

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 9
“IF-THEN”



DISUSUN OLEH:
DAFFA TSAQIFNA FAUZTSANY
103112400032
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:

Dasar Teori: Struktur Percabangan dalam Pemrograman

Dalam dunia pemrograman, struktur percabangan digunakan untuk membuat program mengambil keputusan berdasarkan kondisi tertentu. Percabangan memungkinkan eksekusi kode yang berbeda sesuai dengan syarat yang terpenuhi.

1. Paradigma Percabangan

- ☐ If-Then dan Else-If adalah dua pendekatan utama dalam struktur kontrol.
- ☐ If-Then digunakan untuk mengeksekusi satu blok instruksi jika kondisi tertentu benar (*true*).
- ☐ Else-If adalah perluasan dari If-Then, yang memungkinkan beberapa kondisi dievaluasi secara berurutan.

2. Komponen Dasar

1. Kondisi:

- Syarat atau logika boolean yang harus terpenuhi.
- Misalnya, `a > b`, `x == y`, atau `nilai >= 75`.

2. Aksi:

- Kumpulan instruksi yang dieksekusi jika kondisi bernilai **true**.
- Contoh dalam pseudocode:

```
if kondisi then
  aksi
end if
```

3. Aksi Lain:

- Dijalankan jika kondisi bernilai false (hanya ada pada struktur Else-if)
- Contoh:

```
if kondisi_1 then
  aksi 1
else if kondisi_2 then
  aksi 2
else
  aksi lain
end
```

3. Implementasi

Contoh sederhana dengan bahasa Go untuk menentukan bilangan positif:

```
if bilangan > 0 {
    fmt.Println("Positif")
} else {
```

```
        fmt.Println("Bukan Positif")
    }
```

4. Kelebihan Struktur Percabangan

- **Flexibilitas:** Mampu menangani keputusan kompleks.
- **Efisiensi:** Meminimalkan kode redundan.
- **Keterbacaan:** Membantu programmer memahami alur logika program dengan jelas.

5. Contoh Penerapan

a. Menghitung nilai mutlak:

```
if bilangan < 0 {
    bilangan = -bilangan
}
fmt.Println(bilangan)
```

b. Menentukan bilangan genap negatif:

```
hasil = bilangan % 2 == 0 && bilangan < 0
fmt.Println(hasil) // true jika genap dan negatif
```

c. Mengevaluasi indeks nilai berdasarkan kondisi:

```
if nilai > 75 {
    indeks = 'A'
} else if nilai > 65 {
    indeks = 'B'
} else {
    indeks = 'C'
}
```

CONTOH SOAL

Conso 1:

```
package main

import "fmt"

func main() {
    var bilangan int
    fmt.Scan(&bilangan)
    if bilangan < 0 {
        bilangan = -bilangan
    }
    fmt.Println(bilangan)
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-1.go'
-3
3
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-1.go'
5
5
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-1.go'
0
0
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-1.go'
-9999
9999
```

Deskripsi Program:

Program Go di atas digunakan untuk menghitung nilai absolut (absolute value) dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Nilai absolut adalah nilai positif dari sebuah bilangan tanpa memperhatikan tanda (positif atau negatif). Berikut adalah penjelasan rinci mengenai program tersebut:

1. Deklarasi variabel:

```
var bilangan int
```

- **bilangan:** Menyimpan bilangan bulat yang akan dimasukkan oleh pengguna.

2. Membaca input dari pengguna:

```
fmt.Scan(&bilangan)
```

- Program meminta pengguna memasukkan sebuah bilangan bulat, baik positif maupun negatif.

3. Mengevaluasi nilai absolut:

```
if bilangan < 0 {  
    bilangan = -bilangan  
}
```

Kondisi if:

- Jika bilangan lebih kecil dari 0 (negatif), maka nilai **bilangan** diubah menjadi positif dengan mengalikannya dengan -1.
- Contoh:
 - Jika pengguna memasukkan **-5**, maka kondisi `bilangan < 0` terpenuhi, dan nilai **bilangan** menjadi **5**.

4. Menampilkan hasil:

```
fmt.Println(bilangan)
```

- Program mencetak nilai absolut dari bilangan yang telah dihitung.

Contoh: Jika pengguna memasukkan

```
-7(input)  
7(output)
```

2.cosol2

```
package main  
import "fmt"  
func main() {  
    var bilangan int  
    var teks string  
    fmt.Scan(&bilangan)  
    teks = "bukan positif"  
    if bilangan > 0 {  
        teks = "positif"  
    }  
    fmt.Println(teks)  
}
```

Output:

```

PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-2.go'
10
positif
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-2.go'
-3
bukan positif
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-2.go'
5
positif
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-2.go'
0
bukan positif

```

Deskripsi Program:

Program Go di atas digunakan untuk menentukan apakah sebuah bilangan bulat yang dimasukkan oleh pengguna adalah **bilangan positif** atau **bukan positif** (termasuk nol atau bilangan negatif).

