

**LAPORAN PRAKTIKUM**  
**MODUL: 10**  
**ALGORITMA DAN PEMROGRAMAN**



**DISUSUN OLEH:**

**NAMA: JESIKA METANIA RAHMA ARIFIN**

**NIM: 103112400080**

**KELAS: 12 IF 01**

**DOSEN:**

**Yohani Setiya Rafika Nur, M.Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024/2025**

## DASAR TEORI

### ELSE-IF

1. 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.
2. 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. Sebagai contoh ilustrasi penulisannya adalah sebagai berikut:

Notasi dalam pseudocode      Notasi dalam bahasa Go

Sebagai contoh ilustrasi penulisannya adalah sebagai berikut:

Notasi dalam pseudocode	Notasi dalam bahasa Go
<b>if</b> kondisi <b>then</b> // aksi <b>else</b> // aksi lain <b>endifor</b>	<b>if</b> kondisi { // aksi } <b>else</b> { // aksi lain }

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_1 then     // aksi 1 else     if kondisi_2 then         // aksi 2     else         if kondisi_3 then             // aksi 3         else             // aksi lain         endfor     endfor endfor </pre>	<pre> if kondisi_1 {     // aksi 1 }else{     if kondisi_2 {         // 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>

### GUIDED:1

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

```

package main

import "fmt"

func main() {
    var umur int
    var KK bool
    fmt.Scan(&umur, &KK)
    if umur >= 17 && KK {
        fmt.Println("bisa membuat KTP")
    } else {
        fmt.Println("belum bisa membuat KTP")
    }
}

```

#### OUTPUT:



```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\MODUL 10> go run "c:\MODUL 10\coso1.go"
17
TRUE
bisa membuat KTP
PS C:\MODUL 10> go run "c:\MODUL 10\coso1.go"
20
FALSE
belum bisa membuat KTP
PS C:\MODUL 10> go run "c:\MODUL 10\coso1.go"
15
TRUE
belum bisa membuat KTP
PS C:\MODUL 10>

```

#### DESKRPSI PROGRAM:

*Program di atas adalah program sederhana menggunakan bahasa Go (Golang) untuk menentukan apakah seorang penduduk bisa membuat KTP atau tidak menggunakan Else-if.*

#### GUIDED: 2

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

```
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 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\MODUL 10> go run "c:\MODUL 10\coso2.go"
A
vokal
PS C:\MODUL 10> go run "c:\MODUL 10\coso2.go"
f
konsonan
PS C:\MODUL 10> go run "c:\MODUL 10\coso2.go"
1
bukan huruf
PS C:\MODUL 10> go run "c:\MODUL 10\coso2.go"
$
bukan huruf
PS C:\MODUL 10> 
```

DESKRIPSI PROGRAM:

*Program ini adalah program yang dibuat untuk menentukan apakah sebuah karakter alfabet yang diberikan adalah huruf vokal, huruf konsonan, atau bukan huruf. Program ini dirancang sederhana dengan memeriksa input berupa satu huruf dan memberikan keluaran sesuai dengan klasifikasi huruf tersebut.*

### GUIDED : 3

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

```
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:

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\MODUL 10> go run "c:\MODUL 10\coso3.go"
bilangan: 2489
digit pada bilangan 2489 terurut membesar
PS C:\MODUL 10> go run "c:\MODUL 10\coso3.go"
bilangan: 3861
digit pada bilangan 3861 tidak terurut
PS C:\MODUL 10> go run "c:\MODUL 10\coso3.go"
bilangan: 9651
digit pada bilangan 9651 terurut mengecil
PS C:\MODUL 10> 
```

#### DESKRIPSI PROGRAM:

Program ini adalah program untuk menentukan apakah digit-digit dalam sebuah bilangan empat digit (1000–9999) memiliki pola **terurut membesar**, **terurut mengecil**, atau **tidak terurut**. Program menerima input bilangan dari pengguna, lalu mengevaluasi pola dari digit-digit tersebut berdasarkan urutan angka.

#### UNGUIDED: 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 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.

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

```

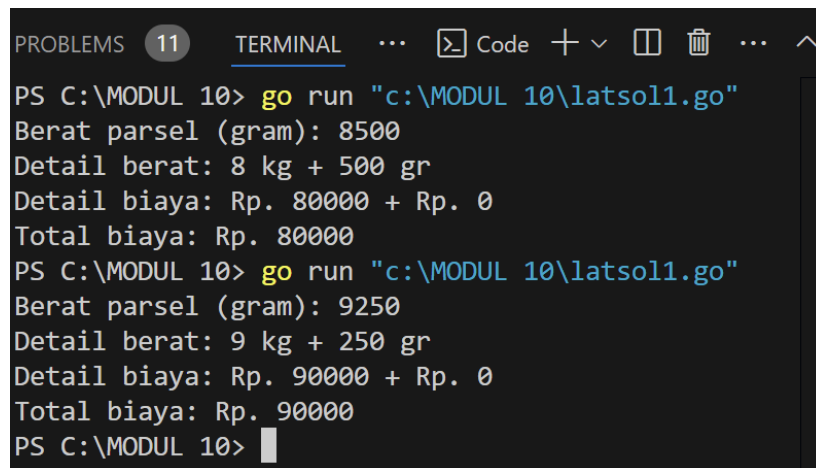
package main

import (
    "fmt"
)

func main() {
    var parsel, ril_berat, berat, biaya, sisa int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scanln(&ril_berat)
    parsel = ril_berat / 1000
    berat = ril_berat % 1000
    fmt.Printf("Detail berat: %d kg + %d gr\n", parsel, berat)
    biaya = parsel * 10000
    if ril_berat > 10 {
        sisa = 0
    } else if berat < 500 {
        sisa = berat * 15
    } else if berat >= 500 {
        sisa = berat * 5
    }
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biaya, sisa)
    fmt.Printf("Total biaya: Rp. %d", biaya+sisa)
}

