

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 11



Disusun oleh:

FATIR GIBRAN

103112430153

S1IF-12-07

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

DASAR TEORI

11.1 Karakteristik Switch-Case

Pada dasarnya switch-case merupakan alternatif dari penggunaan else-if, tetapi pemilihan aksi tidak dilakukan berdasarkan kondisi, tetapi suatu nilai atau operasi tipe data dasar yang tidak menghasilkan boolean.

Penulisan switch case terdiri dari komponen berikut:

- a. Ekspresi, merupakan operasi tipe data yang menghasilkan nilai selain tipe data boolean,
- b. Nilai, pilihan case dari ekspresi yang dideklarasikan pada instruksi switch-case.
- c. Aksi, merupakan kumpulan instruksi yang akan dieksekusi sesuai dengan nilai yang dihasilkan dari ekspresi yang terdapat pada switch-case. Aksi lain hanya akan dieksekusi apabila tidak ada pilihan nilai yang sesuai dengan hasil ekspresi.

Kata kunci default digunakan untuk menyatakan pilihan untuk aksi lain. Selain itu terdapat juga penggunaan switch-case sebagai pengganti if-then, yang mana pada setiap case yang didefinisikan adalah kondisi dari struktur kontrol percabangan,

```
switch {  
case kondisi_1 : // aksi kondisi 1  
case kondisi_2 : // aksi kondisi 2  
case kondisi_2 : // aksi kondisi 3  
default : // aksi lain  
}
```

TUGAS PENDAHULUAN

1. Tugas Pendahuluan 1

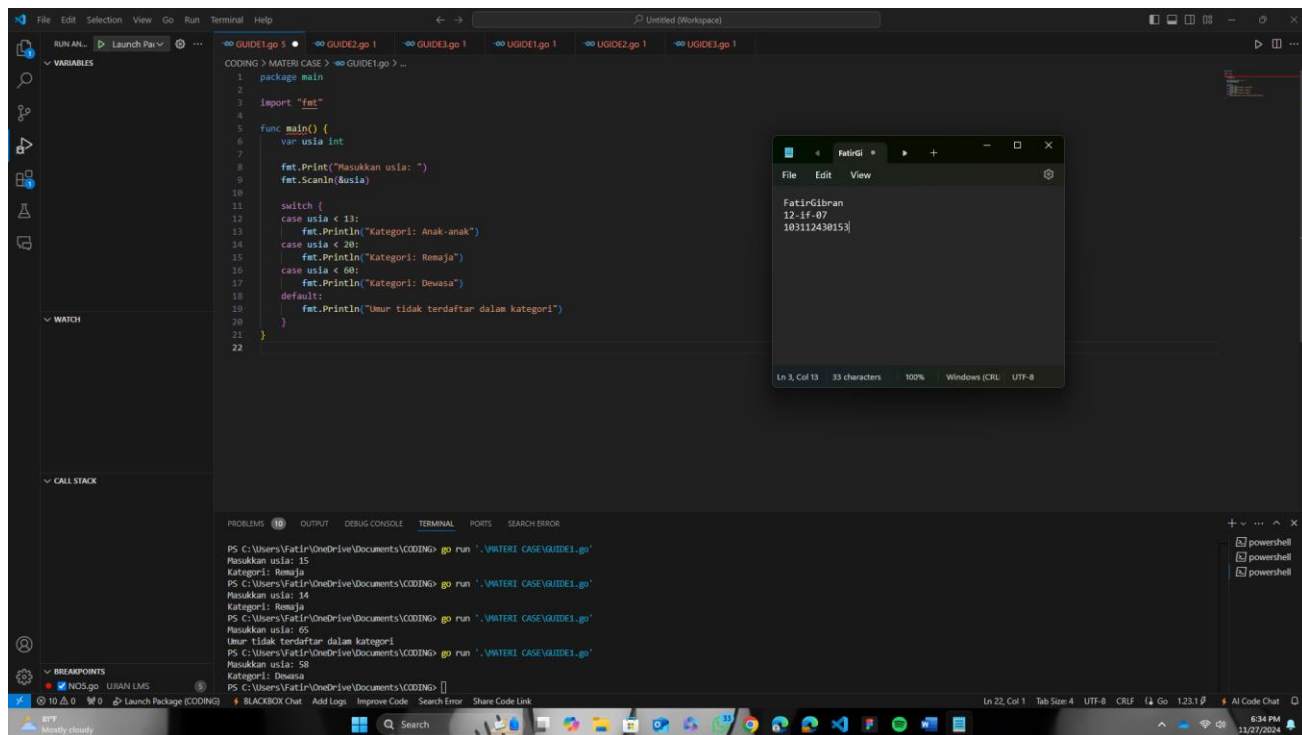
Pseudo code

```
MULAI  
  
memasukkan usia  
  
    JIKA usia < 13 MAKA  
        "Kategori: Anak-anak"  
  
    LAIN JIKA usia < 20 MAKA  
        Kategori: Remaja"  
  
    LAIN JIKA usia < 60 MAKA  
        "Kategori: Dewasa"  
  
    "Umur tidak terdaftar dalam kategori"  
  
    AKHIR JIKA  
  
AKHIR
```

