

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 11
“SWITCH CASE”



DISUSUN OLEH:
RYAN AKEYLA NOVIANTO WIDODO
103112400081
S1 IF-12-01

DOSEN:
Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

DASAR TEORI(DEFAULT)

1.Data

Pengertian data adalah Data: Hasil pencatatan penelitian berupa fakta atau angka.

2.Tipe Data

Dalam pemrograman, tipe data adalah jenis nilai yang akan disimpan dalam variabel. Tipe data berfungsi untuk memberi tahu sistem komputer bagaimana menafsirkan nilai data. Ada 5 tipe Data.

Tipe-Tipe Data:

- 1.Integer, seperti: int, biasanya digunakan pada bilangan bulat
- 2.real, seperti: float64, biasanya digunakan pada bilangan decimal
3. boolean (atau logikal), seperti: bool, biasanya digunakan pada true false
- 4.karakter, seperti: byte, biasanya digunakan pada nama suatu benda atau apapun
- 5.string, seperti: string, biasanya digunakan pada suatu nama benda dan bisa menggunakan angka, hampir sama seperti karakter

3.Variabel

Variabel: Simbol atau nama yang digunakan untuk menyimpan nilai data dalam memori komputer. Variabel dapat berubah nilainya selama program komputer dijalankan.

Variabel adalah nama yang mewakili suatu elemen data, seperti tanggal lahir, untuk tempat lahir, alamat untuk alamat, tangla untuk tanggal lahir, dsb. Ada aturan tertentu yang wajib diikuti dalam pemberian nama variabel, antara lain:

1. Harus dimulai dengan abjad, tidak boleh dimulai dengan angka atau symbol. Khusus untuk PHP, variabel selalu ditulis dengan awalan berupa &.
2. Tidak boleh ada spasi.
3. Jangan menggunakan symbol-simbol yang bisa membingungkan, seperti titik dua, titik koma, koma, dsb.
4. Sebaiknya memiliki arti yang sesuai dengan elemen data.
5. Sebaiknya tidak terlalu panjang. Contoh nama variabel yang benar &nama, &nilai_ujian
Contoh nama variabel yang salah 4xyz, &ip rata.

Variabel menurut saya adalah suatu simbol yang bisa diibaratkan seperti sebuah wadah. Apa maksud wadah itu? Maksudnya adalah wadah itu biasa diisi suatu benda. Jadi variabel itu bisa diibaratkan seperti: mobil, computer, motor, handphone, pohon, ataupun benda yang lainnya. Contohnya: suatu x merupakan benda yang bisa dimakan dan y adalah minuman. Nah di sinilah variabel diibaratkan seperti itu. Jadi x bisa menjadi nasi goreng, atau apapun makanan, dan y bisa menjadi minuman apapun seperti: Es teh, Es jeruk, dll.

4. Deklarasi Variabel

Deklarasi variabel memastikan program memiliki informasi yang cukup tentang variabel sebelum digunakan, yang mana akan membantu dalam menghindari kesalahan seperti halnya penggunaan variabel yang tidak terdefinisi

5. Konstanta

Konstanta itu, seperti variabel biasa, namun... Dia mempunyai nilai tetap dan tidak dapat diubah nilainya setelah dideklarasikan. Seperti halnya dengan nilai phi, yang mana akan selalu tetap 22/7 atau 3,14

6. Input

Input adalah kegiatan memasukkan data yang mana akan kita cari hasilnya nanti

7. Output

Output adalah kegiatan yang mana setelah kita memasukkan nilai input beserta programnya seperti misalnya rumus kubus mungkin dengan rumus $p.l.t$ dan setelah diinput variabelnya maka nantinya akan menghasilkan yang namanya Output

8. Paradigma Perulangan

Perulangan merupakan salah satu struktur kontrol yang memungkinkan suatu instruksi yang sama dilakukan berulang kali dalam waktu atau jumlah yang lama. Tanpa instruksi perulangan, maka suatu instruksi akan ditulis dalam jumlah yang sangat banyak.

Sebagai contoh adalah menuliskan suatu teks "CAK1BAB3 Algoritma Pemrograman 1" sebanyak 1000 baris.

9. Karakteristik For-Loop (Perulangan berdasarkan iterasi)

Salah satu instruksi perulangan yang paling mudah adalah **for-loop**, yang mana dengan instruksi ini dapat digunakan untuk mengulangi instruksi sebanyak ***n*** kali (iterasi). Batasan besar nilai dari ***n*** menyesuaikan dengan batasan dari tipe data integer yang digunakan.

Instruksi for-loop memiliki beberapa komponen, yaitu:

1.inisialisasi merupakan assignment **variabel iterasi** yang bertipe integer. Pada contoh di atas biasanya **variabel iterasi = 0** atau **1**, artinya iterasi dimulai dari 0 atau 1.

2.kondisi merupakan suatu operasi bernilai boolean yang menyatakan kapan perulangan harus dilakukan. Pada contoh di atas **kondisi** adalah **variabel iterasi <= n** (kurang dari atau sama dengan)

3.update merupakan ekspresi yang menyatakan perubahan nilai dari **variabel iterasi**.

Pada contoh di atas update adalah **variabel iterasi = variabel iterasi + 1**.

SWITCH CASE

Dasar Teori Switch Case dalam Pemrograman Golang

Switch Case adalah struktur kontrol dalam pemrograman Golang yang memungkinkan program untuk mengeksekusi blok kode tertentu berdasarkan nilai dari sebuah ekspresi. Sederhananya, switch case digunakan untuk membandingkan nilai ekspresi dengan beberapa kemungkinan nilai yang telah ditentukan.

Cara Kerja Switch Case dalam Golang:

1. **Ekspresi Evaluasi:** Pertama, ekspresi dalam pernyataan switch dievaluasi.
2. **Pencocokan Case:** Nilai ekspresi kemudian dibandingkan dengan nilai yang ditentukan dalam setiap case.
3. **Eksekusi Blok Kode:** Jika nilai ekspresi cocok dengan nilai dalam salah satu case, blok kode yang terkait dengan case tersebut akan dieksekusi.
4. **Default Case:** Jika tidak ada case yang cocok, blok kode dalam default case (jika ada) akan dieksekusi.

Perbedaan Switch Case di Golang dengan Bahasa Lainnya:

- **Tidak Perlu Perbandingan Eksplisit:** Dalam Golang, Anda tidak perlu menuliskan perbandingan eksplisit (seperti `==`) dalam case. Perbandingan dilakukan secara implisit.
- **Gaya If-Else:** Switch di Golang dapat digunakan dengan gaya if-else. Anda dapat menuliskan kondisi perbandingan langsung dalam case.

Contoh Penggunaan Switch Case dalam Golang: package main

```
import "fmt"

func main() {
    var nilai int = 85

    switch nilai {
    case 90:
        fmt.Println("Nilai A")
    case 80:
        fmt.Println("Nilai B")
    case 70:
        fmt.Println("Nilai C")
    default:
        fmt.Println("Nilai D")
    }
}
```

Dalam contoh ini, nilai variabel `nilai` (85) dibandingkan dengan nilai dalam setiap case. Karena tidak ada case yang cocok, blok kode dalam default case akan dieksekusi, sehingga program akan mencetak "Nilai D".

Keuntungan Menggunakan Switch Case:

- **Kode Lebih Terstruktur:** Switch case membuat kode lebih terstruktur dan mudah dibaca dibandingkan dengan menggunakan serangkaian if-else.
- **Efisiensi:** Dalam beberapa kasus, switch case dapat lebih efisien daripada if-else, terutama jika terdapat banyak kondisi yang perlu diperiksa.
- **Kemudahan Pemeliharaan:** Switch case memudahkan pemeliharaan kode karena perubahan pada kondisi hanya perlu dilakukan di satu tempat.

Kesimpulan:

Switch case adalah struktur kontrol yang powerful dalam pemrograman Golang. Ia menyediakan cara yang efisien dan terstruktur untuk mengeksekusi blok kode yang berbeda berdasarkan nilai dari sebuah ekspresi. Dengan memahami dasar-dasar switch case, Anda dapat menulis kode yang lebih terstruktur, mudah dibaca, dan efisien.

CONTOH SOAL

Contoh 1:

Coding Program Contoh Soal 1:

```
contohsoal1 > go contohsoal1.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var jam12, jam24 int
7      var label string
8      fmt.Scan(&jam24)
9      switch {
10     case jam24 == 0:
11         jam12 = 12
12         label = "AM"
13     case jam24 < 12:
14         jam12 = jam24
15         label = "AM"
16     case jam24 == 12:
17         jam12 = 12
18         label = "PM"
19     case jam24 > 12:
20         jam12 = jam24 - 12
21         label = "PM"
22     }
23     fmt.Println(jam12, label)
24 }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal1\contohsoal1.go"
13
1 PM
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal1\contohsoal1.go"
0
12 AM
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal1\contohsoal1.go"
12
12 PM
PS C:\Ryan\Tugas Alprog\praltikum7> 
```

Deskripsi program dari contoh soal 1:

Tujuan program contoh soal 1 adalah membuat program yang mana kita akan menggunakan untuk melakukan konversi waktu dari bentuk 24 jam ke dalam bentuk 12 jam.

