

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 10

ELSE-IF



Disusun oleh:

Ichsan Maulana Muhammad

109082500093

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 Visual Studio Code interface for a Go project named 'GUIDED1'. The Explorer sidebar lists several files under 'MODUL 4' and 'MODUL 5 DAN 6'. The 'GUIDED1.go' file is open in the editor, containing the following Go 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")
    }
}
```

The terminal window shows the execution of the program with different inputs:

```
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 10\GUIDED1.go"
17 true
bisa membuat KTP
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 10\GUIDED1.go"
20 false
belum bisa membuat KTP
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 10\GUIDED1.go"
15 true
belum bisa membuat KTP
PS C:\Users\ASUS\yy>
```

A separate 'logUp' window displays personal information:

```
NAMA: ICHSAN MAULANA M
NIM: 109082500093
KELAS: IF-13-04
```

Deskripsi program

1. Program diawali dengan deklarasi package main, kemudian melakukan import terhadap package "fmt" yang digunakan untuk input dan output di dalam program. Fungsi main() menjadi fungsi utama tempat seluruh proses dijalankan.
2. Mendeklarasikan dua variabel, yaitu usia dengan tipe data int untuk menampung input usia pengguna, dan kk dengan tipe data bool untuk menampung informasi apakah pengguna memiliki kartu keluarga atau tidak.
3. Program meminta pengguna memasukkan dua input pada baris terpisah, yaitu usia (bilangan bulat) dan status kepemilikan KK (true/false). Kedua input tersebut dibaca menggunakan fmt.Scan(&usia, &kk).
4. Program melakukan pengecekan syarat pembuatan KTP menggunakan struktur seleksi if, pengguna bisa membuat KTP jika memenuhi dua kondisi:
 - Usia ≥ 17 tahun
 - Memiliki kartu keluarga ($kk == \text{true}$)
5. Jika salah satu atau kedua syarat tidak terpenuhi, maka kondisi dianggap belum bisa membuat KTP.
6. Program menampilkan hasil pengecekan menggunakan fmt.Println():
 - Menampilkan "bisa membuat KTP" jika kedua syarat terpenuhi.
 - Menampilkan "belum bisa membuat KTP" jika tidak memenuhi syarat.
7. Output program berupa teks yang menyatakan apakah pengguna dapat membuat KTP atau tidak, sesuai dengan data usia dan kepemilikan KK yang dimasukkan.

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 multiple tabs open. The active tab contains a Go program named `GUIDED2.go`. The code defines a package `main` with a `main()` function. This function reads a character from input using `fmt.Scanf("%c", &x)`. It then checks if the character is a vowel ('a'-'e' or 'A'-'E') and if it is a small letter ('a'-'z'). If both conditions are true, it sets `vKecil` to `true`. Otherwise, it sets `vBesar` to `true`. Finally, it prints the result: "vokal" for vowels, "konsonan" for non-vowels, or "bukan huruf" for non-alphabetic characters. The code editor also shows a terminal window displaying the output of running the program with various inputs like 'a', 'A', 'e', 'I', 'U', 'E', 'O', '1', 'f', and '3'. A sidebar on the left lists other Go files in the project.

