

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 10**

**ELSE-IF**



**Disusun oleh:**

**MUHAMMAD FIRDAUS ARDIANSYAH**

**109082500126**

**S1IF-13-07**

**Asisten Praktikum**

Adithana dharma putra

Apri pandu wicaksono

**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 Go development environment with the following components:

- File Explorer:** Shows a tree view of files and folders under "LATIHAN GO 1".
- Code Editor:** Displays the main.go file from the guided1 module. The code checks if the user's age (usia) is at least 17 and if they are a citizen (kk). If both conditions are met, it prints "bisa membuat KTP"; otherwise, it prints "belum bisa membuat KTP".
- Terminal:** Shows the command "go run d:\Coding\Latihan GO 1\modul10\guided1\main.go" being run twice. The first run with age 17 and citizen status true results in "bisa membuat KTP". The second run with age 20 and citizen status false results in "belum bisa membuat KTP".
- Output:** A message box displays the student's information: MUHAMMAD FIRDAUS ARDIANSYAH, S11F-13-07, and ID 109082500126.

## Deskripsi program

Program ini berfungsi untuk memvalidasi kelayakan seseorang dalam membuat Kartu Tanda Penduduk (KTP). Di dalam fungsi main, dideklarasikan dua variabel input: usia dengan tipe data *integer* (bilangan bulat) dan kk dengan tipe data *boolean* (benar/salah). Program menggunakan fungsi fmt.Scan untuk menerima input langsung dari pengguna melalui terminal, yang kemudian disimpan ke dalam alamat memori kedua variabel tersebut.

Logika penentu keputusan menggunakan struktur percabangan if-else dengan operator logika **AND** (`&&`). Agar program menampilkan pesan "bisa membuat KTP", dua kondisi harus terpenuhi secara bersamaan: nilai variabel usia harus minimal 17 ( $\geq 17$ ) **dan** nilai variabel kk harus bernilai true. Jika salah satu atau kedua kondisi tersebut tidak terpenuhi (seperti pada contoh kedua di terminal di mana kk bernilai false), program akan mengeksekusi blok else dan mencetak output "belum bisa membuat KTP".

## 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

```

File Edit Selection View Go Run Terminal Help ← →
... guided1 U > main.go modul10\guided2 U > main.go modul10\guided3 U > code.png U > main.go modul10\soal2 U > main.go modul10\soal3 U > ...
EXPLORER
LATIHAN GO 1
BERLATIH
Coba
latihan
latihan9
latihan
latihan2
latihan3
latihan10
main.go
main.go
main.go
main.go
Modul1
Modul2
modul3
Modul4
modul5
modul9
modul10
guided1
main.go
guided2
main.go
guided3
soal1
soal2
main.go
soal3
TIMELINE
OUTLINE
GO
PACKAGE OUTLINE
modul10 > guided2 > main.go > main
3 import "fmt"
4
5 func main() {
6     var x rune
7     var huruf, vKecil, vBesar bool
8     fmt.Scanf("%c", &x)
9     huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
10    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
11    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
12    if huruf && (vKecil || vBesar) {
13        fmt.Println("vokal")
14    } else if huruf && !(vKecil || vBesar){
15        fmt.Println("konsonan")
16    } else{
17        fmt.Println("bukan huruf")
18    }
19 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul10\guided2\main.go"
A
vokal
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul10\guided2\main.go"
f
konsonan
PS D:\Coding\Latihan GO 1> []

```

### Deskripsi program

Program ini berfungsi untuk mengklasifikasikan karakter input menjadi huruf vokal, konsonan, atau bukan huruf. Program mendeklarasikan variabel *x* dengan tipe data *rune* untuk menampung karakter tunggal, serta beberapa variabel *boolean* (*huruf*, *vKecil*, *vBesar*) sebagai penanda (flag). Setelah menerima input melalui *fmt.Scanf*, program mengevaluasi apakah karakter tersebut termasuk dalam rentang abjad (a-z atau A-Z) dan apakah karakter tersebut merupakan huruf vokal spesifik, baik huruf kecil maupun kapital.

