

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 11
SWITCH CASE



Disusun oleh:

TIO ARMANI

103112430225

12-IF-07

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

DASAR TEORI

Switch case adalah struktur kontrol percabangan yang digunakan untuk memilih salah satu dari beberapa blok kode untuk dieksekusi, berdasarkan evaluasi suatu ekspresi. Switch pada Golang dirancang dengan sintaks yang lebih sederhana dan aman dibandingkan dengan implementasi pada beberapa bahasa pemrograman lainnya dan juga memungkinkan sebuah program untuk mencocokkan hasil evaluasi ekspresi (atau kondisi) dengan satu atau lebih case, kemudian mengeksekusi blok kode yang sesuai. Ini adalah bentuk kontrol bercabang yang lebih terstruktur dibandingkan dengan rangkaian if-else biasa.

TUGAS PENDAHULUAN

1. Tugas 1

Pseudo code

```
Program Kategori  
  
Kamus  
  
var usia : int  
  
Algoritma :  
  
output ("Masukan Usia : ")  
  
input (usia)  
  
depend on usia  
  
usia < 13:  
output ("Anak-anak")  
  
usia < 20:  
output ("Remaja")  
  
usia < 60:  
output ("Dewasa")  
  
usia >= 60:  
output ("Lansia")  
  
default:  
output ("Tidak terdaftar dalam kategori")  
  
end  
  
endprogram
```

Source code

```
package main

import "fmt"

func main() {

    var usia int

    fmt.Println("Masukan Usia :")
    fmt.Scan(&usia)

    switch {
    case usia < 13:
        fmt.Println("Anak-Anak")
    case usia < 20:
        fmt.Println("Remaja")
    case usia < 60:
        fmt.Println("Dewasa")
    case usia >= 60:
        fmt.Println("Lansia")
    default:
        fmt.Println("Umur tidak terdaftar dalam kategori")
    }
}
```

Screenshoot program

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

*Untitled - Note...
File Edit Format View Help
Nama : Tio Armani
NIM : 103112430225
Kelas : 12-IF-07
100 Windows (CRLF) UTF-8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

```
PS C:\alpro> go run "c:\alpro\guided1.go"
Masukan Usia :
2
Anak-Anak
PS C:\alpro> go run "c:\alpro\guided1.go"
Masukan Usia :
100
Lansia
PS C:\alpro>
```

Deskripsi program

Program ini bertujuan mengelompokkan usia seseorang ke dalam kategori tertentu berdasarkan input yang dimasukkan. Program mulai dengan meminta pengguna untuk memasukkan usia mereka. Input tersebut kemudian disimpan dalam variabel bernama `usia`.

Program menggunakan struktur `switch` pengelompokan usia. Kondisinya adalah sebagai berikut:

- Jika usia kurang dari 13, program akan mencetak "Anak-Anak."
- Jika usia di antara 13 dan kurang dari 20, akan mencetak "Remaja."
- Jika usia di antara 20 dan kurang dari 60, akan mencetak "Dewasa."
- Jika usia 60 atau lebih, akan mencetak "Lansia."

Lalu Jika ada kondisi yang tidak terduga atau usia tidak cocok dengan kategori apa pun, program akan mencetak "Umur tidak terdaftar dalam kategori.". Kesimpulannya, program ini membantu mengelompokkan usia berdasarkan tahap kehidupan, mulai dari anak-anak hingga lansia

2. Tugas 2

Pseudo code

Program Hari

Kamus

var angka :**int**

output ("Masukan angka : ")

input (angka)

depend on angka {

case 1:

output ("senin")

case 2:

output ("Selasa")

case 3:

output ("Rabu")

case 4:

output ("Kamis")

case 5:

output ("Jumat")

case 6:

output ("Sabtu")

case 7:

output ("Minggu")

