

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 10**

**[ELSE IF]**



**Disusun oleh:**

**Rasya Putra Wibowo**

**109082500132**

**S1IF-13-02**

**Asisten Praktikum**

Adithana dharma putra

Alfin Ilham Berlianto

**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 punyaKK bool

    fmt.Scan(&usia)

    fmt.Scan(&punyaKK)

    if usia >= 17 && punyaKK {

        fmt.Println("bisa buat 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 tree view of files under "TUGAS LAPORAN PRAKTIKUM". The "guide1.go" file is selected.
- Editor:** Displays the source code of "guide1.go".

```
1 package main
2 import "fmt"
3
4 func main() {
5     var usia int
6     var punyaKK bool
7     fmt.Scan(&usia)
8     fmt.Scan(&punyaKK)
9     if usia >= 17 && punyaKK {
10         fmt.Println("bisa buat KTP")
11     } else {
12         fmt.Println("belum bisa membuat ktp")
13     }
14 }
```
- Terminal:** Shows the command-line interface with the following session:

```
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM> go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM\guide1.go"
17 true
bisa buat KTP
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM> go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM\guide1.go"
20 false
belum bisa membuat ktp
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM>
```
- Status Bar:** Shows system information like temperature (25°C), battery level (Berawan), and system date/time (23/11/2025).

### Deskripsi program

Kode ini bertujuan untuk menentukan apakah seseorang memenuhi syarat untuk membuat Kartu Tanda Penduduk (KTP) di Indonesia, berdasarkan dua kriteria utama: usia dan kepemilikan Kartu Keluarga (KK)

## 2. Guided 2 Source Code

```
package main

import "fmt"

func main() {
    var n string

    fmt.Print("masukan apa aja: ")
    fmt.Scan(&n)

    if n == "a" || n == "i" || n == "u" || n == "e" || n ==
== "o" {
        fmt.Print("vokal")
    } else if (n >= "b" && n <= "z") || (n >= "B" && n <
"z") {
        fmt.Print("konsonan")
    } else {
        fmt.Print("bukan huruf")
    }
}
```

## Screenshot program

The screenshot shows a Windows desktop environment with several windows open. On the left, there's a file explorer window showing a directory structure for 'TUGAS LAPRAK' projects. The main window is a code editor displaying a Go program named 'soal1modul.go'. The code prints a character based on its vowel or consonant status. Below the code editor is a terminal window titled 'NAMA' showing command-line interactions. The terminal output includes commands like 'go run' and 'ps' along with some text input from the user. A status bar at the bottom indicates the current file is 'soal1modul.go'.

```
File Edit Selection View Go Run Terminal Help ↻ → TUGAS LAPORAN PRAKTIKUM BLACKBOX ...  
EXPLORER ... soal1modul.go soal1.go TUGAS LAPRAK 10 1 soal3.go TUGAS LAPRAK 10 1 guide1.go TUGAS LAPRAK 10 4 ... guide2.go TUGAS LAPRAK 10 1 BLACKBOX ...  
TUGAS LAPRAK 5 & 6  
main.go  
main2.go  
main3.go  
soalmodul1.go  
soalmodul2.go  
soalmodul3.go  
TUGAS LAPRAK 7  
soal1.go  
soal2.go  
soal3.go  
TUGAS LAPRAK 9  
guide1.go  
guide2.go  
guide3.go  
soal1.go  
soal2.go  
soal3.go  
soalmodul1.go  
TUGAS LAPRAK 10  
guide1.go  
guide2.go  
soal1.go  
soal2.go  
soal3.go  
soalmodul1.go  
OUTLINE  
TIMELINE  
GO  
PACKAGE OUTLINE  
B 0 🔍 BLACKBOX Agent Open Website
```

```
TUGAS LAPRAK 10 > soal1modul.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n string
7
8     fmt.Println("masukan apa aja: ")
9     fmt.Scan(&n)
10
11    if n == "a" || n == "i" || n == "u" || n == "e" || n == "o" {
12        fmt.Println("vokal")
13    } else if (n >="b" && n <= "z") || (n >="B" && n <= "Z") {
14        fmt.Println("konsonan")
15    } else {
16        fmt.Println("bukan huruf")
17    }
18 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM\soal1modul.go"
masukan apa aja: 1
bukan huruf
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM\soal1modul.go"
masukan apa aja: e
vokal
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM\soal1modul.go"
masukan apa aja: b
konsonan
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM
```

NAMA: RASYA PUTRA WIBOWO  
NIM: 189028500132

File Edit Lihat H1 ... + - x powershell S LAPRAK 10\g S LAPRAK 10\g S LAPRAK 10\g

Ln 18, Col 1 Tab Size 4 UTT-B CRLF ↻ Go 1,251 ⚡ 🔍 BLACKBOX Autocomplete: ON 🔍 BLACKBOX: Open Chat

## **Deskripsi program**

Mengidentifikasi apakah input yang dimasukkan adalah:

1. Vokal (hanya huruf kecil: a, i, u, e, o).
2. Konsonan (huruf kecil selain vokal, atau huruf kapital).
3. Bukan Huruf (angka, simbol, atau huruf vokal kapital).

### **3. Guided 3**

#### **Source Code**

```
package main

import (
    "fmt"
)

func main() {
    var bilangan int

    fmt.Println("Masukkan 4 digit bilangan:")
    fmt.Scan(&bilangan)

    if bilangan >= 1000 && bilangan <= 9999 {
        digit1 := bilangan / 1000
        digit2 := (bilangan / 100) % 10
        digit3 := (bilangan / 10) % 10
        digit4 := bilangan % 10

        if digit1 < digit2 && digit2 < digit3 && digit3 < digit4 {
            fmt.Println("Bilangan terurut membesar")
        } else if digit1 > digit2 && digit2 > digit3 && digit3 > digit4 {
            fmt.Println("Bilangan terurut mengecil")
        } else {
            fmt.Println("Bilangan terurut tidak membesar atau mengecil")
        }
    } else {
        fmt.Println("Input harus antara 1000-9999")
    }
}
```

## Screenshoot program

The screenshot shows a Windows desktop environment. In the center is a terminal window titled 'TUGAS LAPRAK 10' with the command 'go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM\guide3.go"'. The output shows the user inputting '1234' and the program responding with 'Bilangan terurut membesar'. Below the terminal is a code editor window for 'guide3.go' which contains Go code for digit comparison. To the left of the code editor is a file explorer showing various Go files in the 'TUGAS LAPORAN PRAKTIKUM' directory. The taskbar at the bottom includes icons for File Explorer, Edge browser, FileZilla, and others.

## Deskripsi program

Program ini digunakan untuk memeriksa apakah sebuah bilangan 4 digit memiliki urutan digit yang membesar, mengecil, atau tidak berurutan sama sekali.

1. Program membaca bilangan 4 digit dari pengguna.
2. Program memastikan angkanya benar-benar berada pada rentang 1000–9999.
3. Bilangan dipecah menjadi 4 digit:
  - digit1 = ribuan
  - digit2 = ratusan
  - digit3 = puluhan
  - digit4 = satuan
4. Program kemudian membandingkan keempat digit:
  - Jika  $\text{digit1} < \text{digit2} < \text{digit3} < \text{digit4}$  → terurut membesar
  - Jika  $\text{digit1} > \text{digit2} > \text{digit3} > \text{digit4}$  → terurut mengecil
  - Selain itu → tidak terurut membesar atau mengecil

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import (
    "fmt"
)
```

```

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

    kg := beratParsel / 1000
    sisa := beratParsel % 1000

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

    biayaDasar := kg * 10000

    biayaTambahan := 0

    if kg > 10 {
        biayaTambahan = 0
    } else {
        if sisa > 0 {
            if sisa < 500 {
                biayaTambahan = sisa * 5
            } else {
                biayaTambahan = sisa * 15
            }
        }
    }

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

    total := biayaDasar + biayaTambahan
    fmt.Printf("Total biaya: Rp. %d\n", total)
}

```

## Screenshot program

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

- File Explorer:** Shows a tree view of files across several packages (TUGAS LAPRAK 5, TUGAS LAPRAK 5 & 6, TUGAS LAPRAK 7, TUGAS LAPRAK 9, TUGAS LAPRAK 10) and a package outline.
- Code Editor:** Displays the source code for `soal1.go` from the `TUGAS LAPRAK 10` package.
- Terminal:** Shows command-line output for running the program. It includes the command `go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM\soal1.go"`, the input "Berat parsel (gram): 8500", and the resulting output: "Detail berat: 8 kg + 500 gr", "Detail biaya: Rp. 80000 + Rp. 7500", and "Total biaya: Rp. 87500".
- Output Panel:** Shows the standard output of the program, matching the terminal output.
- Search Bar:** Located at the top center of the interface.
- System Tray:** Shows icons for battery level (25%), system status, and network connection.

### **Deskripsi program**

Program ini menghitung biaya kirim paket berdasarkan berat total dalam gram, dengan aturan:

1. Berat diubah menjadi kilogram dan gram tambahan (lebih).  
-kg := berat / 1000  
-lebih := berat % 1000
2. Biaya per kilogram adalah Rp 10.000.  
-biayaKg = kg \* 10000
3. Bagian gram tambahan (“lebih”) dihitung dengan aturan:  
-Jika berat total > 10 kg, gram tambahan digratiskan.  
-Jika gram tambahan  $\geq$  500 gr, biaya Rp 5 per gram.  
-Jika gram tambahan < 500 gr, biaya Rp 15 per gram.  
-Jika tidak ada gram tambahan → biaya tambahan = 0.
4. Total biaya = biaya kg + biaya gram tambahan.
5. Program kemudian menampilkan:  
-Berat dalam format: x kg + y gr  
-Rincian biaya  
-Total biaya akhir

jadii

Program menghitung biaya pengiriman berdasarkan berat paket dengan aturan biaya berbeda untuk kilogram dan gram tambahan, menggunakan struktur if–else if–else untuk menentukan biaya tambahan sesuai kondisi berat.

## **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 Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The left sidebar displays a file tree with several Go source files and their corresponding mod files across different project folders named TUGAS LAPRAK 5 & 6, TUGAS LAPRAK 7, TUGAS LAPRAK 9, and TUGAS LAPRAK 10. The main editor area contains the following Go code:

```
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"
16    } else if nam >= 50 {
17        nmk = "C"
18    } else if nam >= 40 {
19        nmk = "D"
20    }
21 }
```

Below the editor is a terminal window titled 'TUGAS LAPRAK 10' showing the output of a Go run command:

```
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPRAK PRAKTIKUM> go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPRAK 10\soal2.go"
Nilai akhir mata kuliah: 93,5
Nilai mata kuliah: A
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPRAK PRAKTIKUM> go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPRAK 10\soal2.go"
Nilai akhir mata kuliah: 70,6
Nilai mata kuliah: B
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPRAK PRAKTIKUM> go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPRAK 10\soal2.go"
Nilai akhir mata kuliah: 49,5
```

The system tray at the bottom shows weather information: 'Hari hujan yang...', '25°C'. The taskbar includes icons for File Explorer, Edge browser, and other system applications.

## Deskripsi program

- Sebelum program diperbaiki, jika input nilai 80.1 maka outputnya akan error. Program sebelumnya tidak sesuai spesifikasi karena terdapat kesalahan dalam penanganan tipe data variabel.
- Kesalahan terjadi karena struktur kondisi yang digunakan tidak tepat dengan menggunakan banyak statement if secara terpisah. Hal ini menyebabkan semua kondisi akan dicek secara berurutan tanpa memperhatikan apakah kondisi sebelumnya sudah terpenuhi. Alur program yang seharusnya adalah ketika pengguna menginput nilai akhir mata kuliah, program harus mengevaluasi nilai tersebut secara berurutan dari kategori tertinggi. Jika nilai memenuhi kriteria untuk grade A, maka program langsung menetapkan nilai A dan tidak perlu melanjutkan pengecekan ke kondisi berikutnya. Apabila tidak memenuhi syarat A, program kemudian mengecek apakah nilai termasuk dalam kategori AB, kemudian B, BC, C, dan seterusnya, sampai akhirnya jatuh ke grade

E jika nilainya berada di bawah semua threshold yang ditentukan. Dengan pendekatan ini, hanya satu kategori nilai yang akan dipilih, sehingga hasil konversi nilai menjadi akurat dan sesuai ekspektasi.

- c. Program yang sudah diperbaiki dapat dilihat pada bagian screenshot program terlampir.

### 3. Tugas 3

#### Source code

```
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()

    var prima bool

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

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

#### Screenshot program

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows a tree view of files under "TUGAS LAPRAK".
- Editor:** Displays a Go program named "soal3.go". The code prints factors and checks if a number is prime.
- Terminal:** Shows the output of running the program with input 12 and 7.
- Status Bar:** Shows file paths, line numbers (Ln 19, Col 27), and terminal settings (UTF-8).

```
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var b int
9     fmt.Println("Bilangan: ")
10    fmt.Scan(&b)
11
12    fmt.Println("Faktor: ")
13
14    jumlahFaktor := 0
15
16    for i := 1; i <= b; i++ {
17        if b%i == 0 {
18            fmt.Print(i, " ")
19            jumlahFaktor++
20        }
21    }
22    fmt.Println()
23
24    var prima bool
25
26    if jumlahFaktor == 2 {
27        prima = true
28    } else {
29    }
30 }
```

```
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM> go run "c:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM\soal3.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Users\ACER\OneDrive\Documents\TUGAS LAPORAN PRAKTIKUM>
```

## Deskripsi program

Program ini digunakan untuk menampilkan semua faktor dari sebuah bilangan sekaligus menentukan apakah bilangan tersebut merupakan bilangan prima, dengan menggunakan logika if – else if dan boolean.