

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



**DISUSUN OLEH:**  
**ANGGUN WAHYU WIDIYANA**  
**103112480280**  
**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

### 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**.

### 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 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**.
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:

Tanpa Perulangan	Dengan Perulangan
<pre>If kondisi then     // aksi else     // aksi lain endfor</pre>	<pre>If kondisi {; kondisi; update {     // 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>If kondisi then     // aksi else     if kondisi_2 then</pre>	<pre>If kondisi {; kondisi; update {     // aksi }else{     if kondisi_2 {</pre>

---

<pre> // aksi 2 else     if kondisi_3 then         // aksi 3     else         // aksi lain     endfor endfor endfor </pre>	<pre> // aksi 2 }else{     if kondisi_3 {         // aksi 3     }else{         // aksi lain     } } } </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> if kondisi_1 then     // aksi 1 else if kondisi_2 then     // aksi 2 else if kondisi_3 then     // aksi 3 else     // aksi lain endfor </pre>	<pre> if kondisi_1 {     // aksi 1 }else if kondisi_2 {     // aksi 2 }else if kondisi_3 {     // aksi 3 }else{     // aksi lain } </pre>

## 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: 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     }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,
11 mendapat indeks %c\n", nilai, pctHadir, adaTubes, indeks )
```

---

## CONTOH SOAL

### 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. nilai absolut atau mutlak dari bilangan 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\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso1.go"
17
true
bisa membuat KTP
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso1.go"
20
false
belum bisa membuat KTP
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso1.go"
15
true
belum bisa membuat KTP
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso1.go"
```

Deskripsi Program:

Program Go ini bertujuan menentukan apakah seseorang memenuhi syarat untuk membuat KTP berdasarkan dua kriteria: usia minimal 17 tahun dan memiliki kartu keluarga (KK). Jika kedua syarat terpenuhi, program mencetak "bisa membuat KTP"; jika tidak, program mencetak "belum bisa membuat KTP". Program menerima masukan berupa usia (bilangan bulat) dan status kepemilikan KK (boolean).

---

## Latihan2

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")
    }
}
```

Output:

```
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso2.go"
A
vokal
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso2.go"
f
konsonan
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso2.go"
1
bukan huruf
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso2.go"
$
bukan huruf
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso2.go"
```

Deskripsi Program:

Program Go ini menentukan apakah karakter masukan adalah vokal, konsonan, atau bukan huruf. Masukan berupa satu karakter. Program memeriksa apakah karakter termasuk alfabet (huruf kecil atau besar) dan apakah karakter tersebut vokal. Hasilnya berupa teks yang mencetak "vokal", "konsonan", atau "bukan huruf".

---



### Latihan3

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

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 / 10) % 10
    d2 = (bilangan / 100) % 10
    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)
}
```

Output:

```
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso3.go"
Bilangan: 2489
Digit pada bilangan 2489 terurut membesar
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso3.go"
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso3.go"
Bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Cosomod10\Coso3.go"
```

Deskripsi Program:

Program Go ini digunakan untuk menentukan pola urutan digit dalam bilangan empat digit. Masukan berupa bilangan bulat antara 1000 hingga 9999. Program memisahkan setiap digit bilangan dan mengevaluasi apakah digit-digit tersebut terurut membesar (misalnya: 1234), terurut mengecil (misalnya: 4321), atau tidak terurut (misalnya: 1324).

---

## SOAL LATIHAN

### Latihan1

Sebuah 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 parcel (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 parcel (gram): 8500 Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500
2	Berat parcel (gram): 9250 Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750
3	Berat parcel (gram): 11750 Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000 + Rp. 3750 Total biaya: Rp. 113750

Source Code:

```
package main
import "fmt"
func main(){
    var berat, sisa int
    fmt.Print("Berat parcel (gram): ")
    fmt.Scanln(&berat)

    parcel := berat / 1000
    berat_after := berat % 1000
    biaya := parcel * 10000

    if berat_after < 500 {
        sisa = berat_after * 15
    } else {
        sisa = berat_after * 5
    }
}
```

```
        fmt.Printf("Detail berat: %d kg + %d gr\n", parsel,
berat_after)
        fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biaya, sisa)
        fmt.Printf("Total biaya: Rp. %d\n", biaya+sisa)
    }
}
```

### Output:

```
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod1.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod1.go"
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod1.go"
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 113750
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod1.go"
Berat parsel (gram): 1
```

### Deskripsi Program:

Program Go ini menghitung biaya pengiriman parsel berdasarkan berat dalam gram. Berat parsel pertama-tama dihitung dalam kilogram (kg) dan sisa gramnya. Biaya pengiriman dihitung sebagai berikut:

- ✓ Rp 10.000,- per kg untuk setiap kilogram.
- ✓ Rp 5,- per gram jika sisa berat lebih dari atau sama dengan 500 gram.
- ✓ Rp 15,- per gram jika sisa berat kurang dari 500 gram.

Jika total berat lebih dari 10 kg, biaya untuk sisa berat (kurang dari 1 kg) digratiskan.

