

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 10**  
**“IF ELSE”**



**DISUSUN OLEH:**  
**RIZKINA AZIZAH**  
**103112400082**  
**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

Dalam Go, kita dapat menggunakan pernyataan kondisional untuk mengeksekusi kode berdasarkan kondisi yang berbeda. Kita juga dapat menggunakan pernyataan `if`, `else if`, dan `if else` untuk mengeksekusi kode yang diberikan jika suatu kondisi benar atau salah.

- Pernyataan **if** - mengeksekusi beberapa kode jika satu kondisi benar
- Pernyataan **if...else** - mengeksekusi beberapa kode jika suatu kondisi benar dan kode lain jika kondisi tersebut salah
- Pernyataan **if...else if....else** - mengeksekusi kode yang berbeda untuk lebih dari dua kondisi
- Pernyataan **switch...case** - memilih salah satu dari banyak blok kode yang akan dieksekusi

Seleksi kondisi digunakan untuk mengontrol alur eksekusi flow program. Analoginya mirip seperti fungsi rambu lalu lintas di jalan raya. Kapan kendaraan diperbolehkan melaju dan kapan harus berhenti diatur oleh rambu tersebut. Seleksi kondisi pada program juga kurang lebih sama, kapan sebuah blok kode dieksekusi dikontrol. Yang dijadikan acuan oleh seleksi kondisi adalah nilai bertipe `bool`, bisa berasal dari variabel, ataupun hasil operasi perbandingan. Nilai tersebut menentukan blok kode mana yang akan dieksekusi. Go memiliki 2 macam keyword untuk seleksi kondisi, yaitu **if else** dan **switch**.

### **a. Seleksi Kondisi Menggunakan Keyword `if`, `else if`, & `else`**

Cara penerapan `if-else` di Go sama seperti pada bahasa pemrograman lain. Yang membedakan hanya tanda kurungnya (*parentheses*), di Go tidak perlu ditulis.

### **b. Variabel Temporary Pada `if – else`**

Variabel temporary adalah variabel yang hanya bisa digunakan pada deretan blok seleksi kondisi di mana ia ditempatkan. Penggunaan variabel ini membawa beberapa manfaat, antara lain:

- Scope atau cakupan variabel jelas, hanya bisa digunakan pada blok seleksi kondisi itu saja
- Kode menjadi lebih rapi
- Ketika nilai variabel tersebut didapat dari sebuah komputasi, perhitungan tidak perlu dilakukan di dalam blok masing-masing kondisi.

### **c. Seleksi Kondisi Menggunakan Keyword switch – case**

Switch merupakan seleksi kondisi yang sifatnya fokus pada satu variabel, lalu kemudian di-cek nilainya. Contoh sederhananya seperti penentuan apakah nilai variabel x adalah: 1, 2, 3, atau lainnya.

### **d. Pemanfaatan case Untuk Banyak Kondisi**

Sebuah case dapat menampung banyak kondisi. Cara penerapannya yaitu dengan menuliskan nilai pembanding-pembanding variabel yang di-switch setelah keyword case dipisah tanda koma (,).

### **e. Kurung Kurawal Pada Keyword case & default**

Tanda kurung kurawal ( { } ) bisa diterapkan pada keyword case dan default. Tanda ini opsional, boleh dipakai boleh tidak. Bagus jika dipakai pada blok kondisi yang di dalamnya ada banyak statement, dengannya kode akan terlihat lebih rapi.

### **f. Switch Dengan Gaya if - else**

Uniknya di Go, switch bisa digunakan dengan gaya ala if-else. Nilai yang akan dibandingkan tidak dituliskan setelah keyword switch, melainkan akan ditulis langsung dalam bentuk perbandingan dalam keyword case.

### **g. Penggunaan Keyword fallthrough Dalam switch**

Seperti yang sudah dijelaskan sebelumnya, bahwa switch pada Go memiliki perbedaan dengan bahasa lain. Ketika sebuah case terpenuhi, pengecekan kondisi tidak akan diteruskan ke case-case setelahnya.