end

endprogram

Source code

```
package main

import "fmt"

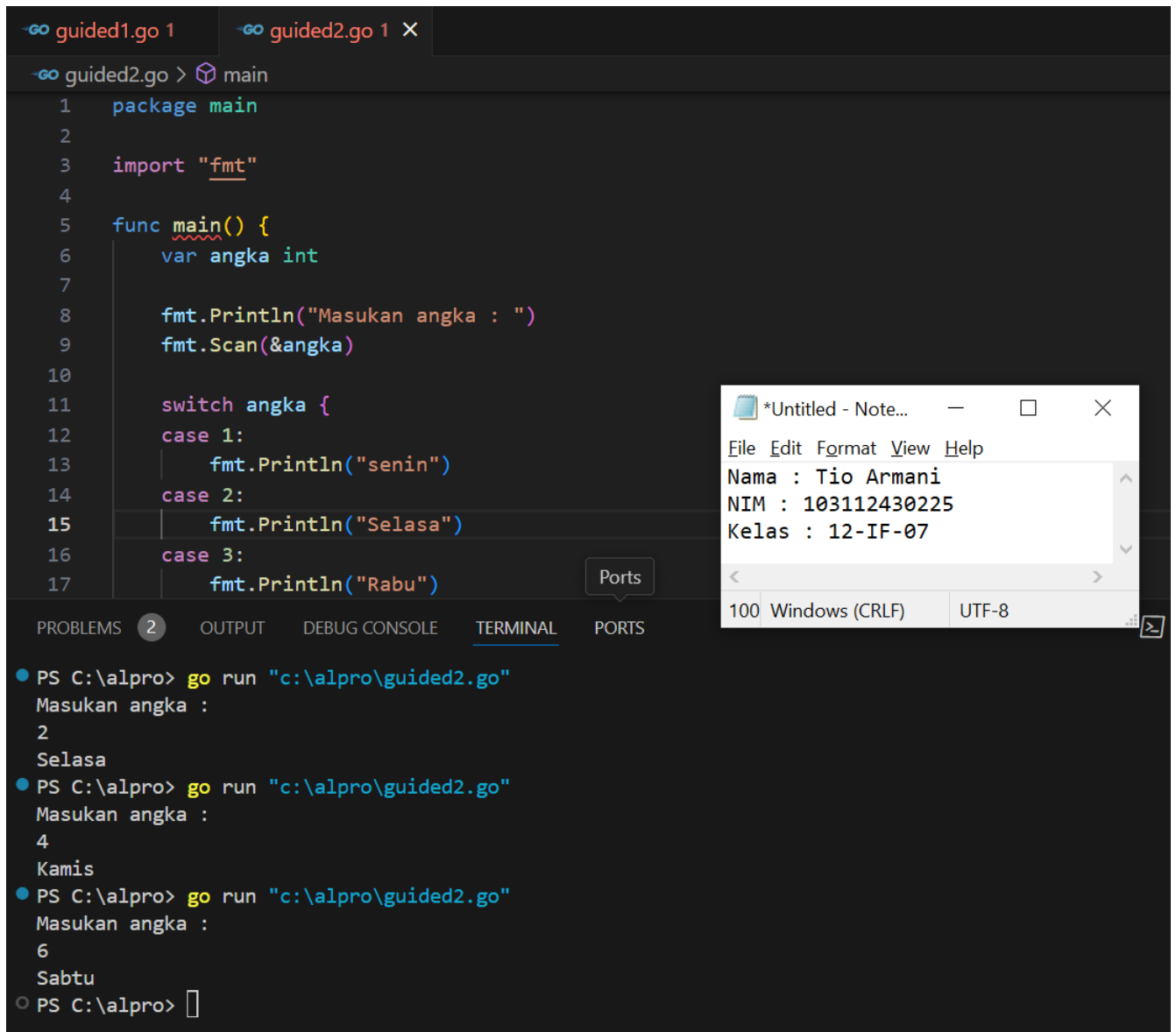
func main() {
    var angka int

    fmt.Println("Masukan angka : ")
    fmt.Scan(&angka)

    switch angka {
    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")

    }
}
```

Screenshoot program



```
guided2.go 1 X
guided2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var angka int
7
8     fmt.Println("Masukan angka : ")
9     fmt.Scan(&angka)
10
11     switch angka {
12     case 1:
13         fmt.Println("senin")
14     case 2:
15         fmt.Println("Selasa")
16     case 3:
17         fmt.Println("Rabu")
18     }
19 }

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\alpro> go run "c:\alpro\guided2.go"
Masukan angka :
2
Selasa
PS C:\alpro> go run "c:\alpro\guided2.go"
Masukan angka :
4
Kamis
PS C:\alpro> go run "c:\alpro\guided2.go"
Masukan angka :
6
Sabtu
PS C:\alpro>
```

*Untitled - Note...
File Edit Format View Help
Nama : Tio Armani
NIM : 103112430225
Kelas : 12-IF-07

Deskripsi program

Program ini dibuat untuk menampilkan nama hari dalam seminggu berdasarkan angka yang dimasukkan oleh pengguna. Program meminta pengguna untuk memasukkan sebuah angka melalui terminal. Angka ini kemudian disimpan di variabel angka.

