

MODUL PRAKTIKUM 10 - ELSE-IF

# ALGORITMA DAN PEMROGRAMAN 1

S1 INFORMATIKA

**GO**

*Published by school of computing*

Our official instagram



@informaticslab\_telu

## LEMBAR PENGESAHAN

Saya yang bertanda tangan di bawah ini:

Nama : Prasti Eko Yunanto, S.T., M.Kom.  
NIP : 19890017  
Koordinator Mata Kuliah : Algoritma dan Pemrograman 1  
Prodi : S1 Informatika

Menerangkan dengan sesungguhnya bahwa modul ini digunakan untuk pelaksanaan praktikum di Semester Ganjil Tahun Ajaran 2024/2025 di Laboratorium Informatika, Fakultas Informatika, Universitas Telkom.

Bandung, 17 Agustus 2024



Mengesahkan,

Koordinator Mata Kuliah  
Algoritma Pemrograman 1

A blue ink signature of Prasti Eko Yunanto, S.T., M.Kom.

Prasti Eko Yunanto, S.T., M.Kom.  
NIP. 19890017



Mengetahui,

Kaprodi S1 Informatika

A blue ink signature of Dr. Erwin Budi Setiawan, S.Si., M.T.

Dr. Erwin Budi Setiawan, S.Si., M.T.  
NIP. 00760045

## MODUL 10. ELSE-IF

### 10.1 Paradigma Percabangan

Sebelumnya telah dipelajari bahwa setiap baris kode program akan dieksekusi satu persatu secara sekuensial. Artinya kode program dari baris ke-1 hingga baris terakhir akan dieksekusi satu persatu. Bagaimana jika kita ingin **baris kode program yang dieksekusi itu berdasarkan syarat atau suatu ketentuan** tertentu? Sebagai analogi misalnya ketika kita berada dipersimpangan jalan ke kiri atau ke kanan. Maka di dalam pemrograman hal tersebut mungkin untuk dilakukan, struktur kontrol yang digunakan adalah **else-if**.

### 10.2 Karakteristik Else-If

Pada dasarnya else-if tidak jauh berbeda dengan struktur kontrol percabangan menggunakan if-then. Perbedaannya terletak pada adanya **aksi lain** yang akan dieksekusi apabila kondisi tidak terjadi atau bernilai false.

Penulisan struktur kontrol percabangan dengan menggunakan else-if terdiri dari dua bagian, yaitu:

- 1) **Kondisi**, yaitu sesuatu syarat atau ketentuan dari suatu percabangan. **Kondisi** ini harus bernilai **boolean**, baik itu variabel ataupun operasi tipe data.
- 2) **Aksi**, yaitu kumpulan instruksi yang akan dilakukan apabila **kondisi terpenuhi atau bernilai true**.
- 3) **Aksi lain**, yaitu kumpulan instruksi yang akan dilakukan apabila **kondisi terpenuhi atau bernilai false**. Artinya aksi dan aksi lain ini merupakan pilihan, yang mana hanya salah satu aksi saja yang akan dieksekusi sesuai dengan nilai dari kondisi.

Sebagai contoh ilustrasi penulisannya adalah sebagai berikut:

Notasi dalam pseudocode	Notasi dalam bahasa Go
<pre>if kondisi then     // aksi else     // aksi lain endfor</pre>	<pre>if kondisi {     // aksi }else{     // aksi lain }</pre>

Penulisan aksi diberi tab atau 4 spasi (sama seperti ), untuk mempermudah pembacaan bagian baris kode mana yang merupakan aksi dan mana yang tidak.