Keyword fallthrough digunakan untuk memaksa proses pengecekan tetap diteruskan ke case selanjutnya dengan **tanpa menghiraukan nilai kondisinya**, efeknya adalah case di pengecekan selanjutnya selalu dianggap true (meskipun aslinya bisa saja kondisi tersebut tidak terpenuhi, akan tetap dianggap true).

### **h. Seleksi Kondisi Bersarang**

Seleksi kondisi bersarang adalah seleksi kondisi, yang berada dalam seleksi kondisi, yang mungkin juga berada dalam seleksi kondisi, dan seterusnya. Seleksi kondisi bersarang bisa dilakukan pada if - else, switch, ataupun kombinasi keduanya.

## CONTOH SOAL

### 1. Latihan1

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

Source Code:

```
package main

import "fmt"

func main(){
    var usia int
    var kk bool
    fmt.Scan(&usia, &kk)
    if usia >= 17 && kk {
        fmt.Println("bisa membuat ktp")
    }else{
        fmt.Println("belum bisa membuat ktp")
    }
}
```

Output:

```
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\ktp.go"
17
true
bisa membuat ktp
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\ktp.go"
20
false
belum bisa membuat ktp
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\ktp.go"
15
true
belum bisa membuat ktp
```

Deskripsi Program:

- Program ini digunakan untuk menentukan apakah seorang penduduk bisa membuat KTP atau tidak. Syarat utama membuat KTP adalah berusia minimal 17 tahun dan memiliki kartu keluarga
- Mendeklarasikan dua variabel yaitu **usia** dengan tipe data integer dan **kk** dengan tipe data boolean
- Intruksi Scan digunakan untuk mendeklarasikan input **kk**
- Menggunakan paradigma percabangan yaitu if-else :  
if sebagai kondisi atau syarat = jika usia  $\geq 17$  dan true  
then ( { } ) sebagai aksi = maka outputnya “bisa membuat KTP”  
apabila syarat tidak terpenuhi akan menjalankan aksi lain atau disebut dengan  
else = maka outputnya “belum bisa membuat KTP”

## 2. Latihan1

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

Source Code:

```
package main

import "fmt"

func main(){
    var x rune
    var huruf, vkecil, vbesar bool
    fmt.Scanf("%c", &x)
    huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
    vkecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
    vbesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
    if huruf && (vkecil || vbesar) {
        fmt.Println("vokal")
    } else if huruf && !(vkecil || vbesar) {
        fmt.Println("konsonan")
    } else {
        fmt.Println("bukan huruf")
    }
}

// tidak pake scanf bisa pake string.Tolower, hurufnya cukup deskripsikan antara kecil
atau besar aja
```

Output:

```
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\alphabet.go"
A
vokal
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\alphabet.go"
f
konsonan
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\alphabet.go"
1
bukan huruf
```

#### Deskripsi Program:

- Program ini digunakan untuk menentukan suatu alfabet yang diberikan adalah vokal atau konsonan.
- Mendeklarasikan tiga variabel yaitu **x** dengan tipe data character dan **huruf**, **vkecil**, **vbesar** dengan tipe data boolean
- Intruksi Scan digunakan untuk mendeklarasikan input **x**
- Dengan ketentuan :  
Variabel huruf berisi huruf a-z baik huruf kecil atau besar  
Variabel vkecil berisi huruf vocal kecil  
Variabel vbesar berisi huruf vocal besar
- Menggunakan paradigma percabangan yaitu if-else :
  - If kondisi atau syarat = jika huruf dan berupa vkecil atau vbesar  
then ( { } ) sebagai aksi = maka outputnya “vocal”
  - apabila syarat pertama tidak terpenuhi akan menjalankan syarat dan aksi lain atau disebut dengan else if = jika huruf dan tidak berupa vkecil atau vbesar aksi lain = maka outputnya “konsonan”
  - apabila syarat pertama dan kedua tidak terpenuhi maka akan menjalankan aksi lain = maka outputnya “bukan huruf”

