

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 11
SWITCH CASE



Disusun oleh:

RASYA SYAHRI RAMADHAN

103112430189

S1IF-12-07

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

DASAR TEORI

1. IF

Pernyataan **if** digunakan untuk menentukan apakah suatu kondisi tertentu bernilai **true** terpenuhi. Jika kondisi pada pernyataan **if** terpenuhi (**true**), maka blok kode di dalam pernyataan tersebut akan dieksekusi. Sebaliknya, jika kondisi tidak terpenuhi (**false**), maka blok kode di dalam pernyataan tersebut dilewati, dan program melanjutkan eksekusi di luar blok **if**.

2. ELSE

Pernyataan **if-else** digunakan ketika ada dua jalur logika, yaitu:

- a. Jalur yang dijalankan jika kondisi bernilai **true**.
- b. Jalur alternatif yang dijalankan jika kondisi bernilai **false**. dijalankan.

Blok **if** dieksekusi jika kondisi bernilai **true**, dan jika tidak, blok **else** akan dijalankan. Pada akhirnya, salah satu blok akan dijalankan, tetapi tidak keduanya.

3. ELSE-IF

Pernyataan **else-if** digunakan ketika ada lebih dari dua jalur logika yang ingin diperiksa. Berbeda dengan **if-else**, pernyataan **else-if** memungkinkan Anda memeriksa beberapa kondisi secara berurutan. Program akan memeriksa kondisi satu per satu dari atas ke bawah. Jika salah satu kondisi bernilai **true**, blok kode yang sesuai akan dijalankan, dan semua kondisi berikutnya dilewati. Jika tidak ada kondisi yang bernilai **true**, maka blok **else** (opsional) akan dijalankan untuk menangani kasus default

TUGAS PENDAHULUAN

1. Tugas 1

Pseudo code

```
rumus
var umur int

algor
itma
input
umur

switch
case kondisi_1 : jika umur == 0 // aksi kondisi 1 :
output tidak terdefinisi dalam umur

case kondisi_2 : jika umur < 13 // aksi kondisi 2 : output
anak anak

case kondisi_3 : jika umur < 20 // aksi
kondisi 3 : output remaja

case kondisi_4 : jika umur < 60 // aksi
kondisi 4 : output dewasa

default // aksi : output lansia

end switch

end program
```

Source code

```
package main

import "fmt"

func main() {

    var usia int

    fmt.Print("Masukkan usia: ")

    fmt.Scan(&usia)

    switch {

    case usia < 0:

        fmt.Println("Usia tidak valid.")

    case usia < 13:

        fmt.Println("Kategori: Anak-anak")

    case usia < 20:

        fmt.Println("Kategori: Remaja")

    case usia < 60:

        fmt.Println("Kategori: Dewasa")

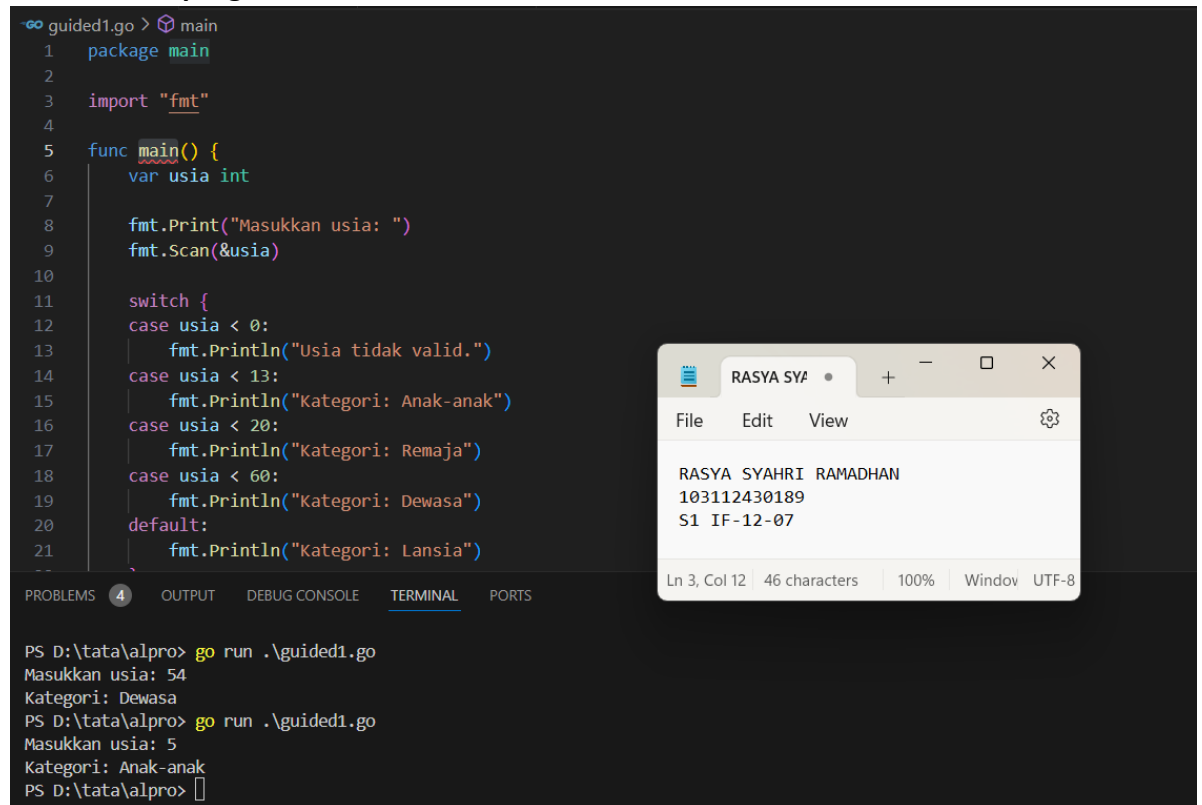
    default:

        fmt.Println("Kategori: Lansia")

    }

}
```

