

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



**Disusun oleh :**  
**NAMA : Felix Pedrosa Valentino**  
**NIM : 103112400056**  
**S1 IF – 12 - 01**

**Dosen Pengampu :**  
**Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024/2025**

## DASAR TEORI

### Percabangan

- Setiap baris instruksi dieksekusi satu persatu
- Adanya instruksi bersyarat (kondisi). Instruksi dieksekusi apabila memenuhi kondisi atau syarat tertentu
- Konstruksi dalam algoritma yang memungkinkan kita untuk melakukan pilihan instruksi yang berbeda-beda sesuai dengan kondisi yang dihadapi

### Kondisi

⇒ Ekspresi yang bernilai TRUE atau FALSE (Boolean) Operasi Perbandingan atau Logika

### Aksi

⇒ Kumpulan instruksi/ekspresi yang akan dieksekusi apabila kondisi bernilai TRUE

⇒ Antara kondisi dan aksi terdapat suatu hubungan berpasangan

### Paradigma Percabangan

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 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.

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.

## CONTOH SOAL

### 1.) Contoh Soal 1

Source Code :

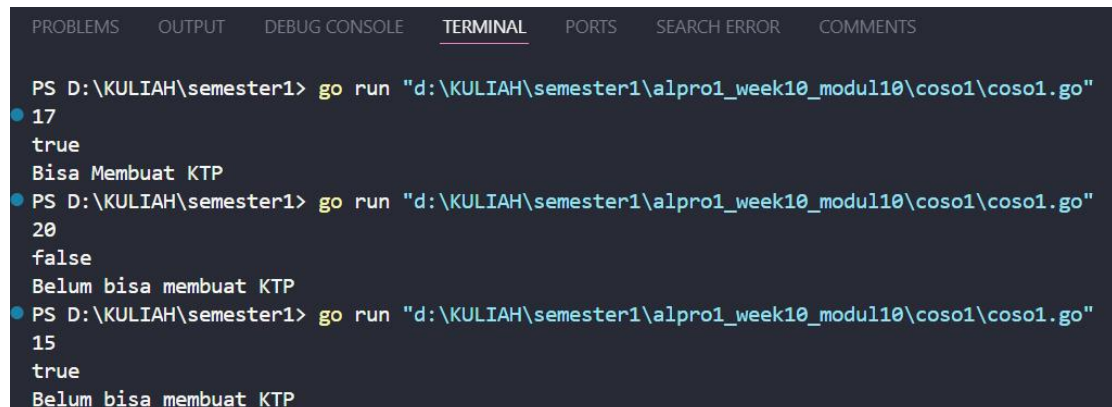
```
package main

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 :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS

PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso1\coso1.go"
17
true
Bisa Membuat KTP
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso1\coso1.go"
20
false
Belum bisa membuat KTP
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso1\coso1.go"
15
true
Belum bisa membuat KTP
```

Deskripsi Program :

Program di atas adalah program yang mengecek kelayakan seseorang untuk membuat KTP berdasarkan usia dan status kepemilikan Kartu Keluarga (KK). Program ini dimulai dengan mendeklarasikan dua variabel : usia bertipe integer dan kk bertipe boolean. Variabel usia akan menyimpan usia pengguna, sedangkan variabel kk akan menyimpan nilai

boolean yang menunjukkan apakah pengguna memiliki Kartu Keluarga atau tidak.

Fungsi `fmt.Scan(&usia, &kk)` digunakan untuk menerima input dari pengguna. Pengguna diminta memasukkan usia (sebagai integer) dan status kepemilikan KK (sebagai nilai `true` atau `false`).

Setelah input diterima, terdapat struktur kondisi `if` yang mengecek apakah usia lebih besar atau sama dengan 17 dan nilai `kk` adalah `true`. Jika kedua kondisi ini terpenuhi, program akan mencetak "Bisa Membuat KTP", yang berarti pengguna layak membuat KTP. Jika salah satu atau kedua kondisi tidak terpenuhi, maka program akan mencetak "Belum bisa membuat KTP", menandakan bahwa pengguna belum memenuhi syarat untuk membuat KTP.