Fungsi Switch untuk Menentukan Hari:

- Program ini menggunakan struktur switch untuk mencocokkan angka yang dimasukkan dengan daftar hari dalam seminggu.
- Kalau angka yang dimasukkan adalah:
 - 1, program akan mencetak "Senin."
 - 2, program akan mencetak "Selasa."
 - 3, program akan mencetak "Rabu."
 - 4, program akan mencetak "Kamis."
 - 5, program akan mencetak "Jumat."
 - 6, program akan mencetak "Sabtu."
 - 7, program akan mencetak "Minggu."

Kalau angka yang dimasukkan tidak sesuai dengan salah satu misalnya angka lebih dari 7 atau kurang dari 1, program tidak memberikan respons apa-apa karena tidak ada blok default yang menangani kasus di luar daftar tersebut.

3. Tugas 3

Pseudo code

```
Program Kode  
Kamus  
Var  
kode: string  
output ("Masukan Kode")  
input (&kode)  
depend on:  
case kode == "G":  
  output ("Semua usia")  
case kode == "PG":  
  output ("Anak-anak di atas 7 tahun")  
case kode == "PG13":  
  output ("Anak-anak di atas 13 tahun")  
case kode == "R":  
  output ("Dewasa")  
default:  
  output ("Kode tidak valid")  
end  
endprogram
```


Source code

```
package main

import "fmt"

func main() {
    var kode string

    fmt.Println("Masukan Kode")
    fmt.Scan(&kode)

    switch {
    case kode == "G":
        fmt.Println("Semua usia")
    case kode == "PG":
        fmt.Println("Anak-anak di atas 7 tahun")
    case kode == "PG13":
        fmt.Println("Anak-anak di atas 13 tahun")
    case kode == "R":
        fmt.Println("Dewasa")
    default:
        fmt.Println("Kode tidak valid")
    }
}
```

Screenshoot program

The screenshot shows a Go program in a code editor and its execution in the terminal. The code is a simple switch statement that checks for input codes and prints corresponding messages. The terminal shows three successful runs with different inputs: 'G', 'PG13', and 'R'.

```
guided3.go > main
5 func main() {
7
8     fmt.Println("Masukan Kode")
9     fmt.Scan(&kode)
10
11     switch {
12     case kode == "G":
13         fmt.Println("Semua usia")
14     case kode == "PG":
15         fmt.Println("Anak-anak di atas 7 tahun")
16     case kode == "PG13":
17         fmt.Println("Anak-anak di atas 13 tahun")
18     case kode == "R":
19         fmt.Println("Dewasa")
20     default:
21         fmt.Println("Kode tidak valid")
22     }
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Semua usia
● PS C:\alpro> go run "c:\alpro\guided3.go"
Masukan Kode
PG
Anak-anak di atas 7 tahun
● PS C:\alpro> go run "c:\alpro\guided3.go"
Masukan Kode
PG13
Anak-anak di atas 13 tahun
● PS C:\alpro> go run "c:\alpro\guided3.go"
Masukan Kode
R
Dewasa
○ PS C:\alpro>

*Untitled - Note...
File Edit Format View Help
Nama : Tio Armani
NIM : 103112430225
Kelas : 12-IF-07

100 Windows (CRLF) UTF-8

Deskripsi program

Program ngecek kode yang dimasukkan lewat struktur switch. Setiap kode punya arti masing-masing:

- Kalau kode-nya "G", program bakal cetak "Semua usia." Artinya, film atau konten aman buat siapa aja.
- Kalau kode-nya "PG", program bilang "Anak-anak di atas 7 tahun." Ini buat konten yang perlu pengawasan orang tua buat anak-anak kecil.
- Kalau kode-nya "PG13", program cetak "Anak-anak di atas 13 tahun." Maksudnya, konten ini cocok buat remaja atau lebih tua.
- Kalau kode-nya "R", program kasih tahu "Dewasa." Artinya, konten cuma buat orang dewasa.
- Kalau kode yang dimasukkan nggak sesuai sama yang diharapkan, program bakal cetak "Kode tidak valid."

Program ini buat ngecek arti dari kode rating, terutama yang biasa dipakai buat film atau konten digital. Tapi, pastikan kode yang dimasukkan sesuai (kayak "G", "PG", "PG13", atau "R"), kalau nggak, bakal dibilang nggak valid.

LATIHAN MODUL

1. Tugas 1

Pseudo code