Screenshoot program



The screenshot shows a Go program in an IDE (Visual Studio Code) and its execution in a terminal. The program is a simple age categorization tool. It prompts the user to enter an age and then prints the corresponding category based on a switch statement. The terminal shows two runs: one with age 54 resulting in 'Dewasa' and one with age 5 resulting in 'Anak-anak'.

```
guided1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var usia int
7
8     fmt.Print("Masukkan usia: ")
9     fmt.Scan(&usia)
10
11     switch {
12     case usia < 0:
13         fmt.Println("Usia tidak valid.")
14     case usia < 13:
15         fmt.Println("Kategori: Anak-anak")
16     case usia < 20:
17         fmt.Println("Kategori: Remaja")
18     case usia < 60:
19         fmt.Println("Kategori: Dewasa")
20     default:
21         fmt.Println("Kategori: Lansia")
22     }
23 }
```

Terminal Output:

```
PS D:\tata\alpro> go run .\guided1.go
Masukkan usia: 54
Kategori: Dewasa
PS D:\tata\alpro> go run .\guided1.go
Masukkan usia: 5
Kategori: Anak-anak
PS D:\tata\alpro>
```

Deskripsi program

Program di atas adalah sebuah aplikasi sederhana dalam bahasa Go yang menerima input usia dari pengguna, kemudian mengelompokkan usia tersebut ke dalam kategori tertentu (anak-anak, remaja, dewasa, atau lansia).

switch tanpa ekspresi: Program tidak mengevaluasi nilai tunggal, melainkan langsung memeriksa kondisi boolean di setiap **case**.

Kondisi dalam case:

1. **case usia < 0:** Jika usia kurang dari 0, output akan menyatakan bahwa usia tidak valid.
2. **case usia < 13:** Usia di bawah 13 tahun termasuk kategori *anak-anak*.
3. **case usia < 20:** Usia 13–19 tahun termasuk kategori *remaja*.
4. **case usia < 60:** Usia 20–59 tahun termasuk kategori *dewasa*.
5. **default:** Usia 60 tahun ke atas termasuk kategori *lansia*.