## 2.) Contoh Soal 2

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 :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS

PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso2\coso2.go"
A
Vokal
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso2\coso2.go"
f
Konsonan
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso2\coso2.go"
1
Bukan Huruf
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso2\coso2.go"
$
Bukan Huruf
```

### Deskripsi Program :

Program di atas adalah program yang bertujuan untuk mengidentifikasi jenis karakter yang diinput pengguna, apakah itu vokal, konsonan, atau bukan huruf.

Pada awal program, variabel dideklarasikan, x sebagai variabel bertipe rune untuk menyimpan karakter yang diinput, variabel huruf, vKecil, dan vBesar bertipe boolean yang digunakan untuk menyimpan kondisi terkait karakter yang diinput.

Selanjutnya, program memeriksa apakah karakter tersebut merupakan huruf dengan menggunakan kondisi yang memeriksa rentang nilai ASCII untuk huruf kecil dan huruf besar. Kemudian, program juga memeriksa apakah karakter tersebut adalah huruf vokal, baik dalam bentuk kecil (a, i, u, e, o) maupun besar (A, I, U, E, O). Berdasarkan hasil pemeriksaan ini, program akan mencetak "Vokal" jika karakter yang dimasukkan adalah huruf vokal, "Konsonan" jika karakter tersebut adalah huruf konsonan, dan "Bukan Huruf" jika karakter yang dimasukkan bukan merupakan huruf.

### 3.) Contoh Soal 3

Source Code :

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3, d4 int
```

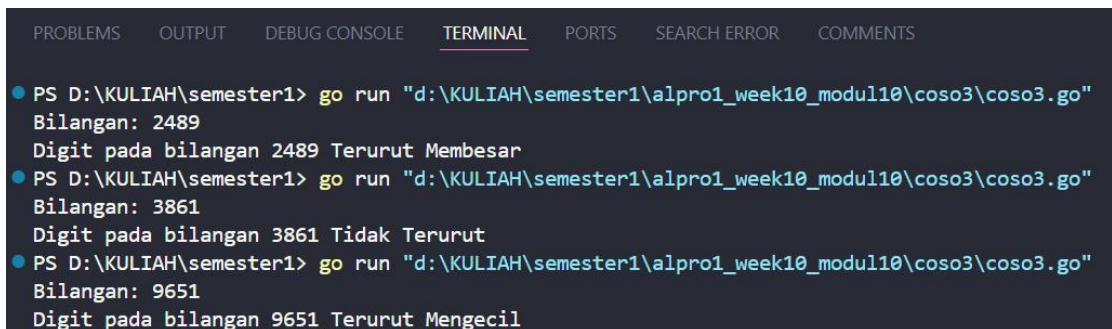
```

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)
}

```

Output :



```

PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso3\coso3.go"
Bilangan: 2489
Digit pada bilangan 2489 Terurut Membesar
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso3\coso3.go"
Bilangan: 3861
Digit pada bilangan 3861 Tidak Terurut
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\coso3\coso3.go"
Bilangan: 9651
Digit pada bilangan 9651 Terurut Mengecil

```

Deskripsi Program :

Program di atas adalah program untuk mengevaluasi urutan digit dari sebuah bilangan bulat empat digit yang dimasukkan oleh pengguna. Program dimulai dengan mendeklarasikan variabel untuk menyimpan bilangan dan setiap digitnya, serta sebuah variabel string untuk menyimpan hasil evaluasi. Setelah meminta pengguna untuk memasukkan bilangan, program kemudian memisahkan bilangan tersebut menjadi empat digit terpisah menggunakan operasi modulus dan pembagian. Selanjutnya, program melakukan pemeriksaan untuk menentukan apakah digit-digit tersebut terurut dalam urutan membesar, mengecil, atau tidak terurut sama sekali. Jika digit pertama lebih kecil

dari digit kedua, digit kedua lebih kecil dari digit ketiga, dan digit ketiga lebih kecil dari digit keempat, maka program akan mengeluarkan hasil "Terurut Membesar". Sebaliknya, jika digit pertama lebih besar dari digit kedua, digit kedua lebih besar dari digit ketiga, dan digit ketiga lebih besar dari digit keempat, maka hasil yang ditampilkan adalah "Terurut Mengecil". Jika kedua kondisi tersebut tidak terpenuhi, program akan mencetak "Tidak Terurut". Dengan demikian, program ini memberikan analisis yang jelas tentang urutan digit dalam bilangan yang diberikan.



## Latihan Soal

### 1.) Latihan Soal 1

Source Code :

```
package main

import "fmt"

func main() {

    var berat, beratKg, sisaGram, biayaKg, biayaSisaGram, totalBiaya
    int

    fmt.Print("Masukkan berat parsel (gram): ")
    fmt.Scan(&berat)

    beratKg = berat / 1000
    sisaGram = berat % 1000

    biayaKg = beratKg * 10000

    if sisaGram > 0 {
        if beratKg > 10 {
            biayaSisaGram = 0
        } else if sisaGram >= 500 {
            biayaSisaGram = sisaGram * 5
        } else {
            biayaSisaGram = sisaGram * 15
        }
    }

    totalBiaya = biayaKg + biayaSisaGram

    fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisaGram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg,
    biayaSisaGram)
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal1\latsoal1.go"
Masukkan berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal1\latsoal1.go"
Masukkan berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal1\latsoal1.go"
Masukkan berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
```