Untuk penulisan dengan kondisi yang lebih banyak maka, notasi dapat ditulis ke dalam bentuk seperti berikut ini:

Notasi dalam pseudocode	Notasi dalam bahasa Go
<pre> <b>if</b> kondisi_1 <b>then</b>     // aksi 1 <b>else</b>     <b>if</b> kondisi_2 <b>then</b>         // aksi 2     <b>else</b>         <b>if</b> kondisi_3 <b>then</b>             // aksi 3         <b>else</b>             // aksi lain         <b>endfor</b>     <b>endfor</b> <b>endfor</b> </pre>	<pre> <b>if</b> kondisi_1 {     // aksi 1 <b>}else{</b>     <b>if</b> kondisi_2 {         // aksi 2     <b>}else{</b>         <b>if</b> kondisi_3 {             // aksi 3         <b>}else{</b>             // aksi lain         <b>}</b>     <b>}</b> <b>}</b> </pre>

Setiap aksi ke-i hanya akan dieksekusi apabila kondisi ke-i bernilai true atau benar, sedangkan aksi lain hanya akan dieksekusi apabila semua kondisi dari  $i = 1$  s.d  $n$  bernilai false. Bentuk penulisan lain yang lebih sederhana adalah sebagai berikut ini:

Notasi dalam pseudocode	Notasi dalam bahasa Go
<pre> <b>if</b> kondisi_1 <b>then</b>     // aksi 1 <b>else if</b> kondisi_2 <b>then</b>     // aksi 2 <b>else if</b> kondisi_3 <b>then</b>     // aksi 3 <b>else</b>     // aksi lain <b>endfor</b> </pre>	<pre> <b>if</b> kondisi_1 {     // aksi 1 <b>}else if</b> kondisi_2 {     // aksi 2 <b>}else if</b> kondisi_3 {     // aksi 3 <b>}else{</b>     // aksi lain <b>}</b> </pre>

### 10.3 Implementasi menggunakan Go

Sebagai contoh, misalnya terdapat suatu program yang akan menampilkan hasil operasi pembagian ( $a/b$ ). Hasil pembagian hanya akan ditampilkan apabila pembagi **tidaklah bernilai 0**. Apabila pembagi bernilai 0 maka akan ditampilkan suatu pesan "Variabel b bernilai nol".

```

1 // filename: elseif1.go
2 package main
3 import "fmt"
4
5 func main() {
6     var a, b, hasil float64
7
8     fmt.Scan(&a, &b)
9     if b != 0 {
10         hasil = a / b
11         fmt.Println("Hasil pembagian adalah", hasil)
12     }else{
13         fmt.Println("Variabel b bernilai nol")
14     }
15     fmt.Println("Program selesai")
16 }

```

```

C:\users\go\src\hello>go build elseif1.go
C:\users\go\src\hello>.\elseif1
5 2
Hasil pembagian adalah 2.5
Program selesai
C:\users\go\src\hello>.\elseif1
5 0
Variabel b bernilai nol
Program selesai

```

Pada contoh di atas terlihat bahwa, baris ke-13 hanya akan dieksekusi apabila kondisi di baris ke-9 bernilai false, artinya b bernilai 0.

Contoh konversi (nilai, tubes, kehadiran) menjadi indeks nilai.

```

1 if nilai > 75 && adaTubes {
2     indeks = 'A'
3 } else if nilai > 65 {
4     indeks = 'B'
5 } else if nilai > 50 && pctHadir > 0.7 {
6     indeks = 'C'
7 } else {
8     indeks = 'F'
9 }
10 fmt.Printf( "Nilai %v dengan kehadiran %v%% dan buat tubes=%v, mendapat indeks
11 %c\n", nilai, pctHadir, adaTubes, indeks )

```

#### 10.4 Contoh Soal Modul 10

- 1) Buatlah program dengan bahasa Go yang digunakan untuk menentukan apakah seorang penduduk bisa membuat KTP atau tidak. Syarat utama membuat KTP adalah berusia minimal 17 tahun dan memiliki kartu keluarga.

**Masukan** terdiri dari dua baris. Baris pertama adalah bilangan bulat yang menyatakan usia, sedangkan baris kedua adalah boolean yang menyatakan memiliki KTP atau tidak.