2. Tugas 2

Pseudo code

Program_HariMulai

Rumus

var hari int

Algoritma

```
output("Masukkan angka hari: ")input(hari)

switch
case hari = 1: ("Senin")

case hari = 2: ("Selasa")

case hari = 3: ("Rabu")

case hari = 4: ("Kamis")

case hari = 5: ("Jumat")

case hari = 6: ("Sabtu")

case hari = 7:
("Minggu")

default: ("Tidak terdaftar")
endprogram
```

Source code

```
package main

import "fmt"

func main() {
    var angka int
    fmt.Print("Masukkan angka hari (1-7): ")
    fmt.Scan(&angka)

    switch angka {
    case 1:
        fmt.Println("Hari: Minggu")
    case 2:
        fmt.Println("Hari: Senin")
    case 3:
        fmt.Println("Hari: Selasa")
    case 4:
        fmt.Println("Hari: Rabu")
```

```

    case 5:
        fmt.Println("Hari: Kamis")
    case 6:
        fmt.Println("Hari: Jumat")
    case 7:
        fmt.Println("Hari: Sabtu")
    default:
        fmt.Println("Angka yang dimasukkan tidak valid. Masukkan
angka antara 1 hingga 7.")
    }
}

```

Screenshoot program

The screenshot shows a Go program in an IDE. The source code defines a `main` function that prompts the user for a number (1-7) and uses a `switch` statement to print the corresponding day of the week. The terminal output shows the program being run, the user entering '5', and the program outputting 'Hari: Kamis'.

```

guided2.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var angka int
7      fmt.Print("Masukkan angka hari (1-7): ")
8      fmt.Scan(&angka)
9
10     switch angka {
11     case 1:
12         fmt.Println("Hari: Minggu")
13     case 2:
14         fmt.Println("Hari: Senin")
15     case 3:
16         fmt.Println("Hari: Selasa")
17     case 4:
18         fmt.Println("Hari: Rabu")
19     case 5:
20         fmt.Println("Hari: Kamis")
21     case 6:
22         fmt.Println("Hari: Jumat")
23     case 7:
24         fmt.Println("Hari: Sabtu")
25     default:
26         fmt.Println("Angka yang dimasukkan tidak valid. Masukkan
angka antara 1 hingga 7.")
27     }
28 }

```

Terminal Output:

```

PS D:\tata\alpro> go run .\guided2.go
Masukkan angka hari (1-7): 5
Hari: Kamis
PS D:\tata\alpro>

```

Deskripsi program

Program di atas adalah aplikasi sederhana dalam bahasa Go untuk menentukan nama hari berdasarkan angka yang dimasukkan oleh pengguna. Angka hari tersebut harus berada dalam rentang 1 hingga 7, di mana setiap angka mewakili satu hari dalam seminggu.

switch angka: Mengevaluasi nilai variabel **angka**.

case: Mencocokkan nilai **angka** dengan angka 1 hingga 7:

1. Menampilkan *"Hari: Minggu"*.
2. Menampilkan *"Hari: Senin"*.
3. Menampilkan *"Hari: Selasa"*.
4. Menampilkan *"Hari: Rabu"*.
5. Menampilkan *"Hari: Kamis"*.
6. Menampilkan *"Hari: Jumat"*.
7. Menampilkan *"Hari: Sabtu"*.
8. **default:** Jika **angka** tidak berada dalam rentang 1 hingga 7, program menampilkan pesan *"Angka yang dimasukkan tidak valid"*.

3. Tugas 3

Pseudo code

```
Program_Kode_Usia
Mulai

Rumus
var kode string

Algoritma output("Masukkan Kode: ")
input(kode)

switch kode
case "G": ("Semua usia")

case "PG": ("Anak-anak di atas 7 tahun")

case "PG-13": ("Remaja di atas 13 tahun")

case "R": ("Dewasa")

default: ("Kode tidak terdaftar")
endprogram
```

Source code

```
package main

import "fmt"

func main() {
    var kode string

    fmt.Print("Masukkan kode film (G, PG, PG-13, R): ")
    fmt.Scan(&kode)
```



```

switch kode {
case "G":
    fmt.Println("Semua usia (G)")
case "PG":
    fmt.Println("Anak-anak di atas 7 tahun (PG)")
case "PG-13":
    fmt.Println("Remaja di atas 13 tahun (PG-13)")
case "R":
    fmt.Println("Dewasa (R)")
default:
    fmt.Println("Kode tidak valid.")
}
}