### Deskripsi Program :

Program di atas adalah program untuk menghitung biaya pengiriman parcel berdasarkan berat yang dimasukkan oleh pengguna dalam satuan gram. Program dimulai dengan mendeklarasikan variabel untuk menyimpan berat parcel, berat dalam kilogram, sisa gram, biaya per kilogram, biaya untuk sisa gram, dan total biaya bertipe integer. Kemudian meminta pengguna untuk memasukkan berat parcel dalam satuan gram. Setelah meminta pengguna untuk memasukkan berat parcel, program kemudian mengonversi berat tersebut dari gram ke kilogram dan menghitung sisa gram yang tidak terpakai. Biaya pengiriman dihitung berdasarkan berat dalam kilogram, di mana setiap kilogram dikenakan biaya Rp. 10.000. Untuk sisa gram, program menerapkan tarif yang berbeda: jika berat kilogram lebih dari 10, biaya untuk sisa gram adalah nol; jika sisa gram lebih dari atau sama dengan 500, dikenakan biaya Rp. 5 per gram; dan jika kurang dari 500 gram, dikenakan biaya Rp. 15 per gram. Setelah semua perhitungan dilakukan, program mencetak detail berat dalam kilogram dan gram, rincian biaya untuk setiap kategori, serta total biaya pengiriman.

## 2.) Latihan Soal 2

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 {
        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 if nam <= 40 {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

Output :

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 80
  Nilai mata kuliah: AB
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 80.1
  Nilai mata kuliah: A
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 74
  Nilai mata kuliah: AB
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 70
  Nilai mata kuliah: B
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 64
  Nilai mata kuliah: BC
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 45
  Nilai mata kuliah: D

```

Deskripsi Program :

Program di atas adalah program untuk menghitung dan menampilkan nilai huruf (NMK) berdasarkan input nilai akhir mata kuliah (NAM) yang diberikan oleh pengguna. Program dimulai dengan mendeklarasikan dua variabel, yaitu `nam` yang bertipe `float64` untuk menyimpan nilai akhir mata kuliah, dan `nmk` yang bertipe `string` untuk menyimpan hasil konversi nilai menjadi huruf. Setelah itu, program meminta pengguna untuk memasukkan nilai akhir mata kuliah melalui konsol. Berdasarkan nilai yang dimasukkan, program menggunakan serangkaian pernyataan `if-else` untuk menentukan nilai huruf yang sesuai dengan rentang nilai yang telah ditentukan: nilai di atas 80 akan mendapatkan "A", antara 72.5 hingga 80 mendapatkan "AB", dan seterusnya hingga nilai di bawah atau sama dengan 40 yang mendapatkan "E". Setelah menentukan nilai huruf, program mencetak hasilnya ke layar, memberikan umpan balik yang jelas kepada pengguna mengenai nilai mata kuliah yang telah dihitung. Program ini berguna untuk membantu mahasiswa atau pengajar dalam memahami sistem penilaian yang digunakan dalam mata kuliah.

Jawaban :

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

=> Jika `nam` diberikan adalah 80.1, maka program tersebut akan menghasilkan keluaran Nilai mata kuliah: tanpa nilai apapun, karena variabel `nmk` tidak pernah diubah selama eksekusi. Program ini tidak memenuhi spesifikasi soal karena terdapat beberapa kesalahan, baik dalam penggunaan variabel maupun dalam kondisi logika.

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

=> Kesalahan dalam program :

1. Salah Tipe Variabel :

Variabel `nam` diubah-ubah menjadi string ("A", "AB", dsb.) padahal seharusnya variabel `nmk` yang digunakan untuk menyimpan nilai huruf. Variabel `nam` seharusnya tetap tipe `float64` untuk menyimpan nilai angka asli.

2. Kondisi yang Salah :

Struktur kondisi if tidak tepat. Setiap if berdiri sendiri, sehingga setiap kondisi akan dievaluasi tanpa memperhatikan kondisi sebelumnya. Ini akan menghasilkan output yang tidak diharapkan, karena semua kondisi akan diuji secara berurutan, menyebabkan variabel nmk selalu mendapatkan nilai terakhir yang memenuhi syarat.

### 3. Tidak Ada Pengaturan Nilai nmk :

Variabel nmk tidak pernah diubah atau diisi, sehingga output `fmt.Println("Nilai mata kuliah: ", nmk)` tidak akan menampilkan nilai yang benar.

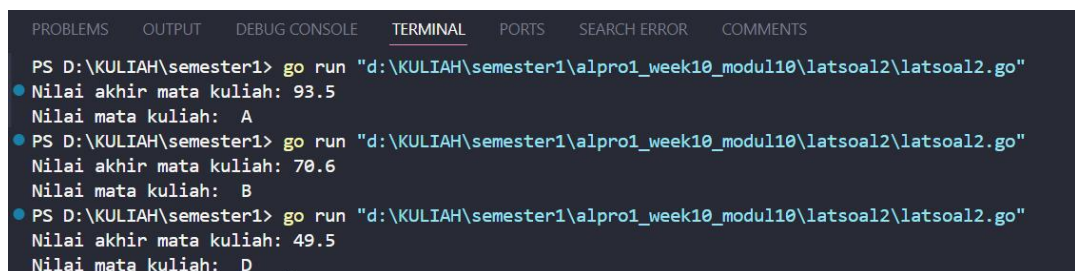
Alur Program Seharusnya :

- Program seharusnya mengecek nilai nam sekali saja, dimulai dari kondisi tertinggi ke terendah, menggunakan struktur `if...else if...else`.
- Variabel nmk diubah sesuai dengan kategori nilai nam.
- Nilai nam tetap di tipe `float64`, dan nmk digunakan untuk nilai huruf.

c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

=> Sudah diperbaiki di source code

Output :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS

PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 93.5
  Nilai mata kuliah: A
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 70.6
  Nilai mata kuliah: B
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal2\latsoal2.go"
● Nilai akhir mata kuliah: 49.5
  Nilai mata kuliah: D
```