Logika penentu output menggunakan struktur percabangan *if-else if-else* yang menggabungkan hasil evaluasi boolean tersebut. Jika karakter teridentifikasi sebagai huruf dan merupakan vokal (salah satu dari *vKecil* atau *vBesar*), program mencetak "vokal". Sebaliknya, jika karakter adalah huruf tetapi bukan vokal (menggunakan operator negasi *!*), program mencetak "konsonan". Jika karakter input tidak memenuhi kriteria sebagai huruf (misalnya angka atau simbol), program akan menjalankan blok terakhir dan mencetak "bukan huruf".

### 3. Guided 3

#### Source Code

```
package main
```

```
import "fmt"
```

```
func main() {
```

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

var teks string

fmt.Println("Bilangan: ")

fmt.Scan(&bilangan)

d4 = bilangan % 10

d3 = (bilangan % 100) / 10

d2 = (bilangan % 1000) / 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)
```

### Screenshot program

The screenshot shows a Go development environment with the following components:

- EXPLORER**: Shows a tree view of project files under "LATHAN GO 1". Files include main.go, guided1.go, guided2.go, guided3.go, and several "soal" files.
- CODE EDITOR**: Displays the content of main.go. The code reads a four-digit integer, extracts digits, and prints their arrangement (terurut besar, terurut mengecil, or tidak terurut).
- TERMINAL**: Shows command-line output for running the program with inputs 2489 and 3861, and for listing files in the module.
- OUTPUT**: Shows no errors or warnings.
- DEBUG CONSOLE**: Not visible in the screenshot.
- PROBLEMS**: Not visible in the screenshot.
- PORTS**: Not visible in the screenshot.

## Deskripsi program

Program ini digunakan untuk menganalisis pola urutan digit pada sebuah bilangan bulat empat digit (maksimal 9999). Program menerima satu input *integer*, lalu memecah bilangan tersebut menjadi digit-digit penyusunnya (ribuan, ratusan, puluhan, dan satuan) menggunakan operasi aritmatika. Operasi modulus (%) digunakan untuk mengambil sisa bagi (mendapatkan digit terakhir), sementara pembagian integer (/) digunakan untuk membuang digit belakang, sehingga nilai d1 hingga d4 dapat merepresentasikan setiap posisi digit dari bilangan tersebut.

Setelah digit-digit berhasil dipisahkan, program melakukan evaluasi logika menggunakan struktur percabangan if-else. Program memeriksa apakah digit-digit tersebut memiliki urutan nilai yang naik secara konsisten ( $d1 < d2 < d3 < d4$ ) untuk menetapkan status "terurut membesar", atau turun secara konsisten ( $d1 > d2 > d3 > d4$ ) untuk status "terurut mengecil". Jika digit-digit tersebut acak atau memiliki angka yang sama sehingga tidak memenuhi kedua pola tersebut, program menetapkan status "tidak terurut" dan menampilkan hasilnya ke layar.

## TUGAS

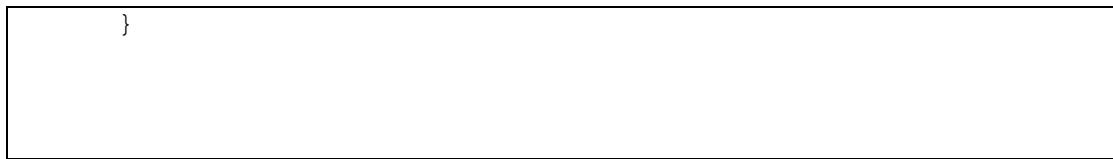
### 1. Tugas 1

#### Source code

```
package main

import "fmt"
```

```
func main() {  
  
    var beratParsel, kg, gram, biayaKg, biayaSisa,  
    totalBiaya int  
  
    fmt.Println("Berat parsel (gram): ")  
    fmt.Scan(&beratParsel)  
  
    kg = beratParsel / 1000  
    gram = beratParsel % 1000  
  
    biayaKg = kg * 10000  
  
    if kg >= 10 {  
        biayaSisa = 0  
    } else if gram >= 500 {  
        biayaSisa = gram * 5  
    } else {  
        biayaSisa = gram * 15  
    }  
  
    totalBiaya = biayaKg + biayaSisa  
  
    fmt.Println("Detail berat:", kg, "kg +", gram,  
    "gr")  
    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.",  
    biayaSisa)  
    fmt.Println("Total biaya: Rp.", totalBiaya)
```



## Screenshot program

```

package main

import "fmt"

func main() {
    var beratParsel, kg, gram, biayaKg, biayaSisa, totalBiaya int

    fmt.Println("Berat parsel (gram): ")
    fmt.Scan(&beratParsel)

    kg = beratParsel / 1000
    gram = beratParsel % 1000

    biayaKg = kg * 10000

    if kg >= 10 {
        biayaSisa = 0
    } else if gram >= 500 {
        biayaSisa = gram * 5
    } else {
        biayaSisa = gram * 15
    }

    totalBiaya = biayaKg + biayaSisa

    fmt.Println("Detail berat:", kg, "kg +", gram, "gr")
    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.", biayaSisa)
    fmt.Println("Total biaya: Rp.", totalBiaya)
}

```

Output from the terminal:

```

PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul10\soal1\main.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\Coding\Latihan GO 1>

```

## Deskripsi program

Program ini menerima input berat total parsel dalam gram, kemudian memecahnya menjadi komponen kilogram dan sisa gram menggunakan operasi pembagian integer dan modulus (%). Biaya dasar pengiriman tetap dihitung berdasarkan berat kilogram mutlak dikalikan dengan tarif Rp. 10.000.

Perubahan signifikan terdapat pada logika penentuan biaya tambahan (biayaSisa) menggunakan struktur if-else. Program kini memiliki aturan prioritas: jika berat paket mencapai 10 kg atau lebih, biaya sisa gram digratiskan (di-set menjadi 0). Jika beratnya kurang dari 10 kg, barulah aturan berjenjang diterapkan: sisa gram  $\geq 500$  dikenakan tarif Rp. 5 per gram, sedangkan sisa di bawah 500 gram dikenakan tarif lebih mahal, yaitu Rp. 15 per gram.

## 2. Tugas 2

### Jawaban Program

#### a.Jika nam = 80.1, apa keluarannya? Apakah sesuai spesifikasi soal?

Tidak Sesuai. Karena program seharusnya menghasilkan nilai huruf, tetapi:

- Program ini salah tipe data (mengisi string ke variabel float).
- Program ini tidak memakai else if.
- Nilai huruf ditulis ke variabel nam, bukan nmk.
- Variabel nmk selalu kosong → output tidak benar.
- Logika kondisi tumpang tindih → hasil tidak pernah benar

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

**1. Kesalahan tipe data**

nam bertipe float64, tetapi diisi string seperti "A", "AB" → **error** dan program tidak bisa dikompilasi.

**2. Variabel yang salah digunakan**

- Variabel yang benar untuk nilai huruf adalah nmk, tetapi program menulis huruf ke nam.
- Karena itu nmk tetap kosong → output tidak ada nilai huruf.

**3. Semua kondisi menggunakan if terpisah**

**4. Rentang kondisi tumpang tindih**

nam > 80, nam > 72.5, nam > 65 semua overlap.

**5. Struktur if–else if–else yang salah**

**6. Output tidak sesuai format soal**

Soal ingin huruf nilai tampil, tetapi variabel yang dicetak (nmk) kosong.

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

**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)
}

```

## Screenshoot program

The screenshot shows a Go development environment in Visual Studio Code. The code editor displays a main.go file with the following content:

```

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)
}

```

The terminal pane shows the following command-line interaction:

```

PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul10\soal2\main.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul10\soal2\main.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS D:\Coding\Latihan GO 1>

```

## Deskripsi program

Fungsi program ini adalah menerjemahkan nilai angka (Nilai Akhir Mata Kuliah) menjadi nilai huruf atau indeks prestasi. Program menggunakan variabel nam dengan

tipe data float64 untuk mengakomodasi input angka desimal yang presisi, serta variabel nmk bertipe string untuk menyimpan hasil konversi hurufnya. Input diambil langsung dari pengguna melalui fungsi fmt.Scan.

Logika konversi diimplementasikan menggunakan struktur percabangan bertingkat if-else if. Program memeriksa nilai input secara berurutan mulai dari batas tertinggi (di atas 80) hingga batas terendah. Jika sebuah kondisi terpenuhi (misalnya nilai > 72.5), program langsung menetapkan huruf mutu yang sesuai (dalam hal ini "AB") dan mengabaikan kondisi di bawahnya. Proses ini berlanjut hingga kondisi terakhir, di mana nilai yang kurang dari atau sama dengan 40 (<= 40) akan dikategorikan sebagai grade "E".