**Keluaran** berupa teks "bisa membuat KTP" atau "belum bisa membuat KTP" sesuai dengan masukan yang diberikan.

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	17 true	bisa membuat KTP
2	20 false	belum bisa membuat KTP
3	15 true	belum bisa membuat KTP

**Jawaban:**

```
1 package main
2 import "fmt"
3 func main() {
4     var usia int
5     var kk bool
6     fmt.Scan(&usia, &kk)
7     if usia >= 17 && kk {
8         fmt.Println("bisa membuat KTP")
9     }else{
10        fmt.Println("belum bisa membuat KTP")
11    }
12 }
```

```
gppras@SR8 GO % go build Demo_Soal.go
```

```
gppras@SR8 GO % ./Demo_Soal
```

```
17
```

```
true
```

```
bisa membuat KTP
```

```
gppras@SR8 GO % ./Demo_Soal
```

```
20
```

```
false
```

```
belum bisa membuat KTP
```

```
gppras@SR8 GO % ./Demo_Soal
```

```
15
```

```
true
```

```
belum bisa membuat KTP
```

- 2) Buatlah suatu program Go yang digunakan untuk menentukan suatu alfabet yang diberikan adalah vokal atau konsonan.

**Masukan** terdiri dari suatu huruf.

**Keluaran** berupa teks yang menyatakan huruf adalah "vokal" atau "konsonan" atau "bukan huruf".

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	A	vokal
2	f	konsonan
3	1	bukan huruf
4	\$	bukan huruf

**Jawaban:**

```
1 package main
2 import "fmt"
3 func main() {
4     var x rune
5     var huruf, vKecil, vBesar bool
6     fmt.Scanf("%c", &x)
7     huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
8     vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
9     vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
10    if huruf && (vKecil || vBesar) {
11        fmt.Println("vokal")
12    } else if huruf && !(vKecil || vBesar) {
13        fmt.Println("konsonan")
14    } else {
15        fmt.Println("bukan huruf")
16    }
17 }
```

```
gppras@SR8 GO % go build Demo_Soal.go
```

```
gppras@SR8 GO % ./Demo_Soal
```

```
A
```

```
vokal
```

```
gppras@SR8 GO % ./Demo_Soal
```

```
f
```

```
konsonan
```

```
gppras@SR8 GO % ./Demo_Soal
```

```
1
```

```
bukan huruf
```

```
gppras@SR8 GO % ./Demo_Soal
```

```
$
```

```
bukan huruf
```

- 3) Buatlah program untuk menentukan apakah digit dalam suatu bilangan terurut membesar, mengecil, atau tidak terurut. Bilangan hanya terdiri dari empat digit saja, atau lebih besar atau sama dengan 1000 dan kecil sama atau sama dengan 9999.

Perhatikan contoh sesi interaksi program berikut (teks bergaris bawah adalah input/read):

No.	Contoh masukan dan keluaran
1	Bilangan: <u>2489</u> Digit pada bilangan 2489 terurut membesar
2	Bilangan: <u>3861</u> Digit pada bilangan 3861 tidak terurut
3	Bilangan: <u>9651</u> Digit pada bilangan 9651 terurut mengecil

**Catatan:** Asumsi digit pada bilangan tidak ada yang sama.

#### Jawaban

```
1 package main
2 import "fmt"
3 func main() {
4     var bilangan, d1, d2, d3, d4 int
5     var teks string
6     fmt.Print("Bilangan: ")
7     fmt.Scan(&bilangan)
8     d4 = bilangan % 10
9     d3 = (bilangan % 100) / 10
10    d2 = (bilangan % 1000) / 10
11    d1 = bilangan / 1000
12    if d1 < d2 && d2 < d3 && d3 < d4 {
13        teks = "terurut membesar"
14    } else if d1 > d2 && d2 > d3 && d3 > d4 {
15        teks = "terurut mengecil"
16    } else {
17        teks = "tidak terurut"
18    }
19    fmt.Println("Digit pada bilangan", bilangan, teks)
20 }
```



```
gppras@SR8 G0 % go build Demo_Soal.go
gppras@SR8 G0 % ./Demo_Soal
Bilangan: 2489
Digit pada bilangan 2489 tidak terurut
gppras@SR8 G0 % ./Demo_Soal
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
gppras@SR8 G0 % ./Demo_Soal
Bilangan: 9651
Digit pada bilangan 9651 tidak terurut
gppras@SR8 G0 % ./Demo_Soal
Bilangan: 5555
Digit pada bilangan 5555 tidak terurut
```