```

Screenshoot program

```

guided3.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var kode string
7
8      fmt.Print("Masukkan kode film (G, PG, PG-13, R): ")
9      fmt.Scan(&kode)
10
11     switch kode {
12     case "G":
13         fmt.Println("Semua usia (G)")
14     case "PG":
15         fmt.Println("Anak-anak di atas 7 tahun (PG)")
16     case "PG-13":
17         fmt.Println("Remaja di atas 13 tahun (PG-13)")
18     case "R":
19         fmt.Println("Dewasa (R)")
20     default:
21         fmt.Println("Kode tidak valid.")
22     }
23 }

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```

PS D:\tata\alpro> go run .\guided3.go
Masukkan kode film (G, PG, PG-13, R): R
Dewasa (R)
PS D:\tata\alpro>

```

Deskripsi program

Program di atas adalah aplikasi dalam bahasa Go yang berfungsi untuk menentukan kategori usia penonton berdasarkan kode film yang dimasukkan oleh pengguna. Setiap kode film mewakili batas usia tertentu untuk menonton film, dan jika kode yang dimasukkan tidak valid, program akan memberikan pesan kesalahan.

switch kode: Mengevaluasi nilai variabel kode.

case: Memeriksa apakah nilai kode cocok dengan salah satu kategori:

1. case "G": Jika kode adalah "G", menampilkan pesan "*Semua usia (G)*".
2. case "PG": Jika kode adalah "PG", menampilkan pesan "*Anak-anak di atas 7 tahun (PG)*".
3. case "PG-13": Jika kode adalah "PG-13", menampilkan pesan "*Remaja di atas 13 tahun (PG-13)*".
4. case "R": Jika kode adalah "R", menampilkan pesan "*Dewasa (R)*".
5. default: Jika tidak ada nilai yang cocok dengan case, menampilkan pesan "*Kode tidak valid.*".

LATIHAN MODUL

1. Tugas 1

Pseudo code

```
Program_pH_Air
Mulai

Rumus
var ph float64

Algoritma
output("Masukkan nilai pH: ")
input(ph)

switch
case ph >= 6.5 && ph <= 8.6: ("Air layak diminum")

case ph < 6.5 && ph > 0 || ph > 8.6 && ph <= 14: ("air tidak layak diminum")

case ph > 14 || ph < 0: ("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14")

endprogram
```

Source code

```
package main

import "fmt"

func main() {
```

```

var ph float64

fmt.Print("Masukkan kadar pH air (0-14): ")

fmt.Scan(&ph)

switch {
case ph < 0 || ph > 14:
    fmt.Println("Input tidak valid, rentang pH 0 - 14")
case ph >= 6.5 && ph <= 8.6:
    fmt.Println("Air Layak Minum")
default:
    fmt.Println("Air Tidak Layak Minum")
}
}

```

Screenshoot program

The screenshot shows a Go program being executed in a terminal window. The program prompts the user to enter a pH value. Three test cases are shown: a value of 9 results in 'Air Tidak Layak Minum', a value of 3 results in 'Air Tidak Layak Minum', and a value of 7 results in 'Air Layak Minum'.

```

unguided1.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var ph float64
7
8      fmt.Print("Masukkan kadar pH air (0-14): ")
9      fmt.Scan(&ph)
10
11     switch {
12     case ph < 0 || ph > 14:
13         fmt.Println("Input tidak valid, rentang pH 0 - 14")
14     case ph >= 6.5 && ph <= 8.6:
15         fmt.Println("Air Layak Minum")
16     default:
17         fmt.Println("Air Tidak Layak Minum")
18     }
19 }
20
PS D:\tata\alpro> go run .\unguided1.go
Masukkan kadar pH air (0-14): 9
Air Tidak Layak Minum
PS D:\tata\alpro> go run .\unguided1.go
Masukkan kadar pH air (0-14): 3
Air Tidak Layak Minum
PS D:\tata\alpro> go run .\unguided1.go
Masukkan kadar pH air (0-14): 7
Air Layak Minum
PS D:\tata\alpro>

```

The IDE window shows the file name 'RASYA SYA' and the menu options 'File', 'Edit', 'View', and 'Settings'. The status bar at the bottom indicates 'Ln 3, Col 12 | 46 characters | 100% | Window | UTF-8'.

Deskripsi program

Program di atas adalah aplikasi sederhana dalam bahasa Go untuk menentukan apakah air dengan kadar pH tertentu termasuk layak diminum, tidak layak diminum, atau inputnya tidak valid. Program ini menggunakan input berupa angka desimal (float) untuk kadar pH, lalu memvalidasi dan mengelompokkannya menggunakan pernyataan switch.

switch tanpa ekspresi:

Evaluasi langsung pada kondisi boolean di setiap **case**.

Kondisi case:

1. **case $ph < 0 \ || \ ph > 14$:**

Jika nilai pH di luar rentang valid (0 hingga 14), program mencetak pesan *"Input tidak valid"*.

2. **case $ph \geq 6.5 \ \&\& \ ph \leq 8.6$:**

Jika nilai pH berada dalam rentang *layak minum* (6.5 hingga 8.6), program mencetak *"Air Layak Minum"*.

3. **default:**

Untuk semua nilai lain dalam rentang 0–14 tetapi di luar kategori layak minum, program mencetak *"Air Tidak Layak Minum"*.

2. Tugas 2

Pseudo code

```
Program_Tarif_Parkir_Kendaraan
Mulai

Rumus
var kendaraan string
var waktu int

Algoritma
output("Masukkan tipe kendaraan dan durasi parkir dalam
jam: ")input(kendaraan, waktu)

switch
case "motor": motor := 2000 * waktu ("Tarif parkir : ",
motor)case "mobil": mobil := 5000 * waktu ("Tarif
parkir : ", mobil)case "truk": truk := 8000 * waktu
("Tarif parkir : ", truk)

default:
("Data tidak ada")
```

Source code

```
package main

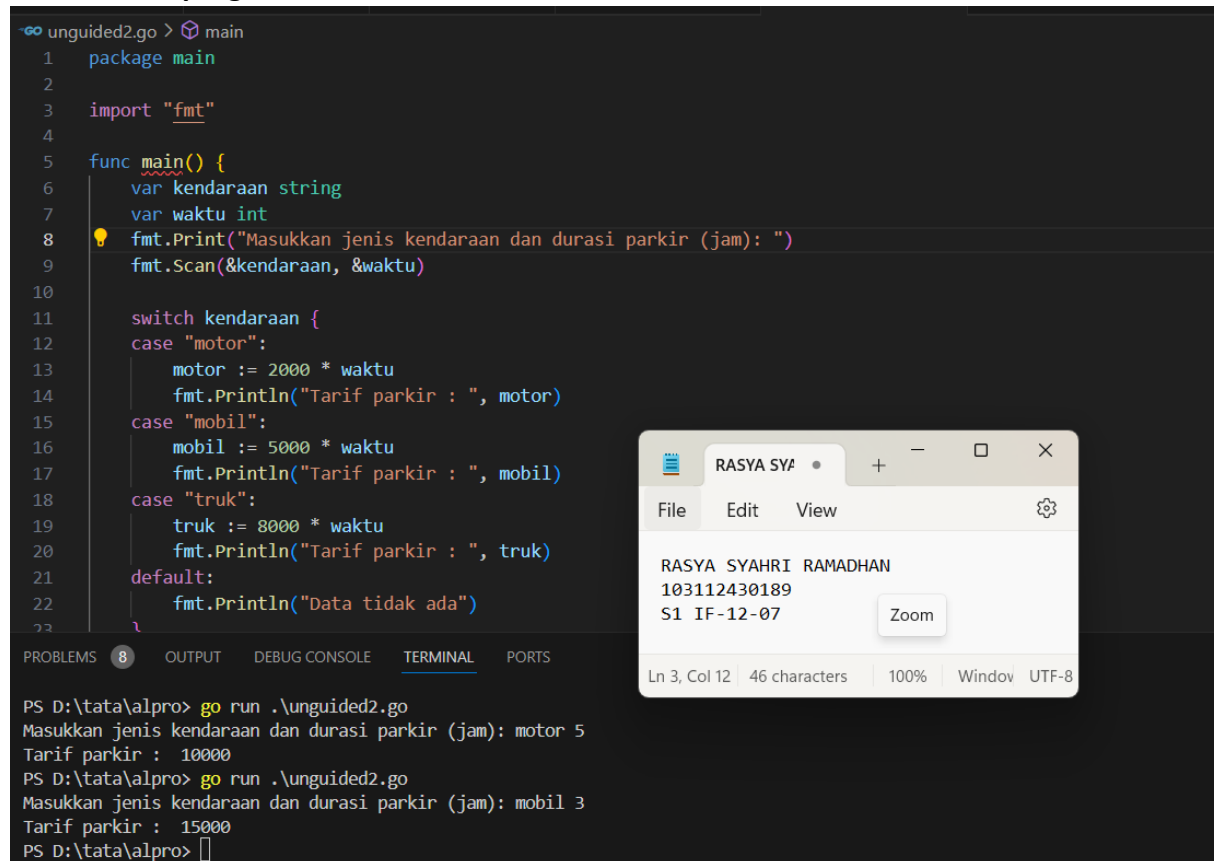
import "fmt"

func main() {
    var kendaraan string
    var waktu int

    fmt.Print("Masukkan jenis kendaraan dan durasi parkir (jam): ")
    fmt.Scan(&kendaraan, &waktu)

    switch kendaraan {
    case "motor":
        motor := 2000 * waktu
        fmt.Println("Tarif parkir : ", motor)
    case "mobil":
        mobil := 5000 * waktu
        fmt.Println("Tarif parkir : ", mobil)
    case "truk":
        truk := 8000 * waktu
        fmt.Println("Tarif parkir : ", truk)
    default:
        fmt.Println("Data tidak ada")
    }
}
```