SOURCES CODE

```
package main  
  
import "fmt"  
  
func main() {  
    var usia int  
  
    fmt.Print("Masukkan usia: ")  
    fmt.Scanln(&usia)  
  
    switch {  
    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("Umur tidak terdaftar dalam kategori")  
    }  
}
```

Screenshoot program



Deskripsi program

Program tersebut menjalankan tentang sebuah program yang dimana fungsi untuk membagi usia menjadi beberapa kategori: anak-anak, remaja, dan dewasa. Pengguna diminta memasukkan usia, lalu program akan menentukan kategori usia berdasarkan input tersebut.

2. Tugas Pendahuluan 2

Pseudo code

Mulai program.
Deklarasikan variabel hari untuk menyimpan input.
Minta pengguna memasukkan nomor hari.
Baca input dari pengguna dan simpan ke dalam variabel hari.
Periksa nilai hari:
Jika hari adalah 1, cetak "Senin".
Jika hari adalah 2, cetak "Selasa", dan seterusnya.
Jika hari tidak sesuai dengan nilai yang sudah ditentukan, cetak "Nomor hari tidak valid".
Akhir program

```
package main

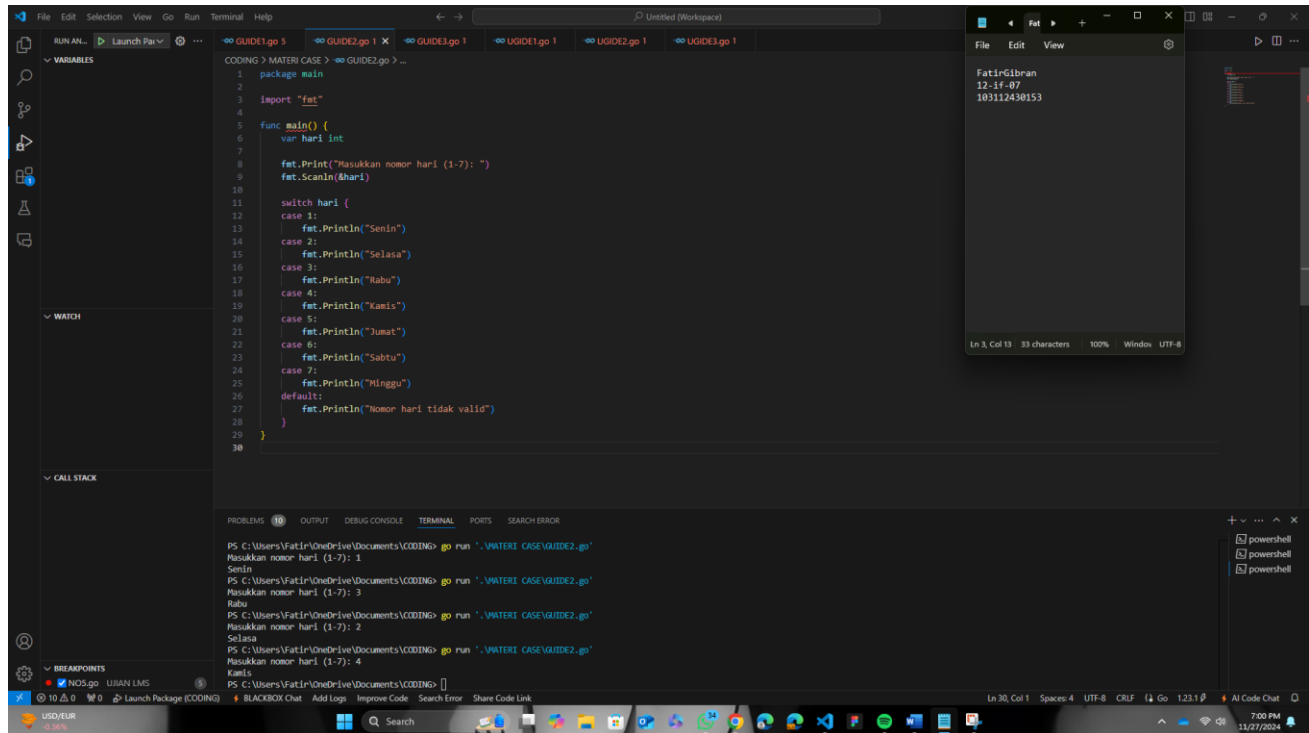
import "fmt"

func main() {
    var hari int

    fmt.Print("Masukkan nomor hari (1-7): ")
    fmt.Scanln(&hari)

    switch hari {
    case 1:
        fmt.Println("Senin")
    case 2:
        fmt.Println("Selasa")
    case 3:
        fmt.Println("Rabu")
    case 4:
        fmt.Println("Kamis")
    case 5:
        fmt.Println("Jumat")
    case 6:
        fmt.Println("Sabtu")
    case 7:
        fmt.Println("Minggu")
    default:
        fmt.Println("Nomor hari tidak valid")
    }
}
```

Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var hari int
7
8     fmt.Print("Masukkan nomor hari (1-7): ")
9     fmt.Scanln(&hari)
10
11     switch hari {
12     case 1:
13         fmt.Println("Senin")
14     case 2:
15         fmt.Println("Selasa")
16     case 3:
17         fmt.Println("Rabu")
18     case 4:
19         fmt.Println("Kamis")
20     case 5:
21         fmt.Println("Jumat")
22     case 6:
23         fmt.Println("Sabtu")
24     case 7:
25         fmt.Println("Minggu")
26     default:
27         fmt.Println("Nomor hari tidak valid")
28     }
29 }
30
```

Terminal Output:

```
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\VATARI CASE\GUIDE2.go'
Masukkan nomor hari (1-7): 1
Senin
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\VATARI CASE\GUIDE2.go'
Masukkan nomor hari (1-7): 3
Rabu
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\VATARI CASE\GUIDE2.go'
Masukkan nomor hari (1-7): 2
Selasa
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\VATARI CASE\GUIDE2.go'
Masukkan nomor hari (1-7): 4
Kamis
PS C:\Users\Fatir\OneDrive\Documents\CODING>
```

Small Window Output:

```
FatirGibran
12-if-07
103112430153
```

Deskripsi program

Program tersebut berfungsi mengubah nomor hari (1-7) menjadi nama hari dalam seminggu. Dengan kata lain, program ini berfungsi sebagai konverter angka menjadi nama hari. Jadi pada inputan berupa int yaitu angka dan string sebagai keterangan.

3. Tugas Pendahuluan 3

Pseudo code

```
Mulai

  Minta pengguna memasukkan kode film (G, PG, PG-13, R)
  Simpan kode film ke dalam variabel kodeFilm

  Jika kodeFilm adalah "G" maka
    Tampilkan "Semua usia"
  Jika kodeFilm adalah "PG" maka
    Minta pengguna memasukkan usia
    Jika usia lebih besar dari 7 maka
      Tampilkan "Boleh menonton"
    Lainnya
      Tampilkan "Terlalu muda"
  Jika kodeFilm adalah "PG-13" maka
    Minta pengguna memasukkan usia
    Jika usia lebih besar dari 13 maka
      Tampilkan "Boleh menonton"
    Lainnya
      Tampilkan "Terlalu muda"
  Jika kodeFilm adalah "R" maka
    Tampilkan "Dewasa saja"
  Jika tidak ada kode film yang cocok maka
    Tampilkan "Kode film tidak valid"