```
Program Menghitung pH
var karpH: float64
input (&karpH)
depend on:
case karpH >= 6.5 && karpH <= 8.6:
  output ("Air layak minum")
case karpH < 6.5 || karpH > 8.6 && karpH < 14:
  output ("Air tidak layak minum")
case karpH > 14:
  output ("Nilai pH tidak valid, Nilai harus antara 0 dan 14 ")
end
endprogram
```

Source code

```
package main

import "fmt"

func main() {
    var kadarpH float64

    fmt.Scan(&kadarpH)

    switch {
    case kadarpH >= 6.5 && kadarpH <= 8.6:
        fmt.Print("Air layak minum")
    case kadarpH < 6.5 || kadarpH > 8.6 && kadarpH < 14:
        fmt.Print("Air tidak layak minum")
    case kadarpH > 14:
        fmt.Print("Nilai pH tidak valid, Nilai harus antara 0 dan 14 ")
    }
}
```

Screenshoot program

```
3  import "fmt"
4
5  func main() {
6      var kadarpH float64
7
8      fmt.Scan(&kadarpH)
9
10     switch {
11     case kadarpH >= 6.5 && kadarpH <= 8.6:
12         fmt.Print("Air layak minum")
13     case kadarpH < 6.5 || kadarpH > 8.6 && kadarpH < 14:
14         fmt.Print("Air tidak layak minum")
15     case kadarpH > 14:
16         fmt.Print("Nilai pH tidak valid, Nilai harus antara 0 dan 14 ")
17     }
18 }
19
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\alpro> go run "c:\alpro\unguided1.go"
4
Air tidak layak minum
- PS C:\alpro> go run "c:\alpro\unguided1.go"
8
Air layak minum
- PS C:\alpro> go run "c:\alpro\unguided1.go"
16
Nilai pH tidak valid, Nilai harus antara 0 dan 14
- PS C:\alpro>

*Untitled - Note...
File Edit Format View Help
Nama : Tio Armani
NIM : 103112430225
Kelas : 12-IF-07
100 Windows (CRLF) UTF-8

Deskripsi Program

Program ini bertujuan untuk mengevaluasi kadar pH air yang dimasukkan oleh pengguna, kemudian menentukan apakah air tersebut layak minum, tidak layak minum, atau nilai pH yang dimasukkan tidak valid.

Lalu cara kerja program ialah:

- Input Kadar pH:
Program meminta pengguna untuk memasukkan nilai pH air. Nilai ini diinput sebagai bilangan desimal (float64) dan disimpan di variabel kadarpH.
- Program menggunakan struktur switch tanpa ekspresi, di mana setiap *case* mengevaluasi kondisi spesifik terhadap nilai kadarpH. Berikut adalah detail masing-masing kondisi:
 - **Case 1:** Jika kadar pH berada di antara 6.5 dan 8.6 (termasuk kedua nilai), maka program mencetak "**Air layak minum.**" Ini menunjukkan bahwa kadar pH dalam rentang normal untuk air yang aman diminum.
 - **Case 2:** Jika kadar pH kurang dari 6.5, atau lebih besar dari 8.6 namun masih di bawah 14, maka program mencetak "**Air tidak layak minum.**" Artinya, kadar pH tersebut terlalu asam atau terlalu basa untuk dikonsumsi manusia.

- **Case 3:** Jika kadar pH lebih besar dari 14, maka program mencetak "**Nilai pH tidak valid, nilai harus antara 0 dan 14.**" Dalam hal ini, input dianggap salah karena nilai pH seharusnya selalu berada dalam rentang 0–14.

Kemudian outputnya program akan memberikan keluaran berupa deskripsi yang sesuai dengan kondisi kadar pH yang dimasukkan pengguna.