Berikut penjelasannya:

1. Deklarasi variabel:

```

var bilangan int
var teks string

```

- **bilangan:** Menyimpan nilai bilangan bulat yang akan dimasukkan oleh pengguna.
- **teks:** Variabel string yang digunakan untuk menyimpan hasil evaluasi (apakah bilangan tersebut "positif" atau "bukan positif").

2. Membaca input dari pengguna:

```

fmt.Scan(&bilangan)

```

- Program meminta pengguna memasukkan sebuah bilangan bulat.

3. Logika evaluasi:

```

teks = "bukan positif"
if bilangan > 0 {
    teks = "positif"
}

```

- **Default nilai teks:** Diinisialisasi dengan "bukan positif". Ini mencakup kondisi jika bilangan nol atau negatif.
- **Kondisi if:**
 1. Jika bilangan lebih besar dari 0, maka nilai **teks** diubah menjadi "positif".
 2. Jika tidak (bilangan nol atau negatif), nilai **teks** tetap "bukan positif".

4. Menampilkan hasil:

```

fmt.Println(teks)

```

- Program mencetak nilai dari variabel **teks** yang menyatakan apakah bilangan tersebut "positif" atau "bukan positif".

Contoh: Jika pengguna memasukkan

```

7(input)

```

3.cosol3

```
package main
import "fmt"
func main() {
    var bilangan int
    var hasil bool
    fmt.Scan(&bilangan)
    hasil = bilangan%2 == 0 && bilangan < 0
    fmt.Println(hasil)
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-3.go'
10
false
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-3.go'
-3
false
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-3.go'
-4
true
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-3.go'
0
false
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\conso6\conso6-3.go'
-2
true
```

Deskripsi Program:

Program Go di atas digunakan untuk mengevaluasi apakah sebuah bilangan bulat yang dimasukkan oleh pengguna memenuhi **dua kondisi berikut secara bersamaan**:

1. Bilangan tersebut adalah **bilangan genap**.
2. Bilangan tersebut adalah **bilangan negatif**.

Jika kedua kondisi tersebut terpenuhi, program mencetak **true**; jika tidak, program mencetak **false**.

1. **Deklarasi variabel:**

```
var bilangan int
var hasil bool
```

- **bilangan**: Menyimpan bilangan bulat yang dimasukkan oleh pengguna.

- **hasil**: Menyimpan hasil evaluasi logika yang akan bernilai **true** jika kondisi terpenuhi, atau **false** jika tidak.

2. Membaca input dari pengguna:

```
fmt.Scan(&bilangan)
```

- Program meminta pengguna untuk memasukkan sebuah bilangan bulat.

3. Evaluasi logika:

```
hasil = bilangan%2 == 0 && bilangan < 0
```

- **bilangan%2 == 0**: Mengecek apakah bilangan genap (sisanya pembagian bilangan dengan 2 adalah 0).
- **bilangan < 0**: Mengecek apakah bilangan negatif.
- **&&**: Operator logika **AND** memastikan bahwa kedua kondisi harus terpenuhi agar hasilnya bernilai **true**.

4. Menampilkan hasil:

```
fmt.Println(hasil)
```

- Program mencetak nilai **hasil** (boolean), yaitu:
 - **true** jika bilangan genap dan negatif.
 - **false** jika salah satu atau kedua kondisi tidak terpenuhi.

Contoh: jika pengguna memasukkan:

```
-4(input)
```

```
true(output)
```


SOAL LATIHAN

1. Latihan1

Source Code:

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x)
    if x == 1 {
        fmt.Print(1)
    } else {
        y = x / 2
        x = x % 2
        if x == 1 {
            fmt.Print(y + 1)
        } else {
            fmt.Print(y)
        }
    }
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-1.go'
10
5
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-1.go'
1
1
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-1.go'
25
13
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-1.go'
9
5
```

Deskripsi Program:

Program Go di atas digunakan untuk menghitung **jumlah bilangan 2 (dua)** yang dapat membagi bilangan x, dengan kemungkinan menyisakan 1 jika x adalah bilangan ganjil. Program ini menghasilkan hasil pembagian tersebut, ditambah satu jika terdapat sisa 1. Berikut adalah penjelasan rinci:

1. Deklarasi variabel:

```
var x, y int
```

- **x**: Menyimpan bilangan bulat yang dimasukkan oleh pengguna.
- **y**: Menyimpan hasil pembagian awal dari x dengan 2.

2. Membaca input dari pengguna:

```
fmt.Scan(&x)
```

- Program meminta pengguna memasukkan sebuah bilangan bulat x.

3. Logika utama:

Kasus 1: Jika x adalah 1

```
if x == 1 {  
    fmt.Print(1)  
}
```

- Jika pengguna memasukkan $x = 1$, program langsung mencetak **1** tanpa melakukan perhitungan lebih lanjut.

Kasus 2: Jika $x > 1$