Screenshoot program



The screenshot shows a Go program in a dark-themed IDE. The code defines a `main` function that prompts the user for vehicle type and parking duration, then calculates the parking fee based on a switch statement. The terminal output shows two successful runs: one for a motor vehicle with a 5-hour duration resulting in a fee of 10,000, and another for a car with a 3-hour duration resulting in a fee of 15,000. A third run is partially visible, showing the prompt for a truck.

```
unguided2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var waktu int
8     fmt.Print("Masukkan jenis kendaraan dan durasi parkir (jam): ")
9     fmt.Scan(&kendaraan, &waktu)
10
11     switch kendaraan {
12     case "motor":
13         motor := 2000 * waktu
14         fmt.Println("Tarif parkir : ", motor)
15     case "mobil":
16         mobil := 5000 * waktu
17         fmt.Println("Tarif parkir : ", mobil)
18     case "truk":
19         truk := 8000 * waktu
20         fmt.Println("Tarif parkir : ", truk)
21     default:
22         fmt.Println("Data tidak ada")
23     }
24 }
```

Terminal Output:

```
PS D:\tata\alpro> go run .\unguided2.go
Masukkan jenis kendaraan dan durasi parkir (jam): motor 5
Tarif parkir : 10000
PS D:\tata\alpro> go run .\unguided2.go
Masukkan jenis kendaraan dan durasi parkir (jam): mobil 3
Tarif parkir : 15000
PS D:\tata\alpro>
```

Deskripsi program

Program di atas adalah aplikasi sederhana dalam bahasa Go untuk menghitung tarif parkir berdasarkan jenis kendaraan dan durasi parkir. Program meminta pengguna untuk memasukkan **jenis kendaraan** (motor, mobil, atau truk) dan **durasi parkir** dalam satuan jam. Tarif dihitung berdasarkan jenis kendaraan dan durasi parkir, lalu hasilnya ditampilkan.

switch kendaraan: Memeriksa jenis kendaraan yang dimasukkan pengguna.

Setiap **case** menghitung tarif parkir berdasarkan jenis kendaraan:

- motor:**
 - Tarif per jam adalah Rp 2.000.
 - Total tarif dihitung dengan **2000 * waktu**.
- mobil:**
 - Tarif per jam adalah Rp 5.000.
 - Total tarif dihitung dengan **5000 * waktu**.
- truk:**
 - Tarif per jam adalah Rp 8.000.
 - Total tarif dihitung dengan **8000 * waktu**.
- Jika jenis kendaraan tidak cocok dengan salah satu kasus, program mencetak **Data tidak ada** di bagian **default**.