Contoh:

- 1.Input: 13, Output: 1 PM
- 2.Input: 0, Output: 12 AM
- 3.Input: 12, Output: 12 PM

Contoh 2:

Coding Program Contoh Soal 2:

```
contohsoal2 > -go contohsoal2.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var nama_tanaman string
7      fmt.Scan(&nama_tanaman)
8
9      switch nama_tanaman {
10         case "nepenthes", "drosera":
11             fmt.Println("Termasuk Tanaman Karnivora")
12             fmt.Println("Asli Indonesia")
13         case "venus", "sarracenia":
14             fmt.Println("Termasuk Tanaman Karnivora")
15             fmt.Println("Tidak Asli Indonesia")
16         default:
17             fmt.Println("Tidak Termasuk Tanaman Karnivora")
18         }
19     }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal2\contohsoal2.go"
nepenthes
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal2\contohsoal2.go"
nepenthes
nepenthes
Termasuk Tanaman Karnivora
Asli Indonesia
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal2\contohsoal2.go"
venus
venus
Termasuk Tanaman Karnivora
Tidak Asli Indonesia
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal2\contohsoal2.go"
karedok
Tidak Termasuk Tanaman Karnivora
PS C:\Ryan\Tugas Alprog\praltikum7> 
```

Deskripsi program dari contoh soal 2:

Tujuan program contoh soal 2 adalah membuat program yang mana akan digunakan untuk menentukan apakah tanaman tersebut termasuk tanaman karnivora atau tidak. Jika ya, apakah tanaman tersebut asli indonesia atau tidak.

Contoh:

1.Input: Nepenthas

Output: Termasuk Tanaman Karnivora, Asli Indonesia

2.Input: Venus

Output: Termasuk Tanaman Karnivora, Tidak Asli Indonesia

2.Input: Karedok

Output: Tidak Termasuk Tanaman Karnivora

Contoh 3:

Coding Program Contoh Soal :

```
contohsoal3 > go contohsoal3.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var tipe_kendaraan string
7      var durasi, tarif int
8
9      fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
10     fmt.Scan(&tipe_kendaraan)
11     fmt.Print("Masukkan durasi parkir (dalam jam): ")
12     fmt.Scan(&durasi)
13
14     switch {
15     case tipe_kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
16         tarif = 7000
17     case tipe_kendaraan == "Motor" && durasi > 2:
18         tarif = 9000
19     case tipe_kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
20         tarif = 15000
21     case tipe_kendaraan == "Mobil" && durasi > 2:
22         tarif = 20000
23     case tipe_kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
24         tarif = 25000
25     case tipe_kendaraan == "Truk" && durasi > 2:
26         tarif = 35000
27     default:
28         fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
29         return
30     }
31
32     fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
33 }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal3\contohsoal3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal3\contohsoal3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal3\contohsoal3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir: Rp 9000
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal3\contohsoal3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\contohsoal3\contohsoal3.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Sepeda
Masukkan durasi parkir (dalam jam): 2
Jenis kendaraan atau durasi parkir tidak valid
PS C:\Ryan\Tugas Alprog\praltikum7> █
```

Deskripsi program dari contoh soal 3:

Tujuan program contoh soal 3 adalah membuat program untuk menentukan tarif parkir berdasarkan jenis kendaraan dan durasi parkir. Terdapat tiga jenis kendaraan yang dapat diparkir dengan tarif berbeda berdasarkan durasi parkir. Untuk motor, tarifnya adalah Rp 7.000 untuk durasi 1-2 jam, dan Rp 9.000 jika durasi lebih dari 2 jam. Mobil dikenakan tarif Rp 15.000 untuk 1-2 jam, dan Rp 20.000 untuk durasi lebih dari 2 jam. Sementara itu, truk dikenakan tarif Rp 25.000 untuk 1-2 jam, dan Rp 35.000 jika lebih dari 2 jam.

Contoh:

1.Input: Motor 2 jam

Output: 7000

2.Input: Mobil 4 jam

Output: 20000

3.Input: Motor 3 jam

Output: 9000

4.Input: Truk 1 jam

Output: 25000

5.Input: Sepeda 2 jam

Output: Jenis kendaraan atau durasi tidak valid

LATIHAN SOAL

Latihan Soal 1:

Coding Program Latihan Soal 1:

```
latsol 1 > go latsol1.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var ph float64
7      fmt.Scan(&ph)
8
9      switch {
10     case ph >= 6.5 && ph <= 8.6:
11         fmt.Println("Air Layak Minum")
12     case (ph > 0 && ph < 6.5) || (ph <= 14 && ph > 8.6):
13         fmt.Println("Air Tidak Layak Minum")
14     default:
15         fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14")
16     }
17 }
```

Hasil Coding:

```
Jenis kendaraan atau durasi parkir tidak valid
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 1\latsol1.go"
8.6
Air Layak Minum
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 1\latsol1.go"
9
Air Tidak Layak Minum
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 1\latsol1.go"
16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14
PS C:\Ryan\Tugas Alprog\praltikum7> █
```

Deskripsi program dari Latihan soal 1:

Tujuan program Latihan soal 1 adalah membuat program yang mana kita akan menentukan apakah kadar pH pada air yang diinput termasuk Air yang layak untuk diminum atau tidak.

