

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**MODUL 10**

**MODUL 10. ELSE-IF**



**Disusun oleh:**

**ALMA BONITA MIA WARDHANA**

**109082500015**

**S1IF-13-04**

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## LATIHAN KELAS – GUIDED

### 1. Guided 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")
    }
}
```

#### Screenshot program

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a project structure with a folder named "MODUL10ALMA" containing a "guided1" folder and a file named "contohsoal1.go".
- Code Editor:** Displays the source code of "contohsoal1.go". The code defines a package "main" with a function "main" that reads an integer "usia" and a boolean "kk" from standard input using `fmt.Scan`. It then prints "bisa membuat KTP" if both conditions are met, or "belum bisa membuat KTP" otherwise.
- Terminal:** Shows the command-line interface (PS) running the program. The output indicates that for an input of 17 and kk set to true, it prints "bisa membuat KTP".
- Output Window:** Shows the printed output: "Nama : Alma Bonita Mia Wardhana", "NIM : 109082500015", and "Kelas : S1IF-13-04".
- System Tray:** Shows various system icons including battery level, signal strength, and date/time.

## **Deskripsi program**

Program ini meminta dua input dari pengguna, yaitu nilai usia dalam bentuk bilangan bulat dan nilai boolean yang menunjukkan apakah pengguna memiliki KK atau tidak. Setelah menerima input tersebut, program melakukan pengecekan kondisi menggunakan operator logika. Jika usia yang dimasukkan bernilai 17 atau lebih dan nilai boolean KK adalah true, maka program menampilkan output “bisa membuat KTP”.

Namun, jika salah satu atau kedua syarat tersebut tidak terpenuhi, program akan menampilkan “belum bisa membuat KTP”. Program ini sederhana dan bertujuan melatih penggunaan input bertipe berbeda serta penggunaan logika AND dalam percabangan.

## **2. Guided 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")
```

```
}
```

```
}
```

## Screenshot program

```
package main
import "fmt"
func main() {
    var x rune
    var huruf, vKecil, vBesar bool
    fmt.Scan("%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")
    }
}
```

The screenshot shows a Go development environment with two files open: `contohsoal1.go` and `contohsoal2.go`. The code in `contohsoal2.go` is a simple program that reads a character from the user and prints whether it is a vowel ('vokal'), a consonant ('konsonan'), or not a letter ('bukan huruf'). The terminal window shows the execution of the program and its output for various characters.

## Deskripsi program

Program di atas digunakan untuk mengecek apakah satu karakter yang diinputkan pengguna merupakan huruf vokal, huruf konsonan, atau bukan huruf. Pertama, program membaca satu karakter ke variabel `x` bertipe `rune`. Lalu program menentukan apakah karakter tersebut termasuk huruf dengan mengecek apakah berada dalam rentang 'a' sampai 'z' atau 'A' sampai 'Z'. Setelah itu, program membandingkan `x` dengan daftar huruf vokal, baik versi kecil maupun besar.

Jika karakter termasuk huruf dan masuk kategori vokal, program mencetak "vokal". Jika karakter termasuk huruf tapi bukan vokal, maka dicetak "konsonan". Namun jika karakter yang dimasukkan bukan huruf sama sekali, program menampilkan "bukan huruf". Program ini sederhana dan langsung memeriksa karakter berdasarkan kode ASCII tanpa library tambahan.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var bilangan int
```

```
var d1, d2, d3, d4 int

var teks string

fmt.Print("Bilangan: ")

fmt.Scan(&bilangan)

d1 = bilangan / 1000

d2 = (bilangan / 100) % 10

d3 = (bilangan / 10) % 10

d4 = bilangan % 10

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)
```

### Screenshoot program

The screenshot shows the GoLand IDE interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The left sidebar has sections for EXPLORER, OUTLINE, TIMELINE, GO, and PACKAGE OUTLINE, with 'MODUL10ALMA' selected. The main editor window displays a Go file named 'contohsoal3.go' with the following code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7     var d1, d2, d3, d4 int
8     var teks string
9
10    fmt.Println("Bilangan: ")
11    fmt.Scan(&bilangan)
12
13    d1 = bilangan / 1000
14    d2 = (bilangan / 100) % 10
15    d3 = (bilangan / 10) % 10
16    d4 = bilangan % 10
17
18    if d1 < d2 && d2 < d3 && d3 < d4 {
19        teks = "terurut membesar"
20    } else if d1 > d2 && d2 > d3 && d3 > d4 {
21        teks = "terurut mengelciil"
22    } else {
23        teks = "tidak terurut"
24    }
25
26    fmt.Println("Digit pada bilangan", bilangan, teks)
27 }
```

