

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



**DISUSUN OLEH:**  
**RAIHAN ADI ARBA**  
**103112400071**  
**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

Percabangan dalam pemrograman adalah konsep yang memungkinkan program untuk mengambil keputusan berdasarkan kondisi yang diberikan, yang merupakan bagian penting dalam pengembangan algoritma. Dalam bahasa pemrograman Go (Golang), salah satu bentuk percabangan yang paling mendasar adalah **if-then**. Struktur ini digunakan untuk mengeksekusi suatu blok kode hanya jika suatu kondisi bernilai benar. Jika kondisi tersebut bernilai salah, maka blok kode tersebut tidak akan dijalankan, dan program akan melanjutkan eksekusi ke bagian lain dari kode. Dengan demikian, percabangan **if-then** memungkinkan program untuk beradaptasi dengan berbagai kondisi dan merespons input atau status yang berbeda secara dinamis.

Dalam prakteknya, percabangan **if-then** digunakan untuk menangani berbagai situasi dalam program, seperti validasi input, pengolahan data, atau pengambilan keputusan berdasarkan hasil perhitungan atau status sistem. Keberadaannya memberikan fleksibilitas untuk mengeksekusi aksi-aksi yang berbeda berdasarkan kondisi yang bervariasi, misalnya mengecek apakah suatu nilai lebih besar atau lebih kecil dari nilai lain, apakah sebuah input memenuhi kriteria tertentu, atau apakah suatu operasi berhasil dilakukan. Dengan kata lain, percabangan **if-then** memfasilitasi pembuatan program yang lebih cerdas dan responsif terhadap berbagai skenario yang dihadapi selama eksekusi.

Secara keseluruhan, percabangan **if-then** merupakan komponen dasar dalam pengendalian alur eksekusi program. Kemampuan untuk membuat keputusan berbasis kondisi memungkinkan pengembangan algoritma yang lebih kompleks, serta memungkinkan pembuatan aplikasi yang lebih interaktif dan adaptif terhadap input atau perubahan status yang terjadi pada saat runtime. Hal ini menjadikannya sebagai salah satu konsep utama yang harus dipahami dalam pemrograman dan pengembangan perangkat lunak.

## A. GUIDED

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

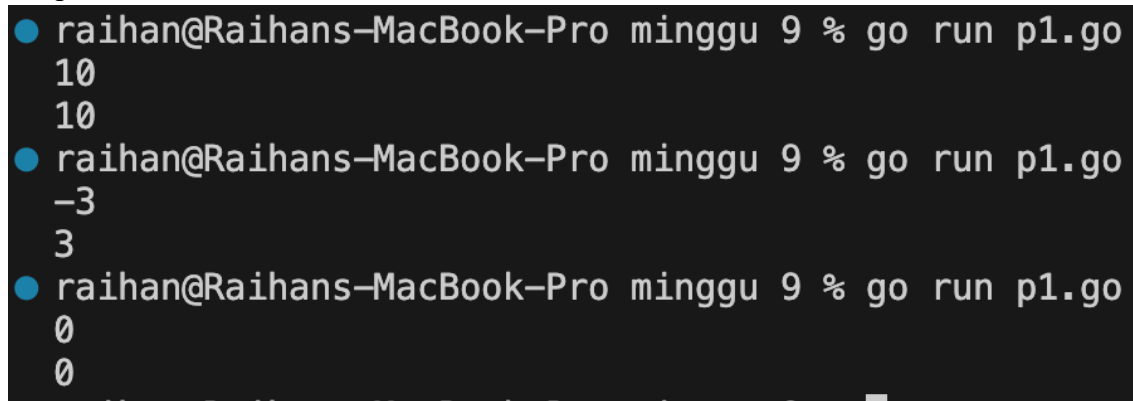
Source code :

```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)
    if n%2 == 0 {
        fmt.Println(n / 2)
    } else {
        fmt.Println((n / 2) + 1)
    }
}
```

Output :



```
● raihan@Raihans-MacBook-Pro minggu 9 % go run p1.go
10
10
● raihan@Raihans-MacBook-Pro minggu 9 % go run p1.go
-3
3
● raihan@Raihans-MacBook-Pro minggu 9 % go run p1.go
0
0
```

Deskripsi :

Program akan meminta pengguna memasukkan sebuah bilangan bulat, kemudian program akan melakukan pengecekan terhadap bilangan tersebut. Jika bilangan yang dimasukkan merondisionalngan genap, maka program akan langsung membagi bilangan tersebut dengan 2 dan menampilkan hasilnya. Namun jika bilangan yang

dimasukkan adalah ganjil, maka program akan membagi bilangan tersebut dengan 2 dan menambahkan 1 pada hasil pembagiannya sebelum menampilkan hasil akhir. Program ini menggunakan struktur kondisional if-else untuk menentukan operasi yang akan dilakukan berdasarkan sifat bilangan input (genap atau ganjil) dan menggunakan operator modulo (%) untuk mengecek apakah suatu bilangan genap atau ganjil.