2. Tugas 2

Pseudo code

Program Parkir

Kamus

var kendaraan: **string**

var durasi: **int**

output ("Jenis Kendaraan & Durasi Parkir")

input (&kendaraan, &durasi)

motor := 2000

mobil := 5000

truk := 8000

depend on:

case kendaraan == "motor" && durasi <= 1:

output (motor)

case kendaraan == "motor" && durasi >= 2:

output (motor * durasi)

case kendaraan == "mobil" && durasi <= 1:

output (mobil)

case kendaraan == "mobil" && durasi >= 2:

output (mobil * durasi)

case kendaraan == "truk" && durasi <= 1:

output (truk)

case kendaraan == "truk" && durasi >= 1:

output (truk * durasi)

end

endprogram

Source code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi int

    fmt.Println("Jenis Kendaraan & Durasi Parkir")
    fmt.Scan(&kendaraan, &durasi)

    motor := 2000
    mobil := 5000
    truk := 8000

    switch {
    case kendaraan == "motor" && durasi <= 1:
        fmt.Println(motor)
    case kendaraan == "motor" && durasi >= 2:
        fmt.Println(motor * durasi)
    case kendaraan == "mobil" && durasi <= 1:
        fmt.Println(mobil)
    case kendaraan == "mobil" && durasi >= 2:
        fmt.Println(mobil * durasi)
    case kendaraan == "truk" && durasi <= 1:
        fmt.Println(truk)
    case kendaraan == "truk" && durasi >= 1:
        fmt.Println(truk * durasi)
    }
}
```

Screenshoot program

The screenshot shows a Go IDE with a code editor and a terminal. The code editor displays the source code for `unguided2.go`, which defines a `main` function that takes vehicle type and duration as input and calculates parking fees using a `switch` statement. The terminal shows three test runs of the program, each with different inputs and corresponding outputs. An overlaid window titled `*Untitled - Note...` displays the user's personal information.

```
guided1.go 4 | guided2.go 1 | guided3.go 1 | unguided1.go 1 | unguided2.go 1 X
-Go unguided2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var durasi int
8     fmt.Println("Jenis Kendaraan & Durasi Parkir")
9     fmt.Scan(&kendaraan, &durasi)
10
11     motor := 2000
12     mobil := 5000
13     truk := 8000
14
15     switch {
16     case kendaraan == "motor" && durasi <= 1:
17         fmt.Println(motor)
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\alpro> go run "c:\alpro\unguided2.go"
Jenis Kendaraan & Durasi Parkir
motor 1 jam
2000
- PS C:\alpro> go run "c:\alpro\unguided2.go"
Jenis Kendaraan & Durasi Parkir
mobil 6 jam
30000
- PS C:\alpro> go run "c:\alpro\unguided2.go"
Jenis Kendaraan & Durasi Parkir
truk 24 jam
192000
- PS C:\alpro>

*Untitled - Note... — □ ×

File Edit Format View Help

Nama : Tio Armani
NIM : 103112430225
Kelas : 12-IF-07

100 Windows (CRLF) UTF-8

Deskripsi program

Program ini digunakan untuk menghitung biaya parkir berdasarkan jenis kendaraan dan durasi parkir. Program ini menerima input berupa jenis kendaraan dan durasi parkir, lalu menggunakan logika percabangan `switch` untuk menentukan biaya parkir yang harus dibayar.

Cara Kerja Program

Input Data oleh Pengguna

Program meminta pengguna untuk memasukkan dua input sekaligus:

- Jenis kendaraan (*string*): Contohnya motor, mobil, atau truk.
- Durasi parkir (*int*): Durasi parkir dalam jam.
- Kedua input ini disimpan di variabel `kendaraan` dan `durasi`.

Biaya Parkir per Jam

- Biaya parkir per jam ditentukan untuk masing-masing jenis kendaraan:
 - Motor: 2000 per jam.
 - Mobil: 5000 per jam.
 - Truk: 8000 per jam.