```
y = x / 2  
x = x % 2  
if x == 1 {  
    fmt.Print(y + 1)  
} else {  
    fmt.Print(y)  
}
```

- **Pembagian awal:**
 - $y = x / 2$: Menghitung berapa kali bilangan 2 masuk ke dalam x secara utuh.
 - $x = x \% 2$: Menghitung sisa pembagian x dengan 2.
 - Jika $x \% 2 = 0$, x adalah genap.
 - Jika $x \% 2 = 1$, x adalah ganjil.
- **Pengecekan sisa:**
 - Jika sisa pembagian $x = 1$ (ganjil), maka hasilnya adalah $y + 1$.
 - Jika tidak ada sisa (genap), hasilnya tetap y .

4. Menampilkan hasil:

```
fmt.Print(y)
```

- Hasil akhir dari perhitungan dicetak menggunakan `fmt.Print(y)`

Contoh: Jika pengguna memasukkan

```
10(input)
```

```
5(output)
```

2. Latihan2

Source Code:

```
package main

import "fmt"

func main() {
    var x int
    fmt.Scan(&x)
    if x < 0 && x%2 == 0 {
        fmt.Print("genap negatif")
    } else {
        fmt.Print("bukan")
    }
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-2.go'
10
bukan
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-2.go'
-4
genap negatif
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-2.go'
0
bukan
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-2.go'
-2
genap negatif
```

Deskripsi Program:

Program Go di atas digunakan untuk menentukan apakah bilangan yang dimasukkan oleh pengguna adalah **bilangan genap negatif**. Jika bilangan memenuhi kriteria, program mencetak "**genap negatif**", jika tidak, program mencetak "**bukan**".

1. Deklarasi variabel:

```
var x int
```

- **x**: Variabel yang menyimpan bilangan bulat yang dimasukkan oleh pengguna.

2. Membaca input dari pengguna:

```
fmt.Scan(&x)
```

- Program meminta pengguna untuk memasukkan sebuah bilangan bulat x.

3. Logika Utama:

```
if x < 0 && x%2 == 0 {
    fmt.Print("genap negatif")
} else {
```

```
        fmt.Print("bukan")
    }
```

- **Kondisi $x < 0$ & $x < 0$:**
 - ☐ Mengecek apakah bilangan xxx adalah bilangan negatif.
- **Kondisi $x \% 2 == 0$:**
 - ☐ Mengecek apakah bilangan xxx adalah bilangan genap (sisa pembagian dengan 2 adalah 0).
- **Operator Logika $\&\&$:**
 - ☐ Kedua kondisi di atas harus terpenuhi secara bersamaan agar program mencetak "genap negatif".
- **Kondisi Else:**
 - ☐ Jika salah satu atau kedua kondisi tidak terpenuhi, program mencetak "**bukan**".

Contoh: Jika pengguna memasukkan:

```
-4(input)
genap negative(output)
6(input)
bukan(output)
```

3. Latihan3

Source Code:

```
package main

import "fmt"

func main() {
    var x, y int
    var a, b bool
    fmt.Scan(&x, &y)
    if y%x == 0 {
        a = true
    }
    if x%y == 0 {
        b = true
    }
    fmt.Print(a, "\n", b)
}
```

Output:

```

PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-3.go'
10 5
false
true
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-3.go'
3 21
true
false
PS D:\test bs> go run 'd:\test bs\lab shit\6th meet\latso6\latso6-3.go'
4 4
true
true

```

Deskripsi Program:

Program di atas mengevaluasi hubungan pembagian antara dua bilangan bulat xxx dan yyy.

Program ini menentukan apakah:

1. y habis dibagi x (**variabel a**).
2. x habis dibagi y (**variabel b**).

Hasil evaluasi untuk kedua kondisi tersebut dicetak sebagai nilai boolean (true atau false).

1. Deklarasi variabel:

```

var x, y int
var a, b bool

```

- **x** dan **y**: Menyimpan bilangan bulat yang dimasukkan pengguna.
- **a**: Menyimpan hasil kondisi $y \% x == 0$
- **b**: Menyimpan hasil kondisi $x \% y == 0$

2. Membaca input dari pengguna:

```

fmt.Scan(&x, &y)

```

- Program meminta pengguna untuk memasukkan dua bilangan bulat x dan y.

5. Logika utama:

Kasus 1: Kondisi $y \% x == 0$:

```

if y%x == 0 {
    a = true
}

```

- Mengecek apakah y habis dibagi x.
- Jika $y \% x == 0$, maka a diset menjadi **true**, jika tidak maka tetap **false** (nilai default).

Kasus 2: Kondisi $x \% y == 0$

```

if x%y == 0 {
    b = true
}

```

- Mengecek apakah x habis dibagi y.
- Jika $x \% y == 0$, maka b diset menjadi **true**, jika tidak maka tetap **false**.

3. Menampilkan hasil:

```
fmt.Print(a, "\n", b)
```

- Program mencetak nilai boolean **a** dan **b**, masing-masing dalam baris terpisah.

Contoh: Jika pengguna memasukkan:

```
4 4(input)
```

```
true(output)
```

```
true(output)
```