Selesai
```

SOURCES CODE

```
package main

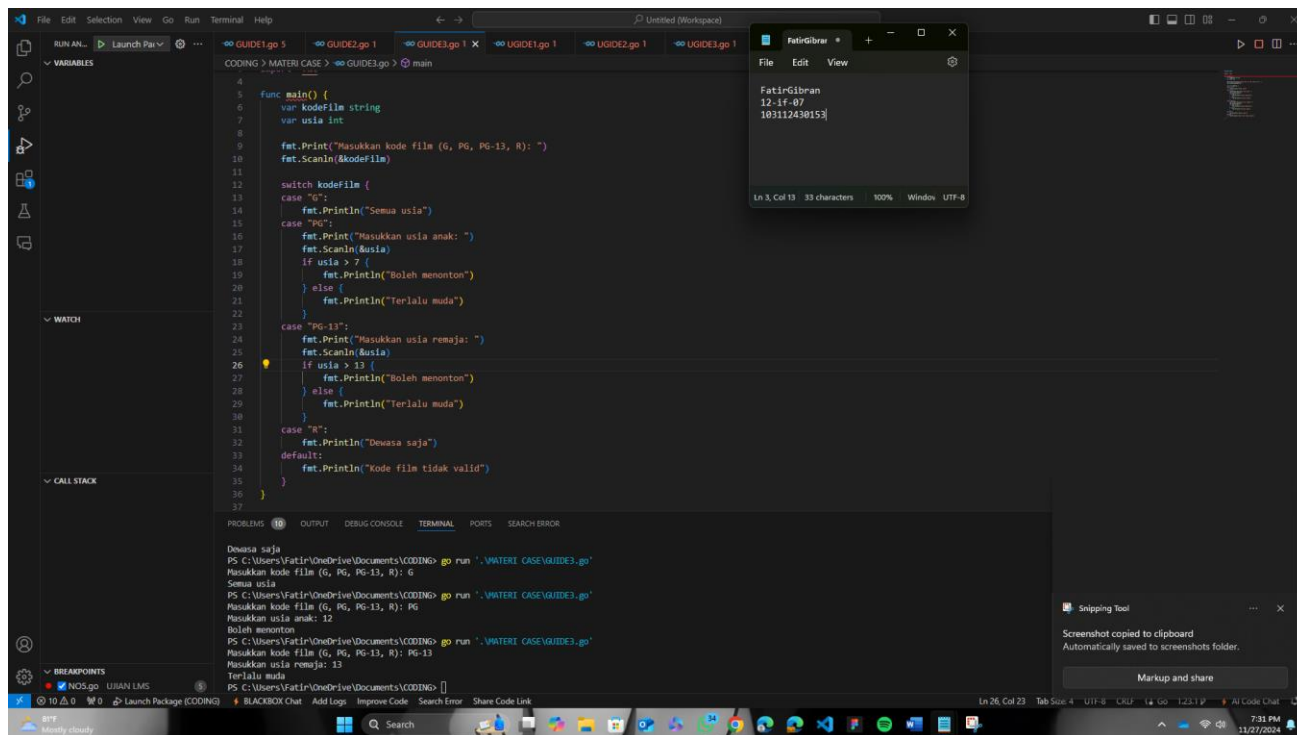
import "fmt"

func main() {
    var kodeFilm string
    var usia int

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

    switch kodeFilm {
    case "G":
        fmt.Println("Semua usia")
    case "PG":
        fmt.Print("Masukkan usia anak: ")
        fmt.Scanln(&usia)
        if usia > 7 {
            fmt.Println("Boleh menonton")
        } else {
            fmt.Println("Terlalu muda")
        }
    case "PG-13":
        fmt.Print("Masukkan usia remaja: ")
        fmt.Scanln(&usia)
        if usia > 13 {
            fmt.Println("Boleh menonton")
        } else {
            fmt.Println("Terlalu muda")
        }
    case "R":
        fmt.Println("Dewasa saja")
    default:
        fmt.Println("Kode film tidak valid")
    }
}
```


SCREENSHOOT PROGRAM



The screenshot shows a Go IDE with a file named `GUIDE3.go` containing the following code:

```
4 func main() {
5     var kodeFilm string
6     var usia int
7
8     fmt.Print("Masukkan kode film (G, PG, PG-13, R): ")
9     fmt.Scanln(&kodeFilm)
10
11     switch kodeFilm {
12     case "G":
13         fmt.Println("Semua usia")
14     case "PG":
15         fmt.Print("Masukkan usia anak: ")
16         fmt.Scanln(&usia)
17         if usia > 7 {
18             fmt.Println("Boleh menonton")
19         } else {
20             fmt.Println("Terlalu muda")
21         }
22     case "PG-13":
23         fmt.Print("Masukkan usia remaja: ")
24         fmt.Scanln(&usia)
25         if usia > 13 {
26             fmt.Println("Boleh menonton")
27         } else {
28             fmt.Println("Terlalu muda")
29         }
30     case "R":
31         fmt.Println("Dewasa saja")
32     default:
33         fmt.Println("Kode film tidak valid")
34     }
35 }
36
37
```

The terminal output shows the execution of the program with the following input and output:

```
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\MATERI CASE\GUIDE3.go'
Masukkan kode film (G, PG, PG-13, R): G
Semua usia
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\MATERI CASE\GUIDE3.go'
Masukkan kode film (G, PG, PG-13, R): PG
Masukkan usia anak: 12
Boleh menonton
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\MATERI CASE\GUIDE3.go'
Masukkan kode film (G, PG, PG-13, R): PG-13
Masukkan usia remaja: 13
Boleh menonton
PS C:\Users\Fatir\OneDrive\Documents\CODING>
```

A small window titled `FatirGibran` is open, displaying the text:

```
FatirGibran
12-11-07
103112430153
```

The status bar at the bottom indicates the file is at line 26, column 23, with a tab size of 4, UTF-8 encoding, and a Go version of 1.22.1 p.

Deskripsi program

Program ini berfungsi untuk menentukan apakah seseorang boleh menonton film berdasarkan ratingnya.

Cara kerjanya:

Program meminta pengguna memasukkan kode rating film (G, PG, PG-13, atau R).

Jika diperlukan, program juga akan meminta pengguna memasukkan usia.

Program kemudian membandingkan kode rating dengan usia penonton.

Terakhir, program akan memberikan keputusan apakah penonton boleh menonton film tersebut atau tidak.

Intinya, program ini seperti seorang penjaga pintu bioskop yang memeriksa tiket dan kartu identitas untuk memastikan penonton sesuai dengan rating film.

LATIHAN MODUL

1. Tugas 1

Pseudo code

Mulai

Minta pengguna memasukkan nilai pH

Baca nilai pH dan simpan ke dalam variabel pH

Jika pH kurang dari 0 atau lebih dari 14 maka

Tampilkan "Input tidak valid, rentang pH 0 - 14"

Jika pH lebih besar sama dengan 6.5 dan kurang dari sama dengan 8.6 maka

Tampilkan "Air Layak Minum"

Lainnya

Tampilkan "Air Tidak Layak Minum"

Selesai

Source code

```
package main

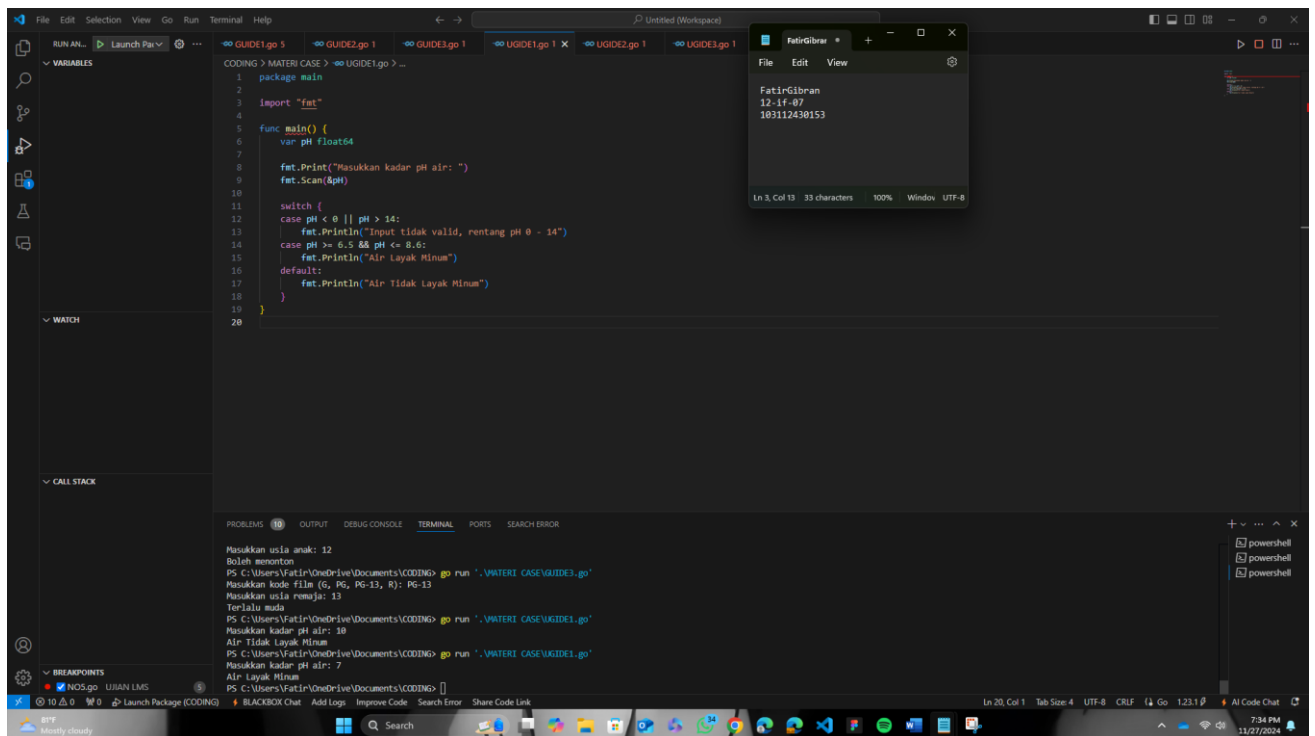
import "fmt"

func main() {
    var pH float64

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



Deskripsi program

Program tersebut berfungsi untuk membantu kita menentukan apakah air yang kita punya aman untuk diminum dengan cara memeriksa nilai pH-nya. Dengan inputan yang diberikan dan dengan aturan pH yang disesuaikan range dari air yang layak dan tak layak.

2.Tugas 2

Pseudo code

```
Mulai
Minta pengguna untuk memasukkan jenis kendaraan
Simpan jenis kendaraan ke dalam variabel kendaraan
Minta pengguna untuk memasukkan durasi parkir dalam jam
Simpan durasi parkir ke dalam variabel durasi

Jika durasi parkir kurang dari 1 jam, maka tetapkan durasi parkir menjadi 1 jam

Berdasarkan jenis kendaraan, tentukan tarif parkir:
  Jika kendaraan adalah motor, maka tarif parkir adalah 2000
  Jika kendaraan adalah mobil, maka tarif parkir adalah 5000
  Jika kendaraan adalah truk, maka tarif parkir adalah 8000
  Jika jenis kendaraan tidak valid, tampilkan pesan kesalahan dan akhiri program

Hitung total biaya parkir dengan mengalikan tarif parkir dengan durasi parkir
Tampilkan total biaya parkir kepada pengguna
Akhir
```

Source code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi float64

    fmt.Print("Masukkan jenis kendaraan (motor, mobil, truk): ")
    fmt.Scan(&kendaraan)
    fmt.Print("Masukkan durasi parkir (jam): ")
    fmt.Scan(&durasi)

    if durasi < 1 {
        durasi = 1
    }

    var tarif float64
    switch kendaraan {
    case "motor":
        tarif = 2000
    case "mobil":
        tarif = 5000
    case "truk":
        tarif = 8000
    default:
        fmt.Println("Jenis kendaraan tidak valid")
        return
    }

    totalBiaya := tarif * durasi
    fmt.Printf("Total biaya parkir: Rp %d\n", int(totalBiaya))
}
```

Screenshoot program

The screenshot displays a Go IDE with a workspace named 'Untitled (Workspace)'. The main editor shows a Go file named 'CODING > MATERI CASE > UGIDE1.go' with the following code:

```
1 func main() {
2     var kendaraan string
3     var durasi float64
4
5     fmt.Print("Masukkan jenis kendaraan (motor, mobil, truk): ")
6     fmt.Scan(&kendaraan)
7
8     fmt.Print("Masukkan durasi parkir (jam): ")
9     fmt.Scan(&durasi)
10
11     if durasi < 1 {
12         durasi = 1
13     }
14
15     var tarif float64
16     switch kendaraan {
17     case "motor":
18         tarif = 2000
19     case "mobil":
20         tarif = 5000
21     case "truk":
22         tarif = 8000
23     default:
24         fmt.Println("Jenis kendaraan tidak valid")
25         return
26     }
27
28     totalBiaya := tarif * durasi
29     fmt.Printf("Total biaya parkir: Rp %d\n", int(totalBiaya))
30 }
31
```

A 'FatirGibran' window is open, showing the user's name and ID: 'FatirGibran', '12-11-07', '183112430153'. The terminal at the bottom shows the execution output:

```
PS C:\Users\FatirGibran> go run '.\MATERI CASE\UGIDE1.go'
Masukkan kadar parkir: 7
Air Tidak Layak Minum
PS C:\Users\FatirGibran> go run '.\MATERI CASE\UGIDE2.go'
Masukkan jenis kendaraan (motor, mobil, truk): motor
Masukkan durasi parkir (jam): 9
Total biaya parkir: Rp 18000
PS C:\Users\FatirGibran> go run '.\MATERI CASE\UGIDE2.go'
Masukkan jenis kendaraan (motor, mobil, truk): truk
Masukkan durasi parkir (jam): 12
Total biaya parkir: Rp 96000
PS C:\Users\FatirGibran> go run '.\MATERI CASE\UGIDE2.go'
```

Deskripsi program

Program ini dapat digunakan sebagai dasar untuk membangun sistem parkir yang lebih kompleks, misalnya:

- **Sistem parkir otomatis:** Program dapat diintegrasikan dengan sensor kendaraan untuk secara otomatis mendeteksi jenis kendaraan dan durasi parkir.
- **Sistem pembayaran parkir:** Program dapat dihubungkan dengan sistem pembayaran elektronik untuk memfasilitasi pembayaran parkir.
- **Sistem manajemen parkir:** Program dapat digunakan untuk mengelola data parkir, seperti jumlah kendaraan yang parkir, pendapatan parkir, dan sebagainya.

3.Tugas 3

Pseudo code

Mulai

Minta pengguna memasukkan sebuah bilangan bulat

Simpan bilangan tersebut ke dalam variabel bilangan

Jika bilangan dibagi 2 sisanya 1 (bilangan ganjil) maka

Hitung hasil penjumlahan bilangan dengan bilangan berikutnya

Tampilkan kategori "Bilangan Ganjil" dan hasil penjumlahan

Jika bilangan dibagi 2 sisanya 0 (bilangan genap) maka

Hitung hasil perkalian bilangan dengan bilangan berikutnya

Tampilkan kategori "Bilangan Genap" dan hasil perkalian

Jika bilangan dibagi 5 sisanya 0 (kelipatan 5) maka

Hitung kuadrat dari bilangan

Tampilkan kategori "Bilangan Kelipatan 5" dan hasil kuadrat

Jika bilangan dibagi 10 sisanya 0 (kelipatan 10) maka

Hitung hasil pembagian bilangan dengan 10

Tampilkan kategori "Bilangan Kelipatan 10" dan hasil pembagian

Akhir

Source code

```
package main

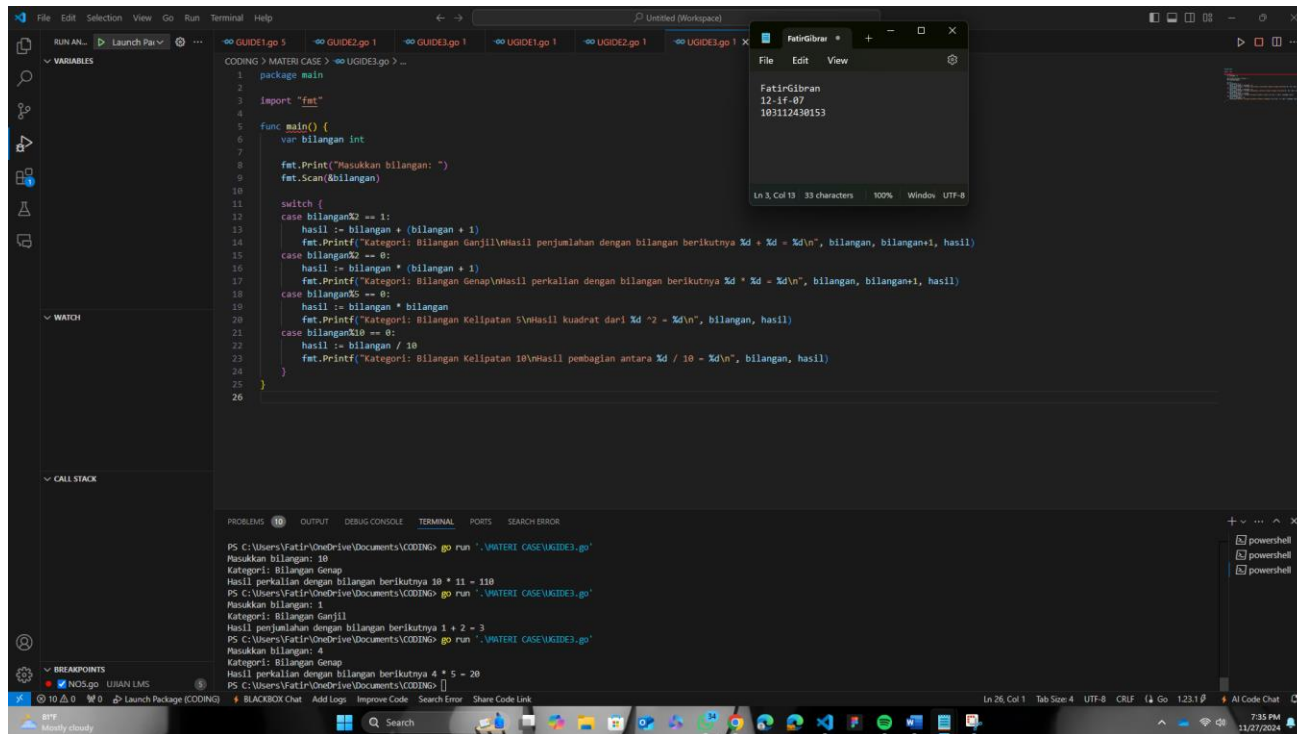
import "fmt"