- Penghitungan Biaya dengan Switch
 - Motor:
 - Jika durasi parkir ≤ 1 jam, biaya tetap adalah 2000.
 - Jika durasi parkir ≥ 2 jam, biaya dihitung dengan mengalikan 2000 dengan durasi.
 - Mobil:
 - Jika durasi parkir ≤ 1 jam, biaya tetap adalah 5000.
 - Jika durasi parkir ≥ 2 jam, biaya dihitung dengan mengalikan 5000 dengan durasi.
 - Truk:
 - Jika durasi parkir ≤ 1 jam, biaya tetap adalah 8000.
 - Jika durasi parkir ≥ 2 jam, biaya dihitung dengan mengalikan 8000 dengan durasi.
- Output Hasil
 - Setelah menemukan kondisi yang sesuai, program mencetak total biaya parkir berdasarkan jenis kendaraan dan durasi parkir.

Kesimpulannya program ini dirancang untuk menghitung biaya parkir berdasarkan jenis kendaraan dan durasi dengan logika sederhana. Program ini cocok untuk digunakan dalam simulasi sistem parkir kecil atau aplikasi pembelajaran logika dasar pemrograman

3. Tugas 4

Pseudo code

Program operasi MTK

Kamus

var input: **int**

output ("Masukan Angka : ")

input (&input)

depend on:

case input%2 != 0 && input < 25:

hasil := input + (input + 1)

output ("Kategori: bilangan ganjil\nHasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", input, input+1, hasil)

case input%2 == 0 && input%5 != 0 && input%10 != 0:

hasil := input * (input + 1)

output ("Kategori: bilangan genap\nHasil perkalian dengan bilangan berikutnya %d * %d = %d\n", input, input+1, hasil)

case input%5 == 0 && input%10 != 0:

Hasil := (input * input)

fmt.Printf("Hasil kuadrat dari 5\nHasil kuadrat dari %d ^2 == %d", input, Hasil)

case input%10 == 0:

hasil := input / 10

output ("Kategori bilangan kelipatan 10 \nHasil pembagian antara %d / %d = %d", input, input/10, hasil)