Tugas 3

Pseudo code

```
Program_Indentifikasi_Pola_Aritmatika
Mulai

Rumus
var angka int

Algoritma
output("Masukkan
angka: ")input(angka)

bagi := angka%10 == 0 && angka > 10
kuadrat := angka%5 == 0 && angka > 5
ganjil := angka%2 != 0
genap := angka%2 == 0

switch
case bagi: hasil := angka / 10
("Kategori = Bilangan Kelipatan 10")
("Hasil penjumlahan bilangan berikutnya %d / 10 = %d", angka, hasil)

case kuadrat: hasil := angka * angka
("Kategori = Bilangan Kelipatan 5")
("Hasil penjumlahan bilangan berikutnya %d^2 = %d", angka, hasil)

case ganjil: angka2 := angka + 1 hasil := angka + angka2
("Kategori = Bilangan Ganjil")
("Hasil penjumlahan bilangan berikutnya %d + %d = %d", angka, angka2, hasil)

case genap: angka2 := angka + 1 hasil := angka * angka2
("Kategori = Bilangan Genap")
("Hasil penjumlahan bilangan berikutnya %d * %d = %d", angka, angka2, hasil)

endprogram
```

Source code

```
package main

import "fmt"

func main() {
    var angka int
    fmt.Print("masukkan angka : ")
    fmt.Scan(&angka)
```

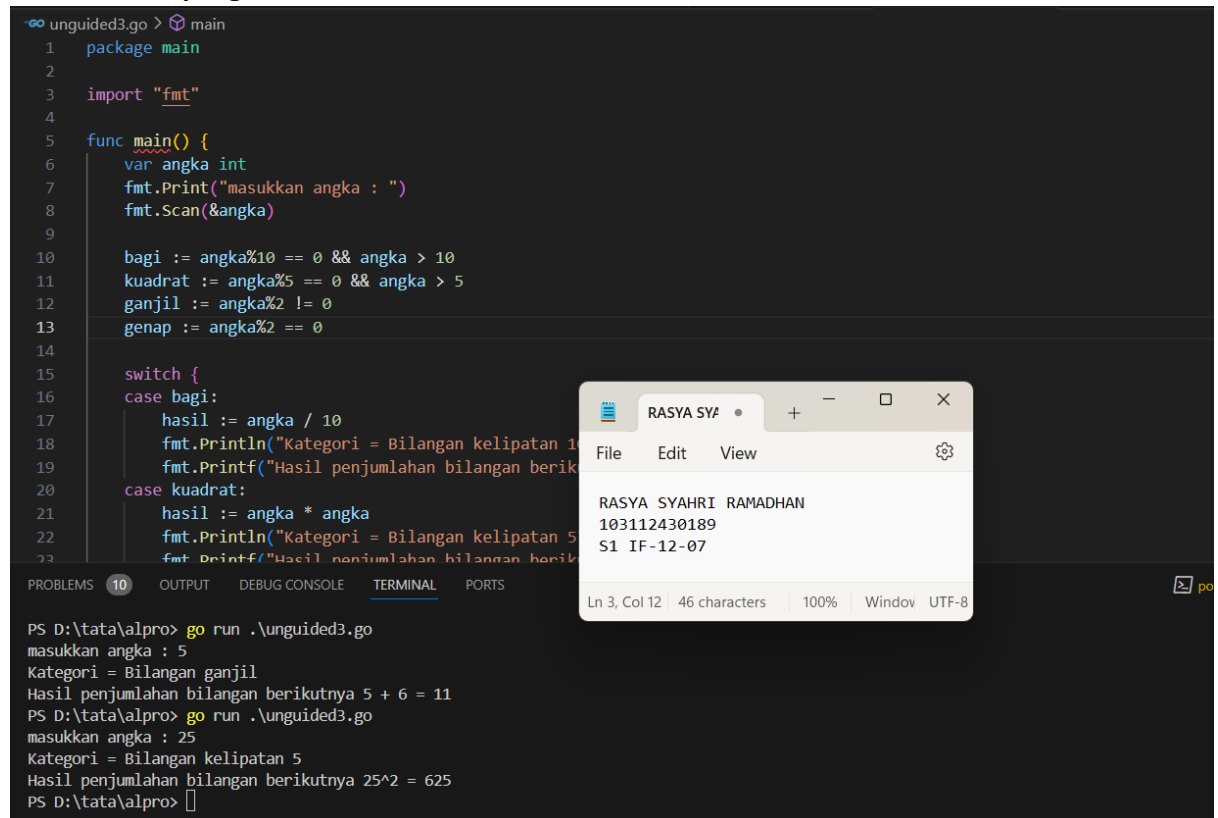
```

    bagi := angka%10 == 0 && angka > 10
    kuadrat := angka%5 == 0 && angka > 5
    ganjil := angka%2 != 0
    genap := angka%2 == 0

    switch {
    case bagi:
        hasil := angka / 10
        fmt.Println("Kategori = Bilangan kelipatan 10")
        fmt.Printf("Hasil penjumlahan bilangan berikutnya %d / 10 = %d", angka, hasil)
    case kuadrat:
        hasil := angka * angka
        fmt.Println("Kategori = Bilangan kelipatan 5")
        fmt.Printf("Hasil penjumlahan bilangan berikutnya %d^2 = %d", angka, hasil)
    case ganjil:
        angka2 := angka + 1
        hasil := angka + angka2
        fmt.Println("Kategori = Bilangan ganjil")
        fmt.Printf("Hasil penjumlahan bilangan berikutnya %d + %d = %d", angka, angka2, hasil)
    case genap:
        angka2 := angka + 1
        hasil := angka * angka2
        fmt.Println("Kategori = Bilangan genap")
        fmt.Printf("Hasil penjumlahan bilangan berikutnya %d * %d = %d", angka, angka2, hasil)
    }
}