```
File Edit Selection View Go Run Terminal Help <- > GUIDED1.go 1 GUIDED2.go 1 guided two.go 1 guided three.go 1 tugas one.go 1 tugas two.go 1 tugas three.go 1 tugas 1.go ... EXPLORER MODUL 5 DAN 6 tugas four.go tugas one.go tugas three.go tugas two.go MODUL 7 guided 1.go guided 3.go guided 4.go tempCodeRunnerFile.go tugas 1.go tugas 2.go tugas 3.go MODUL 9 abcdefwygdeuhv.go Guided 1.go Guided 2.go Guided 3.go pemanasan1.go pemanasan2.go pemanasan3.go pemanasan4.go pemanasan5.go pemanasan6.go Pendahuluan 5.go test.go Tugas 1.go ... MODUL 10 GUIDED1.go GUIDED2.go ... PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS logUp File Edit View NAMA: ICHSAN MAULANA M NIM: 109082500093 KELAS: IF-13-04 PS C:\Users\ASUS\yy> go run "c:/Users/ASUS/yy/MODUL 10/GUIDED2.go" A vokal PS C:\Users\ASUS\yy> go run "c:/Users/ASUS/yy/MODUL 10/GUIDED2.go" f konsonan PS C:\Users\ASUS\yy> go run "c:/Users/ASUS/yy/MODUL 10/GUIDED2.go" 1 bukan huruf PS C:\Users\ASUS\yy> go run "c:/Users/ASUS/yy/MODUL 10/GUIDED2.go" $ bukan huruf PS C:\Users\ASUS\yy> ... Ln 3, Col 16 56 caracte Plain t 100% Windows UTF-16 LE Ln 20, Col 1 Tab Size: 4 UTF-8 CRLF ⌂ Go ⌂ 125.19 ⌂
```

Deskripsi program

1. Program diawali dengan deklarasi package main dan import package "fmt", yang digunakan untuk menangani proses input dan output. Fungsi main() menjadi pusat eksekusi seluruh perintah program.
2. Program mendeklarasikan satu variabel input bernama x bertipe rune untuk menampung satu karakter yang dimasukkan oleh pengguna. Selain itu terdapat tiga variabel bertipe bool, yaitu huruf, vKecil, dan vBesar yang digunakan untuk menyimpan hasil pengecekan jenis karakter.
3. Program membaca satu input berupa sebuah karakter menggunakan `fmt.Scanf("%c", &x)`.
4. Program melakukan pengecekan pertama untuk menentukan apakah input termasuk huruf atau bukan.
Kondisi huruf bernilai true jika x berada pada rentang huruf kecil 'a'-'z' atau huruf besar 'A'-'Z'.
5. Program kemudian mengecek apakah karakter tersebut merupakan huruf vokal, baik vokal kecil (a, i, u, e, o) maupun vokal besar (A, I, U, E, O).
Hasil pengecekan disimpan ke variabel vKecil dan vBesar.
6. Program menggunakan struktur seleksi if–else untuk menentukan keluaran:
 - Jika karakter adalah huruf dan termasuk vokal, program menampilkan "vokal".
 - Jika karakter adalah huruf tetapi bukan vokal, program menampilkan "konsonan".
 - Jika karakter bukan huruf, program menampilkan "bukan huruf".

7. Output program berupa teks yang menyatakan apakah karakter yang dimasukkan merupakan vokal, konsonan, atau bukan huruf, sesuai dengan hasil pengecekan yang dilakukan berdasarkan input pengguna.

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)

    d1 = bilangan / 1000
    d2 = bilangan % 1000 / 100
    d3 = bilangan % 1000 % 100 / 10
    d4 = bilangan % 1000 % 100 % 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)
    }
}

```

Screenshot program

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

- EXPLORER:** Shows a file tree with various Go files and modules.
- EDITOR:** Displays the source code for `GUIDED3.go`. The code defines a `main` package with a `main` function that prints the digits of a given number in ascending order if they are in ascending order, descending order if they are in descending order, or not sorted if they are not in either order.
- TERMINAL:** Shows command-line output for running the program with different inputs (2489, 3861, 9651) and displaying the results.
- LOG:** A separate window titled "logUp" displays the user's name, NIM, and class information.

Deskripsi program

1. Program diawali dengan deklarasi package main serta import package "fmt", yang digunakan untuk melakukan proses input dan output. Seluruh logika utama dijalankan di dalam fungsi main().
2. Program mendeklarasikan variabel bilangan bertipe int sebagai input, serta empat variabel tambahan d1, d2, d3, dan d4 yang digunakan untuk menampung masing-masing digit dari bilangan empat angka tersebut. Selain itu terdapat variabel teks bertipe string untuk menyimpan hasil penentuan jenis urutan digit.
3. Program meminta pengguna memasukkan sebuah bilangan empat digit melalui `fmt.Print()` dan membacanya menggunakan `fmt.Scan(&bilangan)`.
4. Program memecah bilangan menjadi empat digit:
 - d4 = digit satuan
 - d3 = digit puluhan
 - d2 = digit ratusan
 - d1 = digit ribuan
 Pemecahan dilakukan dengan operasi modulus dan pembagian.
5. Program melakukan pengecekan pola urutan digit menggunakan struktur seleksi if–else:
 - Jika $d1 < d2 < d3 < d4$, maka digit dianggap terurut membesar.

- Jika $d_1 > d_2 > d_3 > d_4$, maka digit dianggap terurut mengecil.
 - Jika tidak memenuhi kedua pola tersebut, maka digit dinyatakan tidak terurut.
6. Hasil pengecekan disimpan dalam variabel teks, yang berisi salah satu dari tiga kemungkinan hasil: "terurut membesar", "terurut mengecil", atau "tidak terurut".
 7. Program menampilkan hasil akhir dengan `fmt.Println()`, berupa teks: "Digit pada bilangan <bilangan> <teks>", yang menjelaskan jenis urutan digit berdasarkan input pengguna.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var gram, sisa, kg, total int
    fmt.Println("Berat Parsel: ")
    fmt.Scan(&gram)
    kg = gram / 1000
    sisa = gram % 1000
    fmt.Println("Detail Berat:", kg, "kg", "+", sisa, "gr")
    kg = kg * 10000
    if kg > 100000 {
        sisa = sisa * 0
    } else if kg <= 100000 && sisa >= 500 {
        sisa = sisa * 5
    } else if kg <= 100000 && sisa < 500 {
        sisa = sisa * 15
    }
    fmt.Println("Detail Biaya:", "Rp.", kg, "+", "Rp.", sisa)
    total = kg + sisa
    fmt.Println("Total Biaya:", "Rp.", total)
}
```

Screenshot program

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

- EXPLORER:** Shows various Go files and packages like MODUL 4, MODUL 5 DAN 6, etc.
- TUGAS1.go:** The main file being edited, containing a function `main()` that calculates shipping costs based on weight in grams. The code uses `fmt.Println` for output and `fmt.Scan` for input.
- TERMINAL:** Shows the command `go run "c:\Users\ASUS\yy\MODUL 10\TUGAS1.go"` being run, and the resulting output detailing the calculation for a 9250g parcel.
- PROBLEMS:** No problems found.
- OUTPUT:** Shows the detailed shipping cost calculation.
- DEBUG CONSOLE:** Not visible.
- TERMINAL:** Active tab.
- PORTS:** Not visible.
- RIGHT SIDE:** A separate window titled "logUp" displays personal information: NAMA: ICHSAN MAULANA M, NIM: 109082500093, KELAS: IF-13-04.

Deskripsi program

1. Program diawali dengan deklarasi package main serta import package "fmt", yang digunakan untuk menampilkan output ke layar dan membaca input dari pengguna. Seluruh proses perhitungan ditulis dan dijalankan di dalam fungsi `main()`.
2. Program mendeklarasikan beberapa variabel, yaitu gram untuk menampung input berat parsel dalam gram, kg untuk menyimpan hasil konversi berat ke satuan kilogram, sisa untuk menyimpan sisa berat di bawah 1 kg, dan total untuk menyimpan hasil akhir perhitungan biaya. Selain itu terdapat variabel biayaSisa yang dipakai untuk menampung hasil perhitungan biaya tambahan berdasarkan nilai sisa gram.
3. Program meminta pengguna memasukkan berat parsel dalam satuan gram melalui `fmt.Print()` dan membaca nilai tersebut menggunakan `fmt.Scan(&gram)`.
4. Program mengonversi berat gram menjadi dua bagian:
 - $kg = \text{gram} / 1000 \rightarrow$ untuk mendapatkan berat dalam satuan kilogram
 - $sisa = \text{gram} \% 1000 \rightarrow$ untuk mendapatkan sisa gram di bawah 1 kgPemecahan dilakukan menggunakan operasi pembagian dan modulus.
5. Program menentukan biaya kirim menggunakan struktur seleksi if–else:
 - Jika $kg > 10$, maka sisa gram tidak dikenakan biaya apa pun dan biayaSisa bernilai 0.
 - Jika $kg \leq 10$ dan $sisa > 500$, maka biayaSisa dihitung dengan rumus $sisa \times 5$.
 - Jika $kg \leq 10$ dan $sisa \leq 500$, maka biayaSisa dihitung dengan rumus $sisa \times 15$.

6. Program kemudian menghitung biaya total pengiriman dengan menjumlahkan biaya utama ($\text{kg} \times 10000$) dan biayaSisa yang diperoleh dari kondisi sebelumnya.
7. Program menampilkan hasil akhir berupa rincian biaya dan total biaya kirim menggunakan `fmt.Println()`, yang mencakup:
 - tampilan berat dalam bentuk “kg + sisa gr”
 - rincian biaya kg serta biaya sisa
 - total biaya pengiriman yang harus dibayar.

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"
    }
    if nam > 72.5 && nam <= 80 {
        nmk = "AB"
    }
    if nam > 65 && nam <= 72.5 {
        nmk = "B"
    }
}
```

```

if nam > 57.5 && nam <= 65 {

    nmk = "BC"

}

if nam > 50 && nam <= 57.5 {

    nmk = "C"

}

if nam > 40 && nam <= 50 {

    nmk = "D"

} else if nam <= 40 {

    nmk = "E"

}

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

}

```

Screenshot program

The screenshot shows the Visual Studio Code (VS Code) interface. On the left is the Explorer sidebar displaying a file tree with various Go files. The main editor area contains the Go code provided above. Below the editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing command-line output from running the code. The output pane on the right displays the results of the program execution.

```

File Edit Selection View Go Run Terminal Help ← → 🔍 yy
MODUL 10 > ⚡️ tugass2.go 1 ⚡️ pemanasan2.go 1 ⚡️ pemanasan3.go 5 ⚡️ pemanasan4.go ⚡️ pemanasan5.go 1 ⚡️ pemanasan6.go 1 ⚡️ tugass2.go 1 × ⚡️
EXPLORER
MODUL 3
MODUL 4
MODUL 5 DAN 6
MODUL 7
MODUL 9
MODUL 10
GUIDED1.go
GUIDED2.go
GUIDED3.go
tempCodeRunnerFile.go
TUGAS1.go
tugass.go
Guided 1.go
Guided 2.go
Guided 3.go
pemanasan1.go
pemanasan2.go
pemanasan3.go
pemanasan4.go
pemanasan5.go
pemanasan6.go
Pendahuluan 5.go
test.go
Tugas 1.go
Tugas 2.go
Tugas 3.go
Tugas 4.go
OUTLINE
TIMELINE
GO
PACKAGE OUTLINE
PROBLEMS 0
OUTPUT DEBUG CONSOLE TERMINAL PORTS
File Edit View logUp
File Edit View
NAMA: ICHSAN MAULANA M
NIM: 109082500093
KELAS: IF-13-04
Ln 3, Col 16 56 character Plain t 100% Windows UTF-16 LE
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 10\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 10\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 10\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\Users\ASUS\yy>
Ln 26, Col 18 Tab Size: 4 UFT-8 CRLF ⌂ Go ⌂ 1.25.1 ⌂