## Latihan2

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.

```
1 package main
2 import "fmt"
3 func main() {
4     var nam float64
5     var nmk string
6     fmt.Print("Nilai akhir mata kuliah: ")
7     fmt.Scan(&nam)
8     if nam > 80 {
9         nmk = "A"
10    }
11    if nam > 72.5 {
12        nmk = "AB"
13    }
14    if nam > 65 {
15        nmk = "B"
16    }
17    if nam > 57.5 {
18        nmk = "BC"
19    }
20    if nam > 50 {
21        nmk = "C"
22    }
23    if nam > 40 {
24        nmk = "D"
25    } else if nam <= 40 {
26        nmk = "E"
27    }
28    fmt.Println("Nilai mata kuliah: ", nmk)
29 }
```

Jawablah pertanyaan-pertanyaan berikut:

---

- a. Jika **nam** diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?

Jawab:

Berdasarkan tabel nilai diatas, nilai 80.1 masuk dalam kategori "AB" karena berada di antara  $72.5 < \text{NAM} \leq 88$ . Tetapi, program akan menghasilkan output "B" karena kondisi if dalam programnya tidak sesuai dengan tabel. Jadi eksekusi program tidak sesuai dengan spesifikasi soal.

- b. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!

Jawab:

Kesalahan dari program tersebut adalah:

- ✓ Kondisi if tidak lengkap dan tidak sesuai dengan range nilai pada tabel
  - ✓ Urutan pengecekan kondisi tidak tepat (seharusnya dari nilai tertinggi ke terendah)
  - ✓ Beberapa range nilai tidak diatur dengan benar
- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'!

Jawab:

Perbaiki program yang benar ada dibawah (Source Code)

Output (Screenshot Uji ada dibawah Source Code)

93.5 akan menghasilkan output "A" (karena  $> 88$ )

70.6 akan menghasilkan output "B" (karena antara  $65 < \text{NAM} \leq 72.5$ )

49.5 akan menghasilkan output "D" (karena antara  $40 < \text{NAM} \leq 50$ )

Source Code:

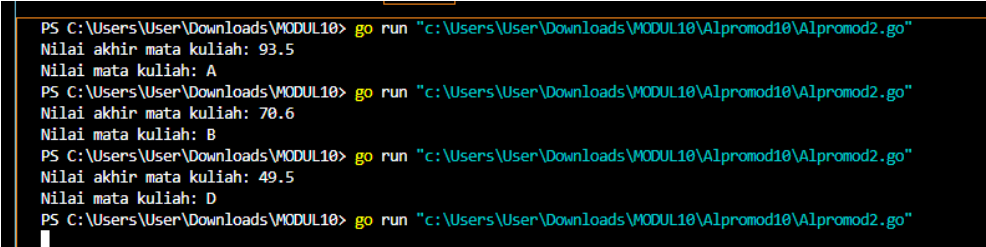
```
package main
import "fmt"
func main(){
    var nam float64
    var nmk string

    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scan(&nam)

    if nam > 88 {
        nmk = "A"
    } else if nam > 72.5 && nam <= 88 {
        nmk = "AB"
    } else if nam > 65 && nam <= 72.5 {
        nmk = "B"
    } else if nam > 57.5 && nam <= 65 {
        nmk = "BC"
    } else if nam > 50 && nam <= 57.5 {
        nmk = "C"
    } else if nam > 40 && nam <= 50 {
        nmk = "D"
    } else {
        nmk = "E"
    }
}
```

```
}  
  
    fmt.Printf("Nilai mata kuliah: %s\n", nmk)  
}
```

Output:



```
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod2.go"  
Nilai akhir mata kuliah: 93.5  
Nilai mata kuliah: A  
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod2.go"  
Nilai akhir mata kuliah: 70.6  
Nilai mata kuliah: B  
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod2.go"  
Nilai akhir mata kuliah: 49.5  
Nilai mata kuliah: D  
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod2.go"
```

Deskripsi Program:

Program Go digunakan untuk menentukan nilai huruf (grade) berdasarkan nilai akhir mata kuliah yang diinputkan.

---

### Latihan3

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: <u>12</u> Faktor: 1 2 3 4 6 12	Bilangan: <u>7</u> 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: <u>12</u> Faktor: 1 2 3 4 6 12 Prima: false	Bilangan: <u>7</u> Faktor: 1 7 Prima: true
-------------------------------------------------------------	--------------------------------------------------

Source Code:

```
package main
import "fmt"
func main(){
    var b int
    fmt.Scanln(&b)

    if b <= 1 {
        fmt.Println("Bilangan harus lebih besar dari 1.")
    }
    fmt.Printf("Bilangan: %d\nFaktor: ", b)
    prima := true

    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Printf("%d ", i)
            if i != 1 && i != b {
                prima = false
            }
        }
    }
    fmt.Printf("\nPrima: %v\n", prima)
}
```



Output:

```
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod3.go"
Bilangan: 12
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod3.go"
Bilangan: 7
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Users\User\Downloads\MODUL10> go run "c:\Users\User\Downloads\MODUL10\Alpromod10\Alpromod3.go"
```

Deskripsi Program:

Program Go ini menghitung faktor-faktor dari suatu bilangan dan menentukan apakah bilangan tersebut prima atau bukan berdasarkan jumlah faktor yang dimiliki. Jika hanya memiliki dua faktor (1 dan dirinya sendiri), maka bilangan tersebut adalah prima.

---

## **DAFTAR PUSTAKA**

School of Computing. *Modul Praktikum 10 – Else-If Algoritma dan Pemrograman*  
*1 S1 Informatika..2024*