Contoh:

1.Input: 8,6

Ouput: Air Layak Minum

2.Input: 9

Ouput: Air Tidak Layak Minum

3.Input 16

Ouput: Nilai pH tidak valid

Latihan Soal 2:

Coding Program Latihan Soal 2:

```
latsol 2 > go latsol2.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var kendaraan string
7      var durasi int
8      var tarif int
9      fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
10     fmt.Scan(&kendaraan)
11     fmt.Print("Masukkan durasi parkir (dalam jam): ")
12     fmt.Scan(&durasi)
13
14     if durasi < 1 {
15         durasi = 1
16     }
17
18     switch kendaraan {
19     case "motor":
20         tarif = durasi * 2000
21
22     case "mobil":
23         tarif = durasi * 5000
24
25     case "truk":
26         tarif = durasi * 8000
27
28     default:
29         fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
30     }
31     fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
32 }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 2\latsol2.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): motor
Masukkan durasi parkir (dalam jam): 3
Tarif Parkir: Rp 6000
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 2\latsol2.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): mobil
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 5000
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 2\latsol2.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): truk
Masukkan durasi parkir (dalam jam): 5
Tarif Parkir: Rp 40000
PS C:\Ryan\Tugas Alprog\praltikum7> █
```

Deskripsi program dari Latihan soal 2:

Tujuan program Latihan soal 2 adalah membuat suatu program untuk menghitung tarif parkir berdasarkan jenis kendaraan dan durasi parkir yang dimasukkan oleh pengguna. Ada tiga jenis kendaraan: motor, mobil, dan truk, dengan masing-masing memiliki tarif parkir yang berbeda.

Contoh:

1.Input: motor 3 jam

Output: 6000

2.Input: mobil 1 jam

Output: 5000

3.Input: truk 5 jam

Output: 40000

Latihan Soal 3:

Coding Program Latihan Soal 3:

```
latsol 3 > latsol3.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n, hasil int
7      fmt.Scan(&n)
8      switch {
9      case n%10 == 0:
10         hasil = n / 10
11         fmt.Printf("Kategori: Bilangan Kelipatan 10\nHasil pembagian antara %d / 10 = %d\n", n, hasil)
12      case n%5 == 0 && n != 5:
13         hasil = n * n
14         fmt.Printf("Kategori: Bilangan Kelipatan 5\nHasil kuadrat dari %d^2 = %d\n", n, hasil)
15      case n%2 == 0:
16         hasil = n * (n + 1)
17         fmt.Printf("Kategori: Bilangan Ganjil\nHasil penjumlahan dengan bilangan %d * %d = %d\n", n, n+1, hasil)
18      case n%2 != 0:
19         hasil = n + (n + 1)
20         fmt.Printf("Kategori: Bilangan Genap\nHasil perkalian dengan bilangan %d + %d = %d\n", n, n+1, hasil)
21      default:
22         fmt.Println("Tidak termasuk kategori apapun.")
23      }
24 }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 3\latsol3.go"
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 3\latsol3.go"
5
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan 5 + 6 = 11
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 3\latsol3.go"
8
Kategori: Bilangan Ganjil
Hasil penjumlahan dengan bilangan 8 * 9 = 72
Hasil penjumlahan dengan bilangan 8 * 9 = 72
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 3\latsol3.go"
25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25^2 = 625
PS C:\Ryan\Tugas Alprog\praltikum7> go run "c:\Ryan\Tugas Alprog\praltikum7\latsol 3\latsol3.go"
20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS C:\Ryan\Tugas Alprog\praltikum7> █
```

Deskripsi program dari Latihan soal 3:

Tujuan program Latihan soal 3 adalah membuat suatu program yang digunakan untuk mengidentifikasi pola aritmatika berdasarkan bilangan yang diinputkan dan melakukan operasi matematika yang sesuai. Di antaranya: Ganjil, Genap, Kelipatan 5, dan Kelipatan 10

Ganjil dan Genap dideteksi dari angka awal seperti: 8 menjadi ganjil yaitu 9

Contoh:

1.Input: 5

Kategori: Bilangan Genap

Hasil perkalian dengan bilangan 5 + 6 = 11

2.Input: 8

Kategori: Bilangan Ganjil

Hasil penjumlahan dengan bilangan 8 * 9 = 72

Hasil penjumlahan dengan bilangan 8 * 9 = 72

3.Input: 25

Kategori: Bilangan Kelipatan 5

Hasil kuadrat dari 25^2 = 625

4.Input: 20

Kategori: Bilangan Kelipatan 10

Hasil pembagian antara 20 / 10 = 2