```


Screenshoot program



The screenshot shows a Go program in a dark-themed IDE. The source code defines a `main` function that takes an integer `angka` and categorizes it based on divisibility by 10, 5, and parity. It also performs calculations: `angka / 10` for multiples of 10 and `angka * angka` for multiples of 5. The terminal output shows two test cases: input 5 results in 'Bilangan ganjil' and $5 + 6 = 11$; input 25 results in 'Bilangan kelipatan 5' and $25^2 = 625$. An overlaid window titled 'RASYA SYA' displays the text 'RASYA SYAHRI RAMADHAN', '103112430189', and 'S1 IF-12-07'.

```
unguided3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var angka int
7     fmt.Print("masukkan angka : ")
8     fmt.Scan(&angka)
9
10    bagi := angka%10 == 0 && angka > 10
11    kuadrat := angka%5 == 0 && angka > 5
12    ganjil := angka%2 != 0
13    genap := angka%2 == 0
14
15    switch {
16    case bagi:
17        hasil := angka / 10
18        fmt.Println("Kategori = Bilangan kelipatan 10")
19        fmt.Printf("Hasil penjumlahan bilangan berikutnya %d + 6 = %d\n", angka, angka+6)
20    case kuadrat:
21        hasil := angka * angka
22        fmt.Println("Kategori = Bilangan kelipatan 5")
23        fmt.Printf("Hasil penjumlahan bilangan berikutnya %d * %d = %d\n", angka, angka, angka*angka)
24    default:
25        if ganjil {
26            fmt.Println("Kategori = Bilangan ganjil")
27        } else {
28            fmt.Println("Kategori = Bilangan genap")
29        }
30    }
31}
```

PS D:\tata\alpro> go run .\unguided3.go
masukkan angka : 5
Kategori = Bilangan ganjil
Hasil penjumlahan bilangan berikutnya 5 + 6 = 11
PS D:\tata\alpro> go run .\unguided3.go
masukkan angka : 25
Kategori = Bilangan kelipatan 5
Hasil penjumlahan bilangan berikutnya 25^2 = 625
PS D:\tata\alpro>

Deskripsi program

Program di atas adalah aplikasi Go yang bertujuan untuk mengidentifikasi kategori bilangan berdasarkan pola tertentu (bilangan kelipatan 10, kelipatan 5, ganjil, atau genap) dan melakukan operasi matematika yang sesuai dengan kategori tersebut.

Urutan Pemeriksaan:

1. bagi diperiksa terlebih dahulu untuk menangani bilangan kelipatan 10 (karena bilangan kelipatan 10 juga memenuhi kategori kelipatan 5 dan genap).
2. kuadrat untuk bilangan kelipatan 5.
3. ganjil untuk bilangan ganjil.
4. genap sebagai kategori terakhir

Cara Kerja Program

1. Program meminta pengguna memasukkan bilangan bulat.
2. Berdasarkan bilangan yang dimasukkan, program mengidentifikasi kategori bilangan menggunakan pernyataan logika.
3. Program menjalankan operasi matematika sesuai dengan kategori bilangan.
4. Hasil operasi matematika dan kategori bilangan ditampilkan ke layar.