end

endprogram

Source code

```
package main

import "fmt"

func main() {
    var input int

    fmt.Print("Masukan Angka : ")
    fmt.Scan(&input)

    switch {

    case input%2 != 0 && input < 25:
        hasil := input + (input + 1)

        fmt.Printf("Kategori: bilangan ganjil\nHasil  
penjumlahan dengan bilangan berikutnya %d + %d = %d\n", input,
input+1, hasil)

    case input%2 == 0 && input%5 != 0 && input%10 != 0:
        hasil := input * (input + 1)

        fmt.Printf("Kategori: bilangan genap\nHasil perkalian  
dengan bilangan berikutnya %d * %d = %d\n", input, input+1, hasil)
    case input%5 == 0 && input%10 != 0:
        Hasil := (input * input)

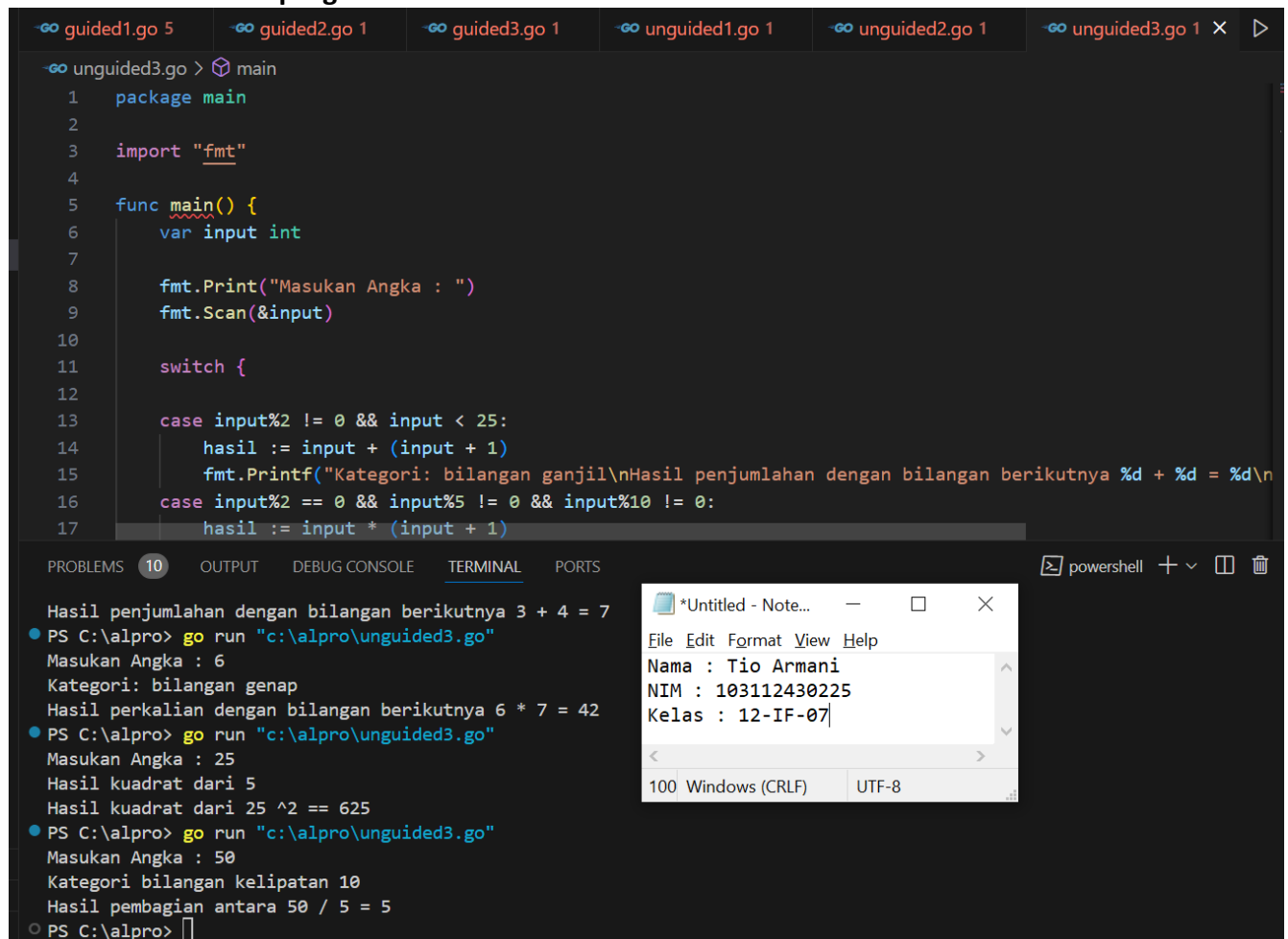
        fmt.Printf("Hasil kuadrat dari 5\nHasil kuadrat dari %d  
^2 == %d", input, Hasil)

    case input%10 == 0:
        hasil := input / 10

        fmt.Printf("Kategori bilangan kelipatan 10 \nHasil  
pembagian antara %d / %d = %d", input, input/10, hasil)

    }
}
```

