

MODUL PRAKTIKUM 9 - IF-THEN

# 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 9. IF-THEN

### 9.1 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? Misalnya baris pertama dan terakhir saja yang dijalankan. Maka di dalam pemrograman hal tersebut mungkin untuk dilakukan, struktur kontrol yang digunakan adalah **if-then**.

### 9.2 Karakteristik If-Then

Penulisan struktur kontrol percabangan dengan menggunakan if-then pada dasarnya 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**. Artinya baris aksi tidak secara default dieksekusi, tetapi ada syarat yang harus terpenuhi terlebih dahulu sehingga aksi dapat dieksekusi.

Sebagai contoh ilustrasi penulisannya adalah sebagai berikut:

Notasi dalam pseudocode	Notasi dalam bahasa Go
<b>if</b> kondisi <b>then</b> // aksi <b>endfor</b>	<b>if</b> kondisi { // aksi }

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

### 9.3 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**.

```

1 // filename: ifthen1.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     }
13     fmt.Println("Program selesai")
14 }

```

```

C:\users\go\src\hello>go build ifthen1.go
C:\users\go\src\hello>.\ifthen1
5 2
Hasil pembagian adalah 2.5
Program selesai
C:\users\go\src\hello>.\ifthen1
5 0
Program selesai

```

Baris ke-10 dan 11 hanya akan dieksekusi apabila kondisi pada baris ke-9 bernilai true. Pada contoh ketika nilai b adalah 2, maka program akan menampilkan hasil pembagian, sedangkan ketika b adalah 0, maka tidak ada hasil operasi yang ditampilkan.

#### 9.4 Contoh Soal Modul 9

- 1) Sebuah program digunakan untuk menghitung nilai absolut atau mutlak dari suatu bilangan.

**Masukan** terdiri dari suatu bilangan bulat.

**Keluaran** berupa nilai absolut atau mutlak dari bilangan yang diberikan.

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	10	10
2	-3	3
3	5	5
4	0	0
5	-9999	9999

## Jawaban

```
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     fmt.Scan(&bilangan)
6     if bilangan < 0 {
7         bilangan = -bilangan
8     }
9     fmt.Println(bilangan)
10 }
```

E:\DEV\NATIVE\GO>go build Demo\_Soal.go

E:\DEV\NATIVE\GO>.\Demo\_Soal

10

10

E:\DEV\NATIVE\GO>.\Demo\_Soal

-3

3

E:\DEV\NATIVE\GO>.\Demo\_Soal

5

5

E:\DEV\NATIVE\GO>.\Demo\_Soal

-9999

9999

- 2) Buatlah program Go yang digunakan untuk menentukan apakah suatu bilangan yang diberikan adalah bilangan positif atau bukan.

**Masukan** terdiri dari suatu bilangan bulat.

**Keluaran** terdiri dari teks yang menyatakan bilangan bulat adalah "positif" atau "bukan positif".

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	10	positif
2	-3	bukan positif
3	5	positif
4	0	bukan positif

## Jawaban

```
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     var teks string
6     fmt.Scan(&bilangan)
7     teks = "bukan positif"
8     if bilangan > 0 {
9         teks = "positif"
10    }
11    fmt.Println(teks)
12 }
```

E:\DEV\NATIVE\GO>go build Demo\_Soal.go

E:\DEV\NATIVE\GO>.\Demo\_Soal

10

positif

E:\DEV\NATIVE\GO>.\Demo\_Soal

-3

bukan positif

E:\DEV\NATIVE\GO>.\Demo\_Soal

5

positif

E:\DEV\NATIVE\GO>.\Demo\_Soal

0

bukan positif



- 3) Buatlah suatu program dengan menggunakan bahasa pemrograman Go, yang mana program digunakan untuk menentukan apakah bilangan yang diberikan adalah bilangan genap negatif atau bukan.

**Masukan** terdiri dari suatu bilangan bulat.

**Keluaran** terdiri dari suatu boolean yang menyatakan true apabila bilangan yang diberikan adalah genap negatif, atau false apabila bukan genap negatif.

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	10	false
2	-3	false
3	-4	true
4	0	false
5	-2	true

**Jawaban:**

```
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     var hasil bool
6     fmt.Scan(&bilangan)
7     hasil = bilangan % 2 == 0 && bilangan < 0
8     fmt.Println(hasil)
9 }
```

E:\DEV\NATIVE\GO>go build Demo\_Soal.go

E:\DEV\NATIVE\GO>.\Demo\_Soal

10

false

E:\DEV\NATIVE\GO>.\Demo\_Soal

-3

false

E:\DEV\NATIVE\GO>.\Demo\_Soal

-4

true

E:\DEV\NATIVE\GO>.\Demo\_Soal

0

false

E:\DEV\NATIVE\GO>.\Demo\_Soal

-2

true

**Catatan:** Tidak semua soal yang terlihat seperti menggunakan percabangan perlu diselesaikan dengan percabangan. Pada soal ini, nilai true dan false dapat langsung dihasilkan dari operasi perbandingan dan logika.