Below the editor, tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS are visible. The TERMINAL tab shows command-line history:

```
PS D:\Modul10Alma> go run "d:\Modul10Alma\guided3\contohsoal3.go"
Bilangan: 2489
PS D:\Modul10Alma> bilangan 2489 terurut membesar
PS D:\Modul10Alma> go run "d:\Modul10Alma\guided3\contohsoal3.go"
Bilangan: 3861
PS D:\Modul10Alma> digit pada bilangan 3861 tidak terurut
PS D:\Modul10Alma> go run "d:\Modul10Alma\guided3\contohsoal3.go"
PS D:\Modul10Alma> digit pada bilangan 9651 terurut mengelciil
PS D:\Modul10Alma> []
```

A floating terminal window displays the student's information:

```
Nama : Alma Bonita Mia Wardhani
NIM : 109982500015
Kelas : S1IF-13-04
```

The bottom status bar shows file information (Ln 1, Col 20), character count (69), and various system icons.

## Deskripsi program

Program di atas berfungsi untuk memeriksa apakah empat digit pada sebuah bilangan tersusun secara membesar, mengecil, atau tidak berurutan. Setelah pengguna memasukkan sebuah bilangan empat digit, program memisahkan setiap digit menggunakan operasi pembagian dan modulo: d1 adalah ribuan, d2 ratusan, d3 puluhan, dan d4 satuan. Kemudian program membandingkan keempat digit tersebut. Jika digit pertama lebih kecil dari digit kedua, lalu lebih kecil dari digit ketiga, dan digit ketiga lebih kecil dari digit keempat, program menyimpulkan bahwa digit-digit itu terurut membesar.

Sebaliknya, jika setiap digit justru lebih besar dari digit setelahnya, maka hasilnya terurut mengecil. Jika tidak memenuhi kedua kondisi tersebut, berarti digit-digitnya tidak terurut. Terakhir, program mencetak hasil analisis berdasarkan bilangan yang dimasukkan.

# TUGAS

## 1. Tugas 1

## Source code

```
package main

import "fmt"

func main() {
    var gram int

    fmt.Println("Berat parsel (gram): ")
```

```
fmt.Scan(&gram)

kg := gram / 1000

sisa := gram % 1000

biayaKg := kg * 10000

var biayaSisa int

if kg > 10 {

    biayaSisa = 0

} else {

    if sisa >= 500 {

        biayaSisa = sisa * 5

    } else {

        biayaSisa = sisa * 15

    }

}

total := biayaKg + biayaSisa

fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisa)

fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg,
biayaSisa)

fmt.Printf("Total biaya: Rp. %d\n", total)

}
```

## Screenshot program

```

package main

import "fmt"

func main() {
    var gram int
    fmt.Println("Berat parsel (gram): ")
    fmt.Scan(&gram)

    kg := gram / 1000
    sisa := gram % 1000

    biayaKG := kg * 10000
    var biayaSisa int
    if kg > 10 {
        biayaSisa = 0
    } else {
        if sisa >= 500 {
            biayaSisa = sisa * 5
        } else {
            biayaSisa = sisa * 15
        }
    }

    total := biayaKG + biayaSisa

    fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisa)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKG, biayaSisa)
    fmt.Printf("Total biaya: Rp. %d\n", total)
}

```

### Deskripsi program

Program di atas digunakan untuk menghitung biaya pengiriman parsel berdasarkan berat yang dimasukkan dalam satuan gram. Pertama, program memecah berat tersebut menjadi satuan kilogram dan sisa gram dengan operasi pembagian dan modulo. Biaya utama dihitung berdasarkan jumlah kilogram, yaitu 10.000 rupiah per kilogram.

Untuk sisa gram, biaya dihitung berbeda: jika total berat lebih dari 10 kilogram maka sisa gram digratiskan, tetapi jika tidak, sisa 500 gram ke atas dikenai tarif 5 rupiah per gram, sedangkan sisa di bawah 500 gram dikenai tarif 15 rupiah per gram. Setelah itu, program menjumlahkan biaya kilogram dan biaya sisa untuk mendapatkan total biaya pengiriman. Bagian akhir program menampilkan rincian berat, rincian biaya, serta total biaya dengan format yang jelas sehingga mudah dipahami pengguna.