#### 3. Latihan1

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: 2489 Digit pada bilangan 2489 terurut membesar
2	Bilangan: 3861 Digit pada bilangan 3861 tidak terurut
3	Bilangan: 9651 Digit pada bilangan 9651 terurut mengecil

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

Source Code:

```
package main

import "fmt"

func main(){
    var bilangan, d1, d2, d3 , d4 int
    var teks string
    fmt.Print("bilangan: ")
    fmt.Scan(&bilangan)
    d4 = bilangan % 10
    d3 = (bilangan % 100) / 10
    d2 = (bilangan % 1000) / 100
    d1 = bilangan / 1000

    if d1 < d2 && d2 < d3 && d3 < d4 {
        teks = "terurut membesar"
    }else if d1 > d2 && d2 > d3 && d3 > d4 {
        teks = "terurut mengecil"
    }else{
        teks = "tidak terurut"
    }
    fmt.Println(" Digit pada bilangan", bilangan, teks)
}

// 1234 / 1000 = 1
// 1234 % 100
```



Output:

```
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\bil.go"
bilangan: 2489
Digit pada bilangan 2489 terurut membesar
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\bil.go"
bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\bil.go"
bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
```

Deskripsi Program:

- Program ini digunakan 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
- Mendeklarasikan lima variabel yaitu `d1`, `d2`, `d3`, `d4`, `bilangan` dengan tipe data integer dan `teks` dengan tipe data boolean
- Intruksi Scan digunakan untuk mendeklarasikan input `bilangan`
- Menggunakan paradigma percabangan yaitu if-else :
  - If kondisi atau syarat = jika `d1` kurang dari `d2` dan `d2` kurang dari `d3` dan `d3` kurang dari `d4`  
then ( { } ) sebagai aksi = maka “terurut membesar”
  - apabila syarat pertama tidak terpenuhi akan menjalankan syarat dan aksi lain atau disebut dengan else if = jika `d1` lebih dari `d2` dan `d2` lebih dari `d3` dan `d3` lebih dari `d4`  
aksi lain = maka “terurut membesar”
  - apabila syarat pertama dan kedua tidak terpenuhi maka akan menjalankan aksi lain = maka “tidak terurut”
- Intruksi Print digunakan untuk mendeklarasikan output aksi yaitu teks

## SOAL LATIHAN

### Statement perulangan

#### 1. Latihan 1

PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, **buatlah program BiayaPos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut!**

Dari berat parsel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

NO	Contoh Masukan dan Keluaran
1	Berat parsel (gram): <b>8500</b> Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500
2	Berat parsel (gram): <b>9250</b> Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750
3	Berat parsel (gram): <b>11750</b> Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000 + Rp. 3750 Total biaya: Rp. 110000

#### Source Code:

```
package main

import "fmt"

func main(){
    var bp, tambah int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&bp)
    kg := bp/1000
    sisa_gr := bp%1000
    kirim := kg * 10000
```

```

if kg > 10 {
    tambah = 0
}else if sisa_gr >= 500 {
    tambah = sisa_gr * 5
}else if sisa_gr < 500 {
    tambah = sisa_gr * 15
}
total := kirim + tambah
fmt.Printf("Detail Berat: %d kg + %d gr\n ", kg, sisa_gr)
fmt.Printf("Detail Biaya: Rp. %d kg + Rp. %d\n", kirim, tambah)
fmt.Printf("Total biaya: Rp. %d\n ", total)
}

```

Output:

```

PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\satu.go"
Berat parsel (gram): 8500
Detail Berat: 8 kg + 500 gr
Detail Biaya: Rp. 80000 kg + Rp. 2500
Total biaya: Rp. 82500

PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\satu.go"
Berat parsel (gram): 9250
Detail Berat: 9 kg + 250 gr
Detail Biaya: Rp. 90000 kg + Rp. 3750
Total biaya: Rp. 93750

PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\satu.go"
Berat parsel (gram): 11750
Detail Berat: 11 kg + 750 gr
Detail Biaya: Rp. 110000 kg + Rp. 0
Total biaya: Rp. 110000