```

#### OUTPUT:



```

PROBLEMS 11 TERMINAL ... Code + - [ ] [ ] ... ^
PS C:\MODUL 10> go run "c:\MODUL 10\latsol1.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 0
Total biaya: Rp. 80000
PS C:\MODUL 10> go run "c:\MODUL 10\latsol1.go"
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 0
Total biaya: Rp. 90000
PS C:\MODUL 10> 

```

#### DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk membantu PT POS menghitung biaya pengiriman berdasarkan berat parsel yang diberikan oleh pengguna. Program ini akan



menerima input berat parcel dalam gram dan menghitung total biaya pengiriman sesuai dengan ketentuan tarif yang telah ditentukan.

## UNGUIDED: 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

```
package main
```

```
import (  
    "fmt"  
)
```

```
func main() {  
    var parcel, ril_berat, berat, biaya, sisa int  
    fmt.Println("Berat parcel (gram): ")  
    fmt.Scanln(&ril_berat)  
    parcel = ril_berat / 1000  
    berat = ril_berat % 1000  
    fmt.Printf("Detail berat: %d kg + %d gr\n", parcel, berat)  
    biaya = parcel * 10000  
    if ril_berat > 10 {  
        sisa = 0  
    } else if berat < 500 {  
        sisa = berat * 15  
    } else if berat >= 500 {  
        sisa = berat * 5  
    }  
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biaya, sisa)  
    fmt.Printf("Total biaya: Rp. %d", biaya+sisa)  
}
```

Jawablah pertanyaan-pertanyaan berikut:

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

Jawaban: Keluaran dari program tersebut tetap nilai nam yang diberikan, karena program tidak dapat mengeksekusi. Eksekusi program tidak sesuai spesifikasi karena nilai nam nya berupa "float64" seharusnya "string"

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

Jawaban: kesalahan program tersebut diantaranya adalah ketidakkonsistenan tipe data, urutan kondisi if yang tidak benar, dan penggunaan else if yang tidak benar. Urutan alur program yang benar seharusnya:

1. Mampu membaca nilai nam
  2. Menggunakan struktur kondisi yang memeriksa nilai tersebut dari yang tertinggi ke terendah
  3. Menampilkan hasil dalam bentuk huruf (A, B, C, dst.) berdasarkan rentang nilai yang sesuai.
- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'

```
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 10 TERMINAL ... Code + - [ ] [X] ... ^
PS C:\MODUL 10> go run "c:\MODUL 10\latsol2.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\MODUL 10> go run "c:\MODUL 10\latsol2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\MODUL 10> go run "c:\MODUL 10\latsol2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\MODUL 10> 
```

### UNGUIDED: 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.

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.

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

```
package main

import "fmt"

func main() {
    var x int
    fmt.Print("bilbul: ")
    fmt.Scanln(&x)
    fmt.Printf("Faktor: ")
    for i := 1; i <= x; i++ {
        if x%i == 0 {
            fmt.Print(i, " ")
        }
    }
}
```

```

    }

    fmt.Print("\n")

    if x%2 == 0 || x%3 == 0 || x%5 == 0 || x%7 == 0 && x != 1 && x != 2 && x != 3
    && x != 5 && x != 7 {
        fmt.Println("FALSE")
    } else {
        fmt.Println("TRUE")
    }
}

```

OUTPUT:

```

PROBLEMS 10 TERMINAL ... Code + v [] [X] ... ^
PS C:\MODUL 10> go run "c:\MODUL 10\latso13.go"
bilbul: 12
Faktor: 1 2 3 4 6 12
FALSE
PS C:\MODUL 10> go run "c:\MODUL 10\latso13.go"
bilbul: 7
Faktor: 1 7
TRUE
PS C:\MODUL 10>

```

DESKRIPSI PROGRAM:

Program ini dibuat untuk menerima input sebuah bilangan bulat  $b$  dengan kondisi  $b > 1$ , kemudian mencari dan menampilkan semua faktor dari bilangan tersebut. Setelah itu, program akan memeriksa apakah bilangan tersebut adalah bilangan prima, yaitu bilangan yang hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri.