

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 10

ELSE-IF



Disusun oleh:

GAMALIEL ALBERT NATANAEL SIMANJUNTAK
109082500067
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:

- Explorer:** Shows files in the directory `LAPRAK MODUL 10 GAMALIEL ANS`, including `guided1.go`, `Modul_10.pdf`, and `tp2.go`.
- Code Editor:** Displays the content of `guided1.go` with syntax highlighting.
- Terminal:** Shows the command `go run "d:\Laprak Modul 10 Gamaliel ANS"` being run in a Windows command prompt. The output indicates:
 - Line 17: `true`
 - Line 18: `bisa membuat KTP`
 - Line 19: `PS D:\Laprak Modul 10 Gamaliel ANS> go run "d:\Laprak Modul 10 Gamaliel ANS"`
 - Line 20: `false`
 - Line 21: `belum bisa membuat KTP`
 - Line 22: `PS D:\Laprak Modul 10 Gamaliel ANS> go run "d:\Laprak Modul 10 Gamaliel ANS"`
 - Line 23: `15`
 - Line 24: `true`
 - Line 25: `belum bisa membuat KTP`
 - Line 26: `PS D:\Laprak Modul 10 Gamaliel ANS>`
- Output:** A floating terminal window displays personal information:
 - NIM: 109082500067
 - KELAS: S1IF-13-04
 - NAMA: GAMALIEL ALBERT NATANIAEL SIMANJUNTAK

Deskripsi program

Program ini adalah validasi syarat pembuatan KTP menggunakan bahasa Go.

Program meminta dua input: Usia dan status KK (Kartu Keluarga). Menggunakan logika DAN (`&&`), program hanya akan mengizinkan pembuatan KTP jika kedua syarat terpenuhi sekaligus:

1. Usia minimal 17 tahun.
2. Memiliki KK (bernilai true).

Jika salah satu syarat tidak terpenuhi, program akan menyatakan pengguna 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

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

- File Explorer:** Shows files in the "LAPRAK MODUL 10 GAMALIEL ANS" folder: guided1.go (1), guided2.go (1), and Modul_10.pdf.
- Code Editor:** The main window displays the content of guided2.go. The code reads a character from standard input, checks if it's a vowel ('a', 'i', 'u', 'e', 'o') or a consonant ('A', 'I', 'U', 'E'), and prints the result. It also handles non-alphabetic characters.
- Terminal:** The terminal window shows the execution of the program. It runs "go run" on guided2.go, then enters a loop where it repeatedly asks for input and prints the classification ("vokal", "konsonan", or "bukan huruf").
- Output:** The output pane shows the results of the classifications.
- Bottom Status Bar:** Includes information like file path (PS D:\Laprak Modul 10 Gamaliel ANS), line and column numbers (Ln 3, Col 33), tab size (4), encoding (UTF-8), and version (1.25.1).

Deskripsi program

Program ini berfungsi untuk mengidentifikasi jenis karakter yang diinputkan oleh pengguna.

1. **Input:** Program membaca satu karakter (rune).
2. **Definisi Logika:**
 - **huruf:** Mengecek apakah karakter adalah abjad (A-Z atau a-z).
 - **vKecil / vBesar:** Mengecek apakah karakter adalah huruf vokal (a, i, u, e, o).
3. **Keputusan (Output):**
 - **"vokal":** Jika input tergolong huruf DAN tergolong vokal.
 - **"konsonan":** Jika input tergolong huruf tetapi bukan vokal.
 - **"bukan huruf":** Jika input berupa angka, simbol, atau tanda baca.

Singkatnya, program ini memilah input menjadi tiga kategori: Vokal, Konsonan, atau Bukan Huruf.

3. Guided 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) / 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 details:

- File Explorer:** Shows files in the 'LAPRAK MODUL 10 GAMALIEL ANS' folder: guided1.go (2), guided2.go (1), and guided3.go (1). A warning message is displayed: "D:\Laprap Modul 10 Gamaliel ANS\guided3.go - 1 problem in this file".
- Code Editor:** The 'guided3.go' file is open, displaying the Go code provided in the text block above.
- Terminal:** The terminal window shows the execution of the Go code. It first asks for a number ('Bilangan: ') and then prints the digits ('Digit pada bilangan'). The output for the number 2489 is:


```

$ bukan huruf
PS D:\Laprap Modul 10 Gamaliel ANS> go run "d:\Laprap Modul 10 Gamaliel ANS\guided3.go"
Bilangan: 2489
Digit pada bilangan 2489 tidak terurut
PS D:\Laprap Modul 10 Gamaliel ANS> go run "d:\Laprap Modul 10 Gamaliel ANS\guided3.go"
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS D:\Laprap Modul 10 Gamaliel ANS> go run "d:\Laprap Modul 10 Gamaliel ANS\guided3.go"
Bilangan: 9651
Digit pada bilangan 9651 tidak terurut
PS D:\Laprap Modul 10 Gamaliel ANS> []
      
```
- Status Bar:** Shows the current file path: "D:\Laprap Modul 10 Gamaliel ANS\guided3.go", line numbers (Ln 3, Col 42), character count (75 characters), encoding (Plain text), and file size (100%). It also indicates "Connected to Discord".

Deskripsi program

Program ini bertujuan untuk menganalisis pola urutan angka di dalam sebuah bilangan (khususnya bilangan 4 digit).

- 1. Ekstraksi Digit:** Program memecah bilangan input menjadi 4 bagian terpisah (d_1, d_2, d_3, d_4) menggunakan operasi sisa bagi (modulus %) dan pembagian integer.
 - *Tujuannya:* Mendapatkan satuan, puluhan, ratusan, dan ribuan.
- 2. Evaluasi Logika:**
 - **Membesar:** Mengecek apakah digit kiri lebih kecil dari kanan secara berurutan ($d_1 < d_2 < d_3 < d_4$). Contoh: 1234.
 - **Mengcil:** Mengecek apakah digit kiri lebih besar dari kanan secara berurutan ($d_1 > d_2 > d_3 > d_4$). Contoh: 4320.
 - **Tidak Terurut:** Jika tidak memenuhi kedua kondisi di atas.
- 3. Output:** Menampilkan hasil status urutan digit tersebut ke layar.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

var beratTotal, kg, sisaGram, biayaKg, biayaSisa, totalBiaya
int

fmt.Print("Berat parsel (gram): ")

fmt.Scan(&beratTotal)

kg = beratTotal / 1000

sisaGram = beratTotal % 1000

biayaKg = kg * 10000

if sisaGram >= 500 {

    biayaSisa = sisaGram * 5

} else {

    biayaSisa = sisaGram * 15

}

if beratTotal > 10000 {

    biayaSisa = 0

}
```

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

Screenshot program

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a tree view of files under "LAPRAK MODUL 10 GAMALIEL ANS". The "guided1.go" file is open.
- Code Editor:** Displays the "guided1.go" file content:

```
package main
import "fmt"
func main() {
    var beratTotal, kg, sisaGram, biayaKg, biayaSisa, totalBiaya int
    fmt.Println("Berat parcel (gram): ")
    fmt.Scan(&beratTotal)
    kg = beratTotal / 1000
    sisaGram = beratTotal % 1000
    biayaKg = kg * 1000
    if sisaGram >= 500 {
        biayaSisa = sisaGram * 5
    } else {
```
- Terminal:** Shows command-line output for running the Go code:

```
PS D:\Laprak Modul 10 Gamaliel ANS> go run "d:\Laprak Modul 10 Gamaliel ANS\guided1.go"
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500

PS D:\Laprak Modul 10 Gamaliel ANS> go run "d:\Laprak Modul 10 Gamaliel ANS\tugas1.go"
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750

PS D:\Laprak Modul 10 Gamaliel ANS> go run "d:\Laprak Modul 10 Gamaliel ANS\tugas1.go"
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
```
- Output Panel:** Displays student information:

```
NIM:109082500067
KELAS:S1IF-13-04
NAMA:GAMALIEL ALBERT NATANAEL SIMANJUNTAK
```
- Status Bar:** Shows file path "D:\Laprak Modul 10 Gamaliel ANS", line 22, column 2, and tabs like "File Size: 4", "UTF-8", "CRLF", "Go", "1.25.1", and "Prettier".

Deskripsi program

Program ini berfungsi sebagai kalkulator biaya pengiriman yang menerima input berat parcel dalam gram.

Logika perhitungannya adalah sebagai berikut:

1. **Konversi Satuan:** Program memisahkan input menjadi satuan Kilogram (kg) dan sisa gram.
 2. **Biaya Utama:** Dikenakan Rp. 10.000 untuk setiap 1 kg.
 3. **Biaya Sisa Gram:**
 - o Jika sisa ≥ 500 gram, biayanya Rp. 5/gram.
 - o Jika sisa < 500 gram, biayanya Rp. 15/gram.
 4. **Pengecualian (Diskon):** Jika total berat melebihi 10 kg, maka biaya untuk sisa gram ditratiskan (Rp. 0).
 5. **Output:** Menampilkan rincian berat (kg + gram), rincian biaya, dan total biaya akhir.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Println("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)
}
```

Screenshot program

The screenshot shows a code editor interface with several tabs open. The main tab displays a Go program named `tugas2.go`. The code defines a function `main()` that reads a float value from the user and prints its grade based on a specific scale. The terminal window below shows the execution of the program with various inputs and their corresponding outputs.

```
File Edit Selection View Go Run ... ← → ⌂ Laprak Modul 10 Gamaliel ANS
EXPLORER LAPRAK MODUL 10 GAMALIEL ANS
guided1.go 4
guided2.go 1
guided3.go 1
Modul_10.pdf
tugas1.go 1
tugas2.go 1
tugas2.go > fx main
1 package main
2 import "fmt"
3 func main() {
4     var nam float64
5     var nmk string
6     fmt.Println("Nilai akhir mata kuliah: ")
7     fmt.Scan(&nam)
8     if nam > 80 {
9         nmk = "A"
10    } else if nam > 72.5 {
11        nmk = "AB"
12    } else if nam > 65 {
13        nmk = "B"
14    } else if nam > 57.5 {
15        nmk = "BC"
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
Nilai akhir mata kuliah: 93.5
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS D:\Laprak Modul 10 Gamaliel ANS> go run "d:\Laprak Modul 10 Gamaliel ANS\tugas2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS D:\Laprak Modul 10 Gamaliel ANS> go run "d:\Laprak Modul 10 Gamaliel ANS\tugas2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS D:\Laprak Modul 10 Gamaliel ANS>
File Edit View AD KELAS:SIIF-13-04 NAMA:GAMALIEL ALBERT NATANIAEL SIMANJUNTAK
NIM:189082500067
Ln 3, Col 42 75 caracte Plain t 100% Wind: UTF-8
Ln 19, Col 18 Tab Size: 4 UTF-8 CRLF { Go 125.1 Prettier
Connected to Discord
```

Deskripsi program

a. Analisis Keluaran Program (Input 80.1)

- **Keluaran:** Program akan mencetak "Nilai mata kuliah: D".
- **Apakah sesuai spesifikasi?** TIDAK, eksekusi program tersebut salah dan tidak sesuai spesifikasi.
- **Alasannya:** Berdasarkan tabel, nilai 80.1 (lebih besar dari 80) seharusnya mendapatkan nilai "A". Namun, karena struktur kode yang salah (dijelaskan di poin b), nilai variabel nmk terus tertimpa hingga kondisi terakhir yang terpenuhi.

b. Kesalahan Program dan Penjelasannya

Kesalahan Utama:

Program menggunakan struktur if terpisah (independent if statements) secara berurutan, bukan if-else if.

Mengapa Demikian (Alur Eksekusi yang Salah):

Komputer mengecek setiap kondisi satu per satu dari atas ke bawah, meskipun kondisi sebelumnya sudah benar. Mari kita telusuri untuk input 80.1:

1. **if nam > 80 (Benar)** \rightarrow nmk diisi "A".
2. **Program lanjut ke baris berikutnya: if nam > 72.5 (Benar, karena 80.1 > 72.5)** \rightarrow nmk ditimpa menjadi "AB".
3. **Lanjut lagi: if nam > 65 (Benar)** \rightarrow nmk ditimpa menjadi "B".
4. **... (terus berlanjut menimpa nilai) ...**
5. **Sampai pada if nam > 40 (Benar)** \rightarrow nmk ditimpa menjadi "D".
6. **Baru pada bagian else if nam <= 40 program berhenti (karena if pasangannya sudah benar).**
7. **Hasil akhirnya yang tercetak adalah "D".**

Alur Program Seharusnya:

Program harus menggunakan struktur if - else if. Dengan struktur ini, begitu satu kondisi terpenuhi (misalnya > 80), program akan berhenti mengecek kondisi di bawahnya dan langsung mencetak hasil.

c. Hasil Uji Coba dengan Kode Perbaikan:

1. Input 93.5: Masuk ke if nam > 80 , set "A", selesai. (Benar)
2. Input 70.6: > 80 (salah), > 72.5 (salah), > 65 (benar), set "B", selesai. (Benar)
3. Input 49.5: Cek terus sampai > 40 (benar), set "D", selesai. (Benar)

3. Tugas 3

Source code

```
package main
import "fmt"
func main() {
    var b, i, jumlahFaktor int
    var isPrime bool
    fmt.Print("Bilangan: ")
    fmt.Scan(&b)
    fmt.Print("Faktor: ")
    for i = 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Printf("%d ", i)
            jumlahFaktor++
        }
    }
    fmt.Println()
    if jumlahFaktor == 2 {
        isPrime = true
    } else {
        isPrime = false
    }
    fmt.Printf("Prima: %t\n", isPrime)
}
```

Screenshot program

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The Explorer sidebar on the left lists files: guided1.go, guided2.go, guided3.go, Modul_10.pdf, tugas1.go, tugas2.go, and tugas3.go. The main editor area displays the following Go code:

```
package main
import "fmt"
func main() {
    var b, i, jumlahFaktor int
    var isPrime bool
    fmt.Println("Bilangan: ")
    fmt.Scan(&b)
    fmt.Println("Faktor: ")
    for i = 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Printf("%d ", i)
            jumlahFaktor++
        }
    }
    fmt.Println()
    if jumlahFaktor == 2 {
        isPrime = true
    } else {
        isPrime = false
    }
}
```

The terminal below shows the output of running the program with input 12 and 7.

```
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Laprak Modul 10 Gamaliel ANS>
> go run "d:\Laprak Modul 10 Gamaliel ANS\tugas3.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Laprak Modul 10 Gamaliel ANS>
```

Deskripsi program

Program ini meminta input sebuah bilangan bulat, lalu menjalankan dua proses utama:

1. **Pencarian Faktor:** Program melakukan perulangan (loop) dari 1 hingga bilangan input. Jika bilangan input habis dibagi angka tersebut, maka angka itu dicetak sebagai faktor.
2. **Pengecekan Prima:** Program menghitung jumlah faktor yang ditemukan.
 - o Jika jumlah faktornya tepat 2 (hanya 1 dan bilangan itu sendiri), program menetapkan status Prima menjadi true.
 - o Jika tidak, status Prima menjadi false.Output akhirnya menampilkan deretan faktor dan status boolean (true/false) apakah bilangan tersebut prima.