Penjelasan Perbaikan :

- Struktur `if...else if...else` digunakan untuk memastikan setiap kondisi diperiksa satu kali saja, mulai dari nilai tertinggi hingga terendah.
- Variabel nam tetap bertipe `float64` untuk menyimpan nilai numerik asli.
- Variabel nmk digunakan untuk menyimpan nilai huruf, dan diubah sesuai dengan rentang nilai nam.

### 3.) Latihan Soal 3

Source Code :

```
package main

import "fmt"

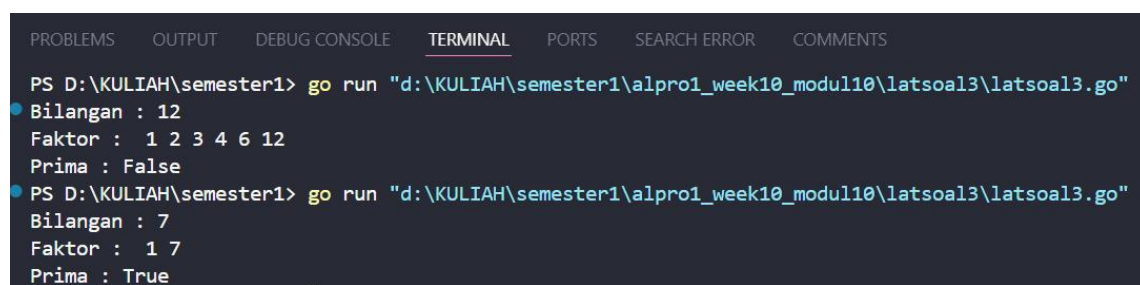
func main() {
    var b int
    fmt.Print("Bilangan : ")
    fmt.Scan(&b)

    fmt.Print("Faktor : ")
    jumlahFaktor := 0
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Print(" ", i)
            jumlahFaktor++
        }
    }

    fmt.Println()

    if jumlahFaktor == 2 {
        fmt.Println("Prima : True")
    } else {
        fmt.Println("Prima : False")
    }
}
```

Output :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS

PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal3\latsoal3.go"
Bilangan : 12
Faktor :  1 2 3 4 6 12
Prima : False
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week10_modul10\latsoal3\latsoal3.go"
Bilangan : 7
Faktor :  1 7
Prima : True
```

### Deskripsi Program :

Program di atas adalah program untuk menentukan faktor-faktor dari sebuah bilangan bulat yang dimasukkan oleh pengguna, serta mengecek apakah bilangan tersebut merupakan bilangan prima. Program dimulai dengan mendeklarasikan variabel `b` bertipe integer untuk menyimpan bilangan yang akan dianalisis. Setelah meminta pengguna untuk memasukkan bilangan, program kemudian mencetak semua faktor dari bilangan tersebut dengan menggunakan loop yang memeriksa setiap angka dari 1 hingga bilangan itu sendiri. Jika angka tersebut dapat membagi bilangan tanpa sisa, maka angka tersebut dicetak sebagai faktor dan jumlah faktor dihitung. Setelah semua faktor dicetak, program memeriksa jumlah faktor yang ditemukan. Jika jumlah faktor sama dengan 2, maka bilangan tersebut dinyatakan sebagai bilangan prima, dan program mencetak "Prima : True". Sebaliknya, jika jumlah faktor lebih dari 2, program mencetak "Prima : False".

## **DAFTAR PUSTAKA**

MODUL PRAKTIKUM 10 - ELSE-IF

ALGORITMA DAN PEMROGRAMAN 1

A.13. Seleksi Kondisi. (n.d.). Retrieved from

<https://dasarpemrogramangolang.novalagung.com/A-seleksi-kondisi.html>