#### d. Tugas 3A

##### Source code

```
package main

import "fmt"

func main() {
    var bilangan int

    fmt.Print("bilangan : ")
    fmt.Scan(&bilangan)

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

##### Screenshoot program

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7
8     fmt.Println("bilangan : ")
9     fmt.Scan(&bilangan)
10
11    fmt.Println("Faktor : ")
12    for i := 1; i <= bilangan; i++ {
13        if bilangan % i == 0 {
14            fmt.Println(i, " ")
15        }
16    }
17}

```

The screenshot shows the VS Code interface with the code editor, terminal, and status bar. The terminal output shows the program running and printing factors of 12 and 7 respectively.

## Deskripsi program

Program ini digunakan untuk mencari dan menampilkan semua faktor dari sebuah bilangan bulat positif. Program mendeklarasikan variabel bilangan dengan tipe data *integer*, lalu meminta kita memasukkan sebuah angka melalui fungsi `fmt.Scan`. Angka yang diinputkan ini akan menjadi batas atas untuk proses pencarian faktor.

Logika inti program menggunakan struktur perulangan `for` yang melakukan iterasi mulai dari angka 1 hingga mencapai nilai bilangan itu sendiri. Di dalam setiap putaran loop, terdapat kondisi `if` yang menggunakan operator modulus (%) untuk memeriksa apakah bilangan habis dibagi oleh angka iterator saat ini (`i`). Jika hasil bagi tersebut tidak memiliki sisa (sama dengan 0), maka `i` dianggap sebagai faktor dari bilangan tersebut dan langsung dicetak ke layar.

## Tugas 3B

### Source code

```

package main

import "fmt"

func main() {
    var bilangan int
    var prima bool

```

```
var jumlahFaktor int = 0

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

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

fmt.Println()

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

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

### Screenshoot program

The screenshot shows a Go development environment with several tabs open. The main tab displays a Go program named 'main.go' from a directory 'modul10\soal3B'. The code implements a function to check if a number is prime by counting its factors. It uses a for loop to iterate through possible divisors, printing them and incrementing a factor count if they divide the number evenly. After the loop, it checks if the factor count is exactly 2, which would mean the number is prime. The output window shows the execution of the program with inputs 12 and 7, and outputs the factors and the result of the primality check.

```
package main

import "fmt"

func main() {
    var bilangan int
    var prima bool
    var jumlahFaktor int = 0

    fmt.Println("Bilangan : ")
    fmt.Scan(&bilangan)

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

    fmt.Println()

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

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

Bilangan : 12  
Faktor : 1 2 3 4 6 12  
Prima: false  
PS D:\Coding\Latihan Go 1> go run "d:\Coding\Latihan Go 1\modul10\soal3B\main.go"  
Bilangan : 7  
Faktor : 1 7  
Prima: true

## Deskripsi program

Program ini merupakan pengembangan lebih lanjut dari soal 3A yang bertujuan untuk menentukan apakah sebuah angka adalah bilangan prima atau bukan. Logika intinya masih sama, yaitu menggunakan perulangan for untuk mencari faktor-faktor bilangan. Namun, program ini menambahkan variabel jumlahFaktor yang bertugas menghitung ada berapa banyak faktor yang ditemukan. Setelah perulangan selesai, program melakukan evaluasi akhir: jika jumlahFaktor tepat berjumlah 2 (karena bilangan prima hanya bisa dibagi oleh angka 1 dan dirinya sendiri), maka variabel boolean prima akan bernilai true, dan sebaliknya false.

Hubungannya dengan kode sebelumnya adalah kode ini memanfaatkan data yang dihasilkan oleh kode sebelumnya untuk menarik kesimpulan baru. Jika program sebelumnya ("Soal 3A") hanya berfungsi untuk menampilkan (output) deretan angka faktor, program ini ("Soal 3B") memproses data tersebut lebih lanjut dengan cara menghitungnya. Kode sebelumnya adalah dasar algoritmanya (mekanisme pencarian), sedangkan kode ini adalah implementasi logika matematika untuk mengklasifikasikan bilangan tersebut berdasarkan hasil pencarian faktor tadi.