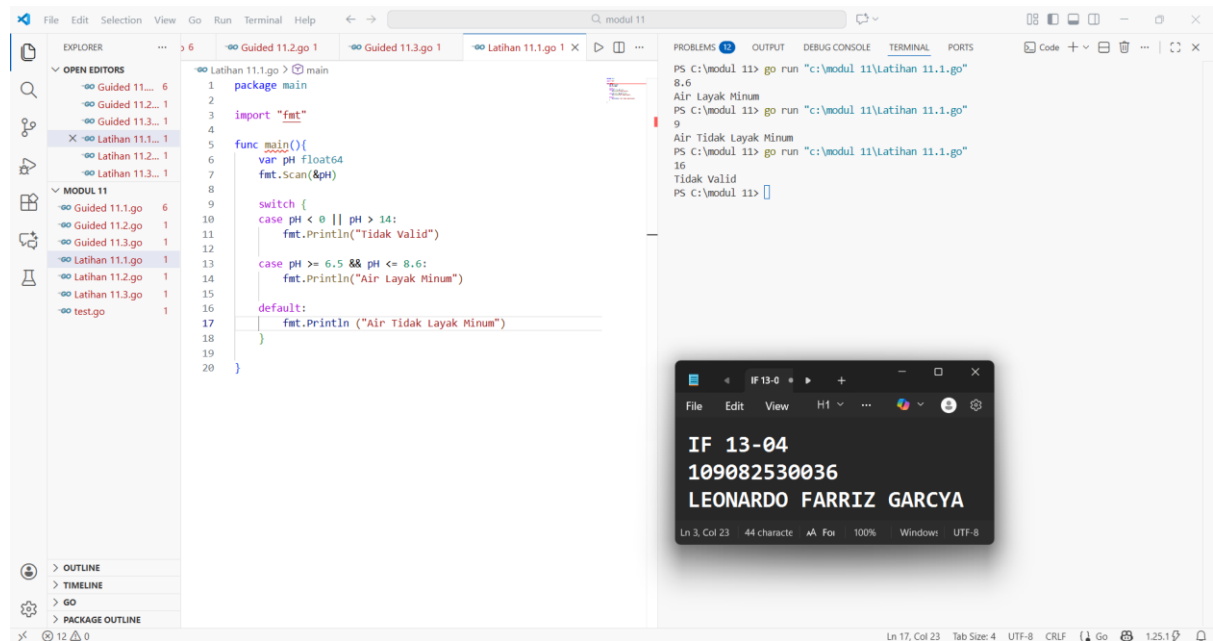
    default:

        fmt.Println (" Air Tidak Layak Minum")

    }

}
```

Screenshoot program



Deskripsi program

1.package main

- Bahwa program ini adalah program utama yang bisa langsung dijalankan.
- Tanpa main, kode tidak bisa dieksekusi sebagai aplikasi.

2.import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
- "fmt.Println" : menampilkan output ke layar.

3.func main () { ... }

- Fungsi main adalah untuk titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini

4. case pH < 0 || pH > 14:

```
fmt.Println("Tidak Valid")
```

- Jadi case pertama ini menggunakan pH berada pada rentang 0–14.
- Jika lebih kecil dari 0 atau lebih besar dari 14 , maka "tidak valid".

5. case pH >= 6.5 && pH <= 8.6:

```
fmt.Println("Air Layak Minum")
```

- Untuk case ke dua menggunakan pH yang aman untuk air layak minum, ada di rentang 6.5 hingga 8.6.

6. default:

```
fmt.Println(" Air Tidak Layak Minum")
```

- code ini akan jalan jika pH valid tetapi di luar rentang layak minum (0–6.5 atau 8.6–14), maka hasilnya adalah air tidak layak minum.

2. Tugas 2

Source code

```
package main

import "fmt"

func main(){

    var jenis string

    var durasi, harga int

    fmt.Scan(&jenis, &durasi)

    switch jenis {

        case "motor":

            harga = durasi*2000

        case "mobil":

            harga = durasi*5000

        case "truk":

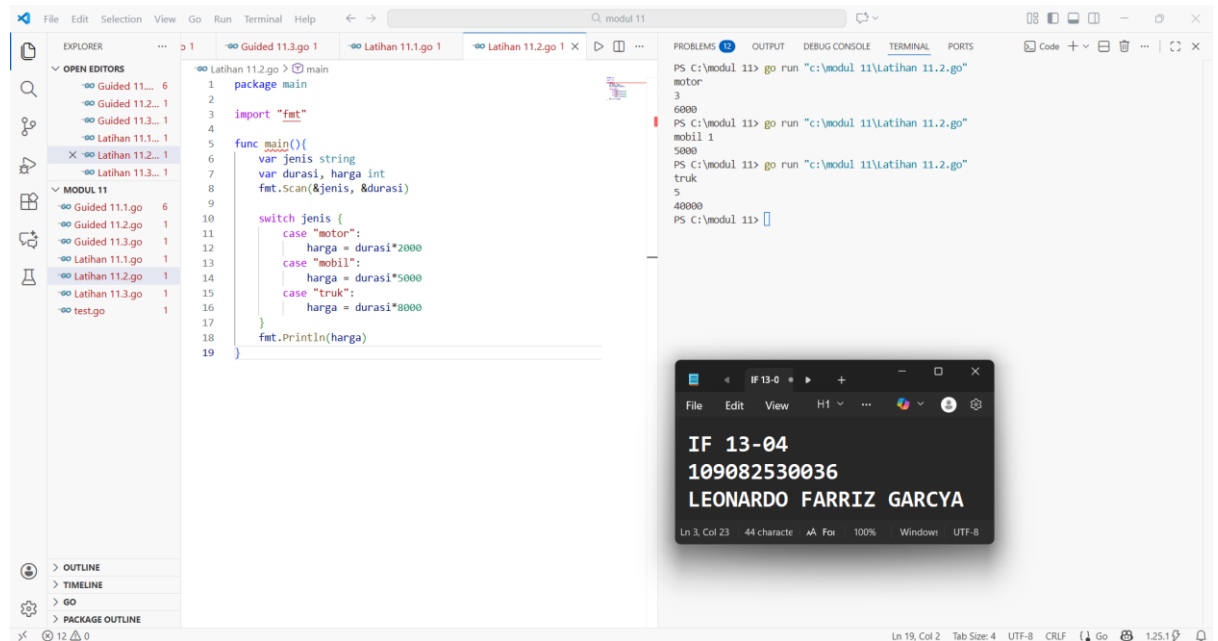
            harga = durasi*8000

    }

    fmt.Println(harga)

}
```

Screenshoot program



Deskripsi program

1.package main

- Bahwa program ini adalah program utama yang bisa langsung dijalankan.
- Tanpa main, kode tidak bisa dieksekusi sebagai aplikasi.

2.import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
- "fmt.Println" : menampilkan output ke layar.

3.func main () { ... }

- Fungsi main adalah untuk titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini

4.var jenis string

- Menyatakan variabel jenis bertipe string.

5.var durasi ,harga int.

- Menyatakan variabel durasi dan harga bertipe integer.

6. fmt.Scan(&jenis, &durasi)

- Membaca input dari pengguna dan menyimpannya ke variabel jenis dan durasi.

7. switch jenis

- Jadi switch di sini membandingkan nilai string dari inputan user.

8. case "motor":

harga = durasi * 2000

- Jadi case pertama ini motor, dengan tarif motor = 2000 per jam.

9. case "mobil":

harga = durasi * 5000

- Disini case ke dua mobil, dengan tarif mobil = 5000 per jam.

10. case "truk":

harga = durasi * 8000

- Dan case ke tiga truk, dengan tarif truk = 8000 per jam.

3. Tugas 3

Source code

```
package main
import "fmt"

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

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

    case n%5 == 0 && n != 5:
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", n, n*n)

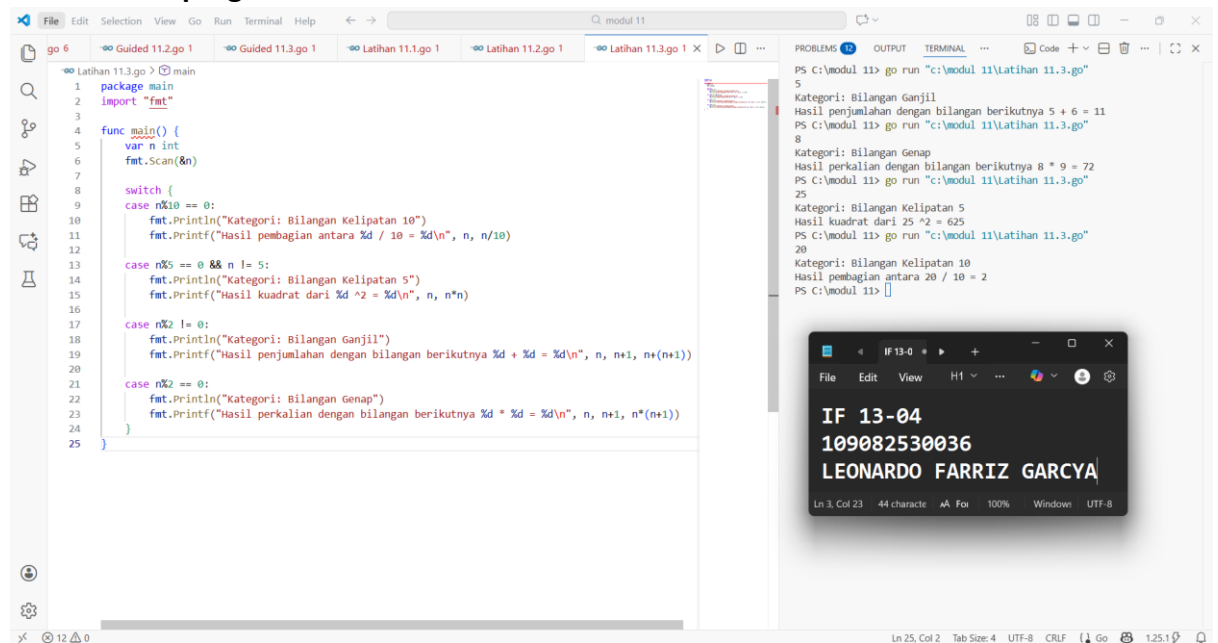
    case n%2 != 0:
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, n+1, n+(n+1))
    }
```

```

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

```

Screenshoot program



Deskripsi program

1.package main

- Bahwa program ini adalah program utama yang bisa langsung dijalankan.
- Tanpa main, kode tidak bisa dieksekusi sebagai aplikasi.

2.import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
- "fmt.Println" : menampilkan output ke layar.

3.func main () { ... }

- Fungsi main adalah untuk titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini

4.var n int

- Menyatakan variabel n bertipe integer

5. fmt.Scan(&n)

- Membaca input dari pengguna dan menyimpannya ke variabel n.

6. case $n \% 10 == 0$:

```
fmt.Println("Kategori: Bilangan Kelipatan 10")
```

```
fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, n/10)
```

- case pertama yaitu kelipatan 10.
- Dengan kondisi $n \% 10 == 0$, artinya n adalah kelipatan 10.
- Jadi outpunya akan mengatakan bahwa bilangan tersebut kelipatan 10 dan menampilkan hasil pembagian, $n \div 10$.

7. case $n \% 5 == 0 \ \&\& \ n \neq 5$:

```
fmt.Println("Kategori: Bilangan Kelipatan 5")
```

```
fmt.Printf("Hasil kuadrat dari %d ^2 = %d\n", n, n*n)
```

- case ke dua yaitu kelipatan 5.
- Dengan kondisi $n \% 5 == 0$, artinya n adalah kelipatan 5.
- Dan $n \neq 5$, maksudnya disini adalah bilangan 5 dikecualikan agar tidak masuk case ini.
- Jadi outpunya akan mengatakan bahwa bilangan tersebut kelipatan 5 dan menampilkan hasil kuadratnya: $n \times n$.

8. case $n \% 2 \neq 0$:

```
fmt.Println("Kategori: Bilangan Ganjil")
```

```
fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, n+1, n+(n+1))
```

- case ketiga yaitu bilangan ganjil.
- Dengan kondisi $n \% 2 \neq 0$, artinya n adalah bilangan ganjil.
- Jadi outpunya akan mengatakan bahwa bilangan tersebut bilangan ganjil dan menampilkan hasil perjumlahan bilangan tersebut dengan bilangan setelahnya ($n + (n+1)$).

9. case $n \% 2 == 0$:

```
fmt.Println("Kategori: Bilangan Genap")
```

```
fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", n, n+1, n*(n+1))
```

- case keempat yaitu bilangan genap.
- Dengan kondisi $n \% 2 == 0$, artinya n adalah bilangan genap.
- Jadi outpunya akan mengatakan bahwa bilangan tersebut bilangan genap dan menampilkan hasil perkalian dengan bilangan setelahnya ($n \times (n + 1)$).