```

Deskripsi Program:

- Program ini digunakan untuk menghitung biaya pengiriman
- Mendeklarasikan dua variabel yaitu **bp** dan **tambah** dengan tipe data integer
- Intruksi Scan digunakan untuk mendeklarasikan input **bp**
- Menggunakan paradigma percabangan yaitu if-else :
  - If kondisi atau syarat = jika berat kilo lebih dari 10  
then ( {} ) sebagai aksi = maka tambah = 0
  - apabila syarat pertama tidak terpenuhi akan menjalankan syarat dan aksi lain  
atau disebut dengan else if = jika sisa gram lebih dari 500  
aksi lain = maka tambah = sisa gram \* 5

- apabila syarat pertama dan kedua tidak terpenuhi maka akan menjalankan aksi lain = maka tambah = sisa gram \* 15
- Menggunakan intruksi Printf sebagai deklarasi output agar otomatis tertulis kg ataupun gr pada hasil keluaran

## 2. Latihan 2

Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian nilai mata kuliah (NMK) sebagai berikut:

NAM	NMK
NAM > 80	A
72.5 < NAM <= 80	AB
65 < NAM <= 72.5	B
57.5 < NAM <= 65	BC
50 < NAM <= 57.5	C
40 < NAM <= 50	D
NAM <= 40	E

Program berikut menerima input sebuah bilangan riil yang menyatakan NAM. Program menghitung NMK dan menampilkannya.

### Source Code:

```
package main
import "fmt" func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scan(&nam)
    if nam > 80 {
        nam = "A"
    }
    if nam > 72.5 {
        nam = "AB"
    }
    if nam > 65 {
        nam = "B"
    }
}
```

```

if nam > 57.5 {
    nam = "BC"
}
if nam > 50 {
    nam = "C"
}
if nam > 40 {
    nam = "D"
}
else if nam <= 40 {
    nam = "E"
}
fmt.Println("Nilai mata kuliah: ", nmk)
}

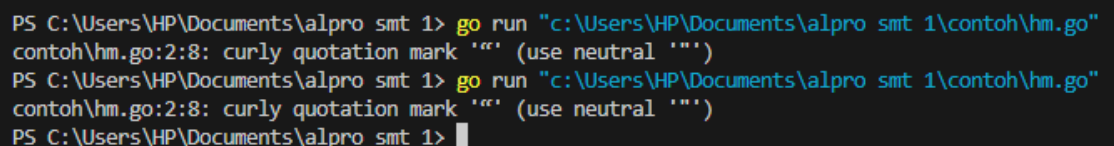
```

Jawablah pertanyaan-pertanyaan berikut:

- Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?
- Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!
- Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Jawaban :

- Keluaran nya error



```

PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\contoh\hm.go"
contoh\hm.go:2:8: curly quotation mark "'" (use neutral "'')
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\contoh\hm.go"
contoh\hm.go:2:8: curly quotation mark "'" (use neutral "'')
PS C:\Users\HP\Documents\alpro smt 1>

```

Keluaran tersebut tidak sesuai dengan spesifikasi soal

- Kesalahan program
  - Menggunakan tanda kutip tanda kutip miring ganda dari awal seharusnya menggunakan tanda kutip ganda lurus

2. Pada variabel terdeklarasi bahwa variabel nam berisi angka namun pada codingan diisi huruf maka seharusnya pada nam yang berisi huruf diganti menjadi nmk
3. Pada paradigma percabangan if apabila ingin menggunakan aksi lain dan tetap berada satu kondisi maka codingan berikutnya menggunakan else if, bukan hanya if saja. Jika hanya menggunakan if saja akan menyebabkan menjadi banyak kondisi