func main() {
    var bilangan int

    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&bilangan)

    switch {
    case bilangan%2 == 1:
        hasil := bilangan + (bilangan + 1)
        fmt.Printf("Kategori: Bilangan Ganjil\nHasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", bilangan, bilangan+1, hasil)
    case bilangan%2 == 0:
        hasil := bilangan * (bilangan + 1)
        fmt.Printf("Kategori: Bilangan Genap\nHasil perkalian dengan bilangan berikutnya %d * %d = %d\n", bilangan, bilangan+1, hasil)
    case bilangan%5 == 0:
        hasil := bilangan * bilangan
        fmt.Printf("Kategori: Bilangan Kelipatan 5\nHasil kuadrat dari %d ^2 = %d\n", bilangan, hasil)
    case bilangan%10 == 0:
        hasil := bilangan / 10
        fmt.Printf("Kategori: Bilangan Kelipatan 10\nHasil pembagian antara %d / 10 = %d\n", bilangan, hasil)
    }
}
```

Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7
8     fmt.Print("Masukkan bilangan: ")
9     fmt.Scan(&bilangan)
10
11     switch {
12     case bilangan%2 == 1:
13         hasil := bilangan + (bilangan + 1)
14         fmt.Printf("Kategori: Bilangan Ganjil\ Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", bilangan, bilangan+1, hasil)
15     case bilangan%2 == 0:
16         hasil := bilangan * (bilangan + 1)
17         fmt.Printf("Kategori: Bilangan Genap\ Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", bilangan, bilangan+1, hasil)
18     case bilangan%5 == 0:
19         hasil := bilangan * bilangan
20         fmt.Printf("Kategori: Bilangan Kelipatan 5\ Hasil kuadrat dari %d ^2 = %d\n", bilangan, hasil)
21     case bilangan%10 == 0:
22         hasil := bilangan / 10
23         fmt.Printf("Kategori: Bilangan Kelipatan 10\ Hasil pembagian antara %d / 10 = %d\n", bilangan, hasil)
24     }
25 }
26
```

Terminal Output:

```
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\MATERI CASE\UGIDE3.go'
Masukkan bilangan: 10
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 10 * 11 = 110
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\MATERI CASE\UGIDE3.go'
Masukkan bilangan: 1
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan berikutnya 1 + 2 = 3
PS C:\Users\Fatir\OneDrive\Documents\CODING> go run '.\MATERI CASE\UGIDE3.go'
Masukkan bilangan: 4
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 4 * 5 = 20
PS C:\Users\Fatir\OneDrive\Documents\CODING>
```

Deskripsi program

Kode di atas dirancang untuk mengklasifikasikan bilangan bulat berdasarkan karakteristik tertentu dan melakukan operasi matematika yang sesuai.

Cara Kerja:

1. **Input Bilangan:** Program meminta pengguna untuk memasukkan sebuah bilangan bulat.
2. **Pemeriksaan Kondisi:** Bilangan yang dimasukkan akan diperiksa terhadap beberapa kondisi:
 - **Bilangan Ganjil:** Jika bilangan habis dibagi 2 dan sisanya 1, maka bilangan tersebut adalah bilangan ganjil. Program kemudian menjumlahkan bilangan tersebut dengan bilangan berikutnya dan menampilkan hasilnya.
 - **Bilangan Genap:** Jika bilangan habis dibagi 2, maka bilangan tersebut adalah bilangan genap. Program kemudian mengalikan bilangan tersebut dengan bilangan berikutnya dan menampilkan hasilnya.
 - **Kelipatan 5:** Jika bilangan habis dibagi 5, maka bilangan tersebut adalah kelipatan 5. Program kemudian mengkuadratkan bilangan tersebut dan menampilkan hasilnya.
 - **Kelipatan 10:** Jika bilangan habis dibagi 10, maka bilangan tersebut adalah kelipatan 10. Program kemudian membagi bilangan tersebut dengan 10 dan menampilkan hasilnya.