```

Deskripsi program

1. Program dimulai dengan deklarasi package main dan import package "fmt", yang digunakan untuk melakukan proses input dan output pada bahasa Go. Seluruh proses utama dijalankan di dalam fungsi main().
2. Program mendeklarasikan dua variabel, yaitu:
 - nam bertipe *float64* untuk menyimpan nilai akhir mata kuliah (NAM) yang diinput pengguna.
 - nmk bertipe *string* untuk menyimpan nilai huruf (NMK) yang akan ditentukan berdasarkan standar penilaian.
3. Program meminta pengguna memasukkan nilai NAM melalui perintah `fmt.Print()` dan membaca input tersebut menggunakan `fmt.Scan(&nam)`.
4. Program menentukan nilai NMK berdasarkan interval nilai NAM dengan menggunakan beberapa kondisi if dan satu else if terakhir:
 - Jika $\text{nam} > 80$, maka NMK adalah A.
 - Jika $72.5 < \text{nam} \leq 80$, maka NMK adalah AB.
 - Jika $65 < \text{nam} \leq 72.5$, maka NMK adalah B.
 - Jika $57.5 < \text{nam} \leq 65$, maka NMK adalah BC.
 - Jika $50 < \text{nam} \leq 57.5$, maka NMK adalah C.
 - Jika $40 < \text{nam} \leq 50$, maka NMK adalah D.
 - Jika $\text{nam} \leq 40$, maka NMK adalah E. Setiap kondisi mengecek interval NAM sesuai tabel penilaian yang diberikan.
5. Hasil nilai huruf yang sudah disesuaikan dengan interval disimpan ke variabel nmk.
6. Program menampilkan nilai huruf (NMK) ke layar menggunakan `fmt.Println()` dalam format:
 - Nilai mata kuliah: <nmk>

Jawablah pertanyaan-pertanyaan berikut:

- a. Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal? Jawab: D, tidak sesuai program sama soalnya
- b. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya! Jawab: 1. Thenny yang harusnya nmk tapi malah nma
2. petik duanya salah semua 3. Kondisi ifnya kurang
- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'. Jawab: udah buktinya diatas

3. Tugas 3

Source code

```
package main

import "fmt"

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

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

    if a == 2 {
        fmt.Println("Prima: true")
    } else {
        fmt.Println("Prima: false")
    }
}
```

Screenshot program

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

- Explorer:** Shows various Go files in the current workspace, including `TUGAS3.go`.
- Code Editor:** Displays the content of `TUGAS3.go`. The code defines a package `main` with a `main()` function that prints factors of a given number `b` and checks if it's prime.
- Terminal:** Shows command-line output for running the program. It prints "Bilangan: 12", lists factors "Faktor: 1 2 3 4 6 12", and checks primality "Prima: false".
- Log Viewer:** Shows log entries for the application, including the user's name, NIM, and class.

Deskripsi program

1. Program dimulai dengan deklarasi package main dan import package "fmt", yang digunakan untuk melakukan proses input dan output dalam bahasa Go. Seluruh proses utama dijalankan di dalam fungsi main().
2. Program mendeklarasikan satu variabel input:
 - `b` bertipe int, yang digunakan untuk menyimpan bilangan bulat yang dimasukkan oleh pengguna.

Selain itu, program menggunakan variabel tambahan:

- `a` bertipe int, yang dipakai sebagai penghitung jumlah faktor dari bilangan `b`.
3. Program meminta pengguna memasukkan sebuah bilangan bulat melalui `fmt.Print()`, kemudian membaca nilai tersebut menggunakan `fmt.Scan(&b)`.
 4. Setelah menerima nilai `b`, program menjalankan proses pencarian semua faktor bilangan dengan perulangan `for i := 1; i <= b; i++`. Pada setiap iterasi, program memeriksa apakah `i` merupakan faktor dari `b` dengan kondisi:
 - `if b % i == 0`
Jika kondisi terpenuhi, maka:
 - nilai `i` ditampilkan sebagai faktor, dan
 - variabel hitung `a` ditambah 1 (`a++`) untuk mencatat jumlah faktor yang ditemukan.
 5. Setelah seluruh faktor ditampilkan, program menentukan apakah bilangan tersebut prima.

Penentuan dilakukan menggunakan struktur if–else berikut:

- Jika nilai $a == 2$, maka bilangan tersebut prima, karena hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri.
 - Jika a tidak sama dengan 2, maka bilangan tersebut bukan prima.
6. Program menampilkan hasil pengecekan bilangan prima ke layar menggunakan `fmt.Println()` dalam format:
- Prima: true jika $a == 2$
 - Prima: false jika selain itu