#### ALUR PROGRAM:

1. Baca input berupa nam yang berisi nilai
2. Eksekusi menggunakan if – else if kriteria mana yang sesuai dengan inputan
3. Eksekusi output berupa nmk yang telah ditentukan (A atau B atau AB dsb)

#### c. Perbaikan program

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scan(&nam)
    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "AB"
    } else if nam > 65 {
        nmk = "B"
    } else if nam > 57.5 {
        nmk = "BC"
    } else if nam > 50 {
        nmk = "C"
    } else if nam > 40 {
        nmk = "D"
    }
```

```

    } else {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

Uji Program :

```

PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D

```

### 3. Latihan 3

Sebuah bilangan bulat **b** memiliki faktor bilangan **f** > 0 jika f habis membagi **b**. Contoh: 2 merupakan faktor dari bilangan 6 karena 6 habis dibagi 2. Buatlah program yang menerima input sebuah bilangan bulat **b** dan **b** > 1. Program harus dapat mencari dan menampilkan semua faktor dari bilangan tersebut! Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

Bilangan: 12	Bilangan: 7
Faktor: 1 2 3 4 6 12	Faktor: 1 7

Bilangan bulat **b** > 0 merupakan bilangan prima **p** jika dan hanya jika memiliki persis dua faktor bilangan saja, yaitu 1 dan dirinya sendiri. Lanjutkan program sebelumnya. Setelah menerima masukan sebuah bilangan bulat **b** > 0. Program tersebut mencari dan menampilkan semua faktor bilangan tersebut. Kemudian, program menentukan apakah **b** merupakan bilangan prima. Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

Bilangan: 12	Bilangan: 7
Faktor: 1 2 3 4 6 12	Faktor: 1 7
Prima: false	Prima: true

Source Code:

```

package main

import "fmt"

```

```

func main() {
    var b, faktor int
    fmt.Print("Bilangan: ")
    fmt.Scan(&b)
    fmt.Print("faktor: ")
    faktor = 0
    for i := 1; i <= b; i++ {
        if b % i == 0 {
            faktor += 1
            fmt.Printf("%d ", i)
        }
    }
    fmt.Println(" ")

    prima := "false"
    if faktor == 2 {
        prima = "true"
    }
    fmt.Print("Prima: ", prima)
}

```

Output:

```

Bilangan: 12
faktor: 1 2 3 4 6 12
Prima: false
PS C:\Users\HP\Documents\alpro smt 1> go run "c:\Users\HP\Documents\alpro smt 1\modul 10\tiga.go"
Bilangan: 7
faktor: 1 7
Prima: true

```

Deskripsi Program:

- Program ini digunakan untuk menentukan factor suatu bilangan dan menentukan apakah bilangan tersebut bilangan prima atau bukan.
- Mendeklarasikan dua variabel yaitu **b** dan **faktor** dengan tipe data integer
- Intruksi Scan digunakan untuk mendeklarasikan input **b**
- Menggunakan paradigma perulangan untuk menentukan factor suatu bilangan



- Menggunakan paradigma percabangan yaitu if-then :
  - if kondisi pertama = jika b habis dibagi i  
then ( {} ) sebagai aksi = maka i adalah factor b  
menghasilkan output dari perulangan
  - if kondisi kedua = jika jumlah perulangan (faktor) ada 2 angka (misal : 1  
17)  
then ( {} ) sebagai aksi = maka output Prima = true

## DAFTAR PUSTAKA

Golang Program, Golang If...Else...Else If Statements

(<https://www.golangprograms.com/golang-if-else-statements.html> )

Novalagung(2024), A.13. Seleksi Kondisi

(<https://dasarpemrogramangolang.novalagung.com/A-seleksi-kondisi.html>,2024)

Akankah Velida(2022), Pernyataan If/Else dan Kondisional dalam Go

(<https://dev.to/willvelida/ifelse-and-conditional-statements-in-go-2k9>,2024)