## 2. Tugas 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 {
    nmk = "E"
}

fmt.Println("Nilai mata kuliah:", nmk)
}
```

### Screenshoot program

## Deskripsi program

Program di atas digunakan untuk menentukan nilai huruf berdasarkan nilai akhir mata kuliah yang dimasukkan oleh pengguna. Program membaca nilai angka bertipe float, kemudian memeriksa rentang nilainya menggunakan struktur *if-else if* secara berurutan. Jika nilai lebih dari 80, mahasiswa mendapat nilai A; jika tidak, program melanjutkan pengecekan untuk nilai AB, B, BC, C, D, hingga akhirnya memberi nilai E bila nilainya berada pada rentang paling rendah.

Logika bertingkat ini memastikan hanya satu nilai huruf yang dipilih sesuai kondisi yang pertama kali terpenuhi. Di akhir, program menampilkan nilai huruf yang diperoleh sehingga pengguna dapat melihat hasil konversi nilai angkanya.

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

Jawab : D

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

Jawab :

1. Tipe data salah  
nam dideklarasikan sebagai float64, tapi diubah menjadi string ("A").  
Ini menyebabkan error program.

2. Menggunakan banyak if terpisah, bukan else if

2. Menggunakan banyak if terpisah, bukan else if  
Karena semua kondisi diperiksa satu per satu, nilai tinggi seperti 80.1 akan memenuhi semua kondisi dan akhirnya mendapat nilai terakhir, yaitu D.

3. Variabel output nilai huruf (nmk) tidak pernah diisi

Sehingga output akan kosong.

4. Kutip tanda petik salah (" " bukan " ")

Ini menyebabkan program tidak bisa dikompilasi.

Alur program seharusnya = Terima nilai angka,Tentukan nilai huruf sesuai range,Gunakan if–else if agar hanya satu kondisi dijalankan,Simpan hasil pada variabel string, Cetak hasil.

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

### 3. Tugas 3

#### Source code

a.

```
package main

import "fmt"

func main() {
    var b int

    fmt.Print("Bilangan: ")
    fmt.Scan(&b)

    fmt.Print("Faktor: ")

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

b.

```
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()

    prima := false
    if jumlahFaktor == 2 {
        prima = true
    }

    fmt.Println("Prima:", prima)
}
```

## Screenshot program

The screenshot shows a Go development environment with two terminal windows and two output windows.

**Terminal 1:**

```
PS D:\Modul10Alma> go run "d:\Modul10Alma\latsol3\latsol3.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
```

**Terminal 2:**

```
PS D:\Modul10Alma> go run "d:\Modul10Alma\latsol3\latsol3.go"
Bilangan: 7
Faktor: 1 7
```

**Output Window 1 (Top):**

```
Name : Alma Bonita Mia Wardhana
NIM : 109082500015
Kelas : S1IF-13-04
```

**Output Window 2 (Bottom):**

```
Name : Alma Bonita Mia Wardhana
NIM : 109082500015
Kelas : S1IF-13-04
```

## Deskripsi program

Program (a) digunakan untuk menampilkan seluruh faktor dari sebuah bilangan yang dimasukkan oleh pengguna. Setelah pengguna memasukkan nilai b, program melakukan perulangan dari 1 hingga b dan setiap angka dicek apakah dapat membagi b tanpa sisa. Jika iya, angka tersebut dicetak sebagai faktor. Program ini hanya menampilkan faktor-faktor bilangan tanpa memberikan analisis tambahan.

### **Penjelasan Program (b)**

Program (b) memiliki fungsi yang sama seperti program pertama, yaitu menampilkan faktor-faktor dari bilangan b menggunakan perulangan dan pemeriksaan  $b \% i == 0$ . Namun, program ini juga menghitung berapa banyak faktor yang dimiliki bilangan tersebut dengan menambah nilai jumlahFaktor setiap kali ditemukan faktor. Setelah semua faktor ditampilkan, program menentukan apakah bilangan tersebut prima dengan memeriksa apakah jumlah faktornya tepat dua.

Jika jumlah faktor = 2, maka bilangan adalah prima; jika lebih dari dua, maka bukan prima. Program ini menampilkan daftar faktor sekaligus hasil pengecekan apakah bilangan tersebut prima atau tidak.