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

Source code :

```
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 :

```

● raihan@Raihans-MacBook-Pro minggu 9 % go run p2.go
10
positif
● raihan@Raihans-MacBook-Pro minggu 9 % go run p2.go
-3
bukan positif
● raihan@Raihans-MacBook-Pro minggu 9 % go run p2.go
5
positif

```

Deskripsi :

Pertama, program mendeklarasikan dua variabel: bilangan bertipe integer untuk menyimpan input angka dan teks bertipe string untuk menyimpan hasil evaluasi. Program meminta input dari pengguna menggunakan `fmt.Scan()`. Secara default, variabel teks diisi dengan nilai "bukan positif". Kemudian program menggunakan struktur kondisional `if` untuk memeriksa apakah bilangan yang dimasukkan lebih besar dari 0. Jika kondisi tersebut terpenuhi ( $\text{bilangan} > 0$ ), maka nilai variabel teks akan diubah menjadi "positif". Terakhir, program akan menampilkan isi dari variabel teks yang menunjukkan hasil evaluasi bilangan tersebut.

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

Source code:

```

package main

import "fmt"

func main() {
    var v1, v2, nn, hasil int

```

```
fmt.Scan(&v1, &v2)
for nn = 1; nn <= v2; nn++ {
    hasil += hasil + v1
}
fmt.Println(hasil)
}
```

Output :

```
● raihan@Raihans-MacBook-Pro minggu 9 % go run p3.go
10
false
● raihan@Raihans-MacBook-Pro minggu 9 % go run p3.go
-3
false
● raihan@Raihans-MacBook-Pro minggu 9 % go run p3.go
0
false
```

Deskripsi :

Program mendeklarasikan empat variabel integer: v1, v2, nn, dan hasil. Pengguna diminta memasukkan dua nilai (v1 dan v2) menggunakan fmt.Scan(). Program menggunakan struktur perulangan for yang akan berjalan dari nn = 1 hingga mencapai nilai v2. Dalam setiap iterasi, program melakukan operasi penjumlahan dengan menambahkan nilai sebelumnya dari hasil dengan v1. Namun, perlu diperhatikan bahwa ada potensi kesalahan logika dalam kode ini karena pada setiap iterasi, nilai hasil ditambahkan dengan dirinya sendiri dan kemudian ditambah v1 (hasil += hasil + v1), yang mungkin tidak memberikan hasil yang diharapkan. Setelah perulangan selesai, program menampilkan nilai akhir dari variabel hasil.

## B. UNGUIDED

### 1. Latihan 1

Sebuah program Go digunakan untuk menentukan jumlah motor yang diperlukan seseorang untuk melakukan touring. Satu motor hanya cukup untuk dua orang, yaitu satu pengemudi dan satu tumpangan. Setiap motor diprioritaskan untuk terisi dua orang.

**Masukan** terdiri dari bilangan bulat yang menyatakan jumlah orang yang akan melakukan touring.

**Keluaran** berupa jumlah motor yang diperlukan peserta touring.

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	10	5
2	1	1
3	25	13
4	9	5

Source Code:

```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)
    if n%2 == 0 {
        fmt.Println(n/2)
    } else {
        fmt.Println((n/2)+1)
    }
}
```

Output :

```
● raihan@Raihans-MacBook-Pro minggu 9 % go run l1.go
10
5
● raihan@Raihans-MacBook-Pro minggu 9 % go run l1.go
1
1
● raihan@Raihans-MacBook-Pro minggu 9 % go run l1.go
9
5
```

Penjelasan Program:

Program memulai dengan mendeklarasikan variabel `n` bertipe integer yang akan menyimpan input dari pengguna melalui fungsi `fmt.Scan()`. Selanjutnya, program menggunakan struktur kondisional `if-else` untuk memeriksa apakah bilangan tersebut genap atau ganjil dengan menggunakan operator modulo (`%`). Jika bilangan tersebut genap (`n%2 == 0`), maka program akan membagi bilangan tersebut dengan 2 (`n/2`) dan menampilkan hasilnya. Namun jika bilangan tersebut ganjil, program akan membagi bilangan dengan 2 dan menambahkan 1 ke hasil pembagian (`(n/2)+1`) sebelum menampilkannya. Program ini berguna untuk kasus-kasus di mana diperlukan pembagian bilangan dengan perlakuan berbeda untuk bilangan genap dan ganjil.

## 2. Latihan 2

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 teks yang menyatakan bilangan adalah "genap negatif" atau "bukan".

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	10	bukan
2	-4	genap negatif
3	0	bukan
4	-2	genap negatif

Source Code:

```
package main

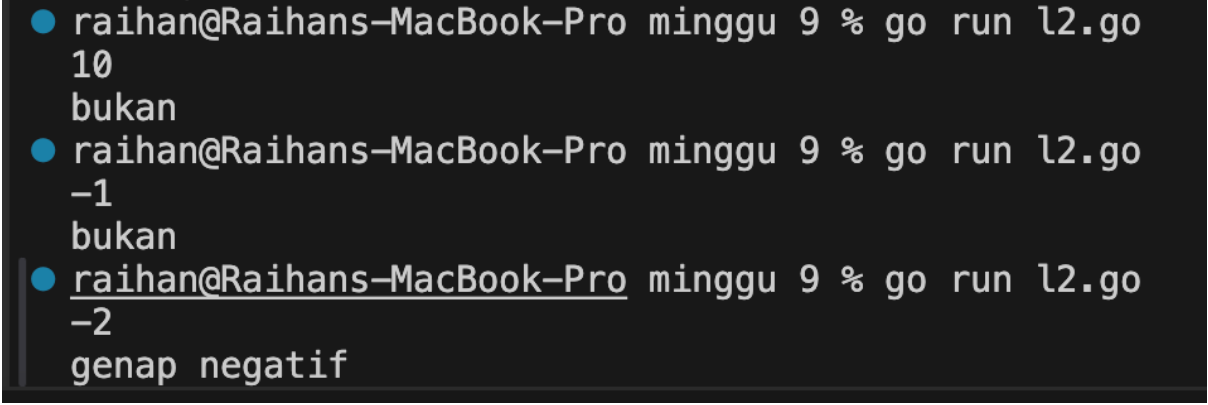
import "fmt"

func main() {
    var n int
    var txt string
    fmt.Scan(&n)
    txt = "bukan"
    if n%2 == 0 && n<0 {
```



```
        teks = "genap negatif"
    }
    fmt.Println(txt)
}
```

Output:



```
● raihan@Raihans-MacBook-Pro minggu 9 % go run l2.go
10
bukan
● raihan@Raihans-MacBook-Pro minggu 9 % go run l2.go
-1
bukan
● raihan@Raihans-MacBook-Pro minggu 9 % go run l2.go
-2
genap negatif
```

Deskripsi Program:

Program mendeklarasikan dua variabel: `n` bertipe integer untuk menyimpan input bilangan dan `txt` bertipe string yang diinisialisasi dengan nilai default "bukan". Program menggunakan `fmt.Scan()` untuk menerima input dari pengguna. Kemudian program menggunakan struktur kondisional `if` dengan dua kondisi yang digabungkan menggunakan operator logika AND (`&&`):

`n%2 == 0` - mengecek apakah bilangan genap

`n < 0` - mengecek apakah bilangan negatif

Namun, terdapat kesalahan dalam kode tersebut dimana variabel `teks` dalam blok `if` tidak sesuai dengan deklarasi variabel `txt` di awal program. Seharusnya menggunakan `txt = "genap negatif"` untuk konsistensi nama variabel. Program akan menampilkan "genap negatif" jika bilangan yang dimasukkan adalah bilangan genap dan negatif, atau menampilkan "bukan" untuk kondisi lainnya.

### 3. Latihan 3

Buatlah program yang digunakan untuk menentukan suatu bilangan adalah faktor dari bilangan yang lain. Suatu bilangan `a` adalah faktor dari `b` apabila bilangan `a` habis membagi bilangan `b`.

**Masukan** terdiri dari dua bilangan bulat positif `x` dan `y`.

**Keluaran** terdiri dari dua baris boolean, yang mana baris pertama adalah boolean yang menyatakan `x` adalah faktor dari `y`. Baris kedua adalah boolean yang menyatakan `y` adalah faktor dari `x`.

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	10 5	false true
2	3 21	true false
3	4 4	true true

Source Code:

```
package main

import "fmt"

func main() {
    var x, y int
    var result bool
    fmt.Scan(&x, &y)
    if y%x == 0 {
        result = true
    } else {
        result = false
    }
    fmt.Println(result)

    if x%y == 0 {
        result = true
    } else {
        result = false
    }
    fmt.Println(result)
}
```

Output:

```
● raihan@Raihans-MacBook-Pro minggu 9 % go run l3.go
10 5
false
true
● raihan@Raihans-MacBook-Pro minggu 9 % go run l3.go
3 4
false
false
```

#### Deskripsi Program:

Program di atas dirancang untuk mengeksekusi operasi perpangkatan dengan memanfaatkan input dari pengguna. Setelah dijalankan, program ini meminta pengguna untuk memasukkan dua nilai integer, X dan Y. Nilai X akan dijadikan sebagai basis, sedangkan Y merupakan eksponen. Program ini menggunakan sebuah loop for untuk mengulang operasi perkalian sebanyak Y kali. Dalam setiap iterasi, nilai X dikalikan dengan variabel hasil, yang semula diinisialisasi dengan nilai 1. Proses ini berlanjut hingga loop selesai.

#### DAFTAR PUSTAKA

**Prayogo, N. A. (2021). *Dasar Pemrograman Go. Ebook***