Screenshoot program



```
guided1.go 5 | guided2.go 1 | guided3.go 1 | unguided1.go 1 | unguided2.go 1 | unguided3.go 1 X ▶

unguided3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var input int
7
8     fmt.Print("Masukan Angka : ")
9     fmt.Scan(&input)
10
11     switch {
12
13     case input%2 != 0 && input < 25:
14         hasil := input + (input + 1)
15         fmt.Printf("Kategori: bilangan ganjil\nHasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", input, input+1, hasil)
16     case input%2 == 0 && input%5 != 0 && input%10 != 0:
17         hasil := input * (input + 1)
18         fmt.Printf("Kategori: bilangan genap\nHasil perkalian dengan bilangan berikutnya %d * %d = %d\n", input, input+1, hasil)
19     }
20 }
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Hasil penjumlahan dengan bilangan berikutnya 3 + 4 = 7
• PS C:\alpro> go run "c:\alpro\unguided3.go"
Masukan Angka : 6
Kategori: bilangan genap
Hasil perkalian dengan bilangan berikutnya 6 * 7 = 42
• PS C:\alpro> go run "c:\alpro\unguided3.go"
Masukan Angka : 25
Hasil kuadrat dari 5
Hasil kuadrat dari 25 ^2 == 625
• PS C:\alpro> go run "c:\alpro\unguided3.go"
Masukan Angka : 50
Kategori bilangan kelipatan 10
Hasil pembagian antara 50 / 5 = 5
• PS C:\alpro>

*Untitled - Note...
File Edit Format View Help
Nama : Tio Armani
NIM : 103112430225
Kelas : 12-IF-07
100 Windows (CRLF) UTF-8

Deskripsi program

Program yang kamu buat bertujuan untuk mengkategorikan dan menghitung operasi tertentu berdasarkan input angka yang dimasukkan oleh pengguna. Program ini menggunakan struktur switch untuk menentukan jenis operasi yang akan dijalankan berdasarkan beberapa kondisi tertentu yang melibatkan pembagian dan perkalian.

Input Angka:

- Program meminta pengguna untuk memasukkan sebuah angka (input), yang kemudian disimpan dalam variabel input.
- Fungsi `fmt.Scan(&input)` digunakan untuk menerima input dari pengguna.

Struktur Switch:

- Program menggunakan struktur switch untuk mengevaluasi kondisi berdasarkan nilai yang dimasukkan oleh pengguna.
- Setiap kondisi dalam switch mengecek beberapa sifat dari angka tersebut, seperti apakah angka tersebut ganjil atau genap, kelipatan 5, kelipatan 10, dan lain sebagainya.

Kondisi dalam Switch:

- **Case 1: Angka Ganjil dan Kurang dari 25**
 - **Kondisi:** Jika angka yang dimasukkan adalah **ganjil** (`input%2 != 0`) dan **kurang dari 25** (`input < 25`).
 - **Operasi:** Program akan menjumlahkan angka dengan angka berikutnya (misalnya `input + (input + 1)`), dan menampilkan hasilnya.
 - **Output:** Program akan menampilkan bahwa angka tersebut adalah **bilangan**

- **ganjil**, dan mencetak hasil penjumlahan antara angka dan angka berikutnya.
- **Contoh:** Jika input adalah 7, hasil penjumlahannya adalah $7 + 8 = 15$.
- **Case 2: Angka Genap dan Tidak Kelipatan 5 atau 10**
 - **Kondisi:** Jika angka yang dimasukkan adalah **genap** ($\text{input} \% 2 == 0$), tetapi **bukan kelipatan 5** ($\text{input} \% 5 != 0$) dan **bukan kelipatan 10** ($\text{input} \% 10 != 0$).
 - **Operasi:** Program akan mengalikan angka dengan angka berikutnya (misalnya $\text{input} * (\text{input} + 1)$), dan menampilkan hasilnya.
 - **Output:** Program akan menampilkan bahwa angka tersebut adalah **bilangan genap**, dan mencetak hasil perkalian antara angka dan angka berikutnya.
 - **Contoh:** Jika input adalah 4, hasil kaliannya adalah $4 * 5 = 20$.
- **Case 3: Angka Kelipatan 5 dan Bukan Kelipatan 10**
 - **Kondisi:** Jika angka yang dimasukkan adalah kelipatan **5** ($\text{input} \% 5 == 0$), tetapi **bukan kelipatan 10** ($\text{input} \% 10 != 0$).
 - **Operasi:** Program akan menghitung **kuadrat** dari angka tersebut (misalnya $\text{input} * \text{input}$), dan menampilkan hasilnya.
 - **Output:** Program akan menampilkan hasil kuadrat dari angka tersebut.
 - **Contoh:** Jika input adalah 5, hasil kuadratnya adalah $5^2 = 25$.
- **Case 4: Angka Kelipatan 10**
 - **Kondisi:** Jika angka yang dimasukkan adalah kelipatan **10** ($\text{input} \% 10 == 0$).
 - **Operasi:** Program akan membagi angka dengan 10 (misalnya $\text{input} / 10$), dan menampilkan hasilnya.
 - **Output:** Program akan menampilkan bahwa angka tersebut adalah **kelipatan 10**, dan mencetak hasil pembagian antara angka dan 10.
 - **Contoh:** Jika input adalah 20, hasil bagiannya adalah $20 / 10 = 2$.

Jadi kesimpulannya adalah

- Program ini mengkategorikan angka berdasarkan sifat-sifat matematika tertentu (ganjil/genap, kelipatan 5, kelipatan 10) dan kemudian melakukan operasi yang berbeda sesuai kategori tersebut.
- Program ini sangat berguna untuk latihan pengkondisian dan struktur kontrol dalam pemrograman, dan memberikan contoh bagaimana switch dapat digunakan untuk menangani banyak kondisi dengan cara yang terorganisir.
- Program ini akan memberikan hasil yang sesuai dengan kategori angka yang dimasukkan dan menghitung hasil operasi matematika yang relevan.