

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 10**

**ELSE-IF**



**Disusun oleh:**

**Anindya Rahadita Yumnaa**

**109082500138**

**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 VS Code interface with the following details:

- File Explorer:** Shows files like "guided 1.go", "guided 2.go 1", "guided 3.go", "soal 1.go", "soal 2.go", and "soal 3.go".
- Search Bar:** Contains the word "Search".
- Terminal:** Displays the command "C: > Users > MSI > Documents > VSCode > MINGGU 10 > LAPRAK > guided 1 > go run main". The code editor shows a Go program:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var usia int
8     var kk bool
9
10    fmt.Scan(&usia, &kk)
11    if usia >= 17 && kk {
12        fmt.Println("bisa membuat KTP")
13    }else{
14        fmt.Println("Belum bisa membuat KTP")
15    }
16 }
```

- Output Panel:** Shows the terminal output of the Go program execution.
- LightClock Panel:** A floating panel titled "LightClock" showing the date and time: "10/08/2500 13:07". It also displays the name "S11F-13-07 ANINDYA RAHADITA YUMNAA".
- Bottom Status Bar:** Shows file path "PS C:\Users\MSI> go run", line count "Ln 10, Col 21", tab size "Tab Size: 4", encoding "UTF-8", and other system information.

## Deskripsi program

Program sederhana ini dibuat untuk **mengecek** apakah seseorang sudah **memenuhi syarat untuk mengurus KTP**. Cara kerjanya, program akan meminta kita memasukkan **usia dan status Kartu Keluarga (KK)**, kemudian menginputkan **ya** atau **tidak** untuk KK-nya. Setelah kedua informasi itu dimasukkan, program langsung mengecek: **apakah usia sudah 17 tahun atau lebih, dan apakah benar-benar sudah punya Kartu Keluarga?** Hanya jika kedua hal itu benar, program akan memberi tahu bahwa "**bisa membuat KTP**". Jika salah satu syarat belum terpenuhi, misalnya baru berumur 16 tahun, maka hasilnya adalah "**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 the Visual Studio Code interface with a Go file named `guided 2.go` open. The code implements a function to determine if a character is a vowel or consonant based on its ASCII value. A terminal window is visible, showing the execution of the program and its output for the string "ANINDYA RAHADITA YUMNAA".

```

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")
    }
}

```

```

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 2\guided 2.go"
A
vokal
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 2\guided 2.go"
F

```

```

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'
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 2\guided 2.go"  
A  
vokal  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 2\guided 2.go"  
F  
konsonan  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 2\guided 2.go"  
1  
bukan huruf  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 2\guided 2.go"  
\$  
bukan huruf  
PS C:\Users\MSI>

### Deskripsi program

Program ini untuk **menganalisis satu karakter** yang akan dimasukkan, bisa berupa huruf, angka, atau simbol, kemudian memutuskan apakah inputan tersebut termasuk kelompok yang mana. Pertama, program akan mengecek, "Apakah ini **huruf** secara umum (A-Z atau a-z)?" Jika ya, program masuk ke tahap berikutnya. Di tahap kedua, program mengecek, "Apakah huruf ini termasuk **vokal** (A, I, U, E, O)?" Jika benar, hasilnya langsung dicap "**vokal**". Namun, jika di tahap kedua ternyata inputan adalah huruf tapi **bukan vokal**, otomatis inputan tersebut dicap sebagai "**konsonan**". Terakhir, jika dari awal karakter itu bahkan **bukan huruf sama sekali** (misalnya, '\$'), maka program akan mencapnya sebagai "**bukan huruf**". Singkatnya, ini adalah sistem otomatis untuk mengklasifikasikan input tunggal menjadi vokal, konsonan, atau lainnya.

### 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) / 100

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)
```

### Screenshoot program

The screenshot shows a VS Code interface with multiple tabs open. The active tab contains Go code for calculating digits of a number. A terminal window is open below, showing the execution of the program and its output.

```
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) / 100
    d1 = bilangan / 1000
}
```

LightClock

File Edit View

109082500138  
S1LF-13-07  
ANINDYA RAHADITA YUMNAA

Ln 3, Col 24 | 48 character Plain t | 100% Wind: UTF-8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 3\guided 3.go"
Bilangan: 2489
Digit pada bilangan 2489 terurut membesar
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 3\guided 3.go"
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\guided 3\guided 3.go"
Bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
PS C:\Users\MSI>
```

## Deskripsi program

Program ini berfungsi untuk **menganalisis susunan digit pada sebuah angka** yang dimasukkan oleh pengguna, untuk mengetahui apakah angka tersebut terurut. Pertama, program akan meminta untuk memasukkan sebuah angka bulat (yang diasumsikan empat digit). Kemudian, program akan memecah angka tersebut menjadi digit-digit penyusunnya, yaitu d1 (ribuan), d2 (ratusan), d3 (puluhan), dan d4 (satuan), menggunakan perhitungan sisa bagi (%) dan bagi (/). Setelah semua digit didapatkan, program akan **memeriksa urutannya**: jika setiap digit berikutnya **selalu lebih besar** dari digit sebelumnya (misalnya 1234), maka hasilnya adalah "**terurut membesar**". Jika setiap digit berikutnya **selalu lebih kecil** (misalnya 4321), maka hasilnya adalah "**terurut mengecil**". Selain kedua kondisi tersebut, program akan menyatakan bahwa

angka itu "**tidak terurut**". Hasil akhirnya akan ditampilkan, memberitahu bahwa status keterurutan digit pada angka yang telah dimasukkan.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {

    var berat, kg, sisa int

    var biayaKg, biayaSisa, total int

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

    fmt.Scan(&berat)

    kg = berat / 1000

    sisa = berat % 1000

    fmt.Println("Detail berat:", kg, "kg +", sisa, "gr")

    biayaKg = kg * 10000

    if sisa == 0 {

        biayaSisa = 0

    } else if sisa >= 500 {

        biayaSisa = sisa * 5

    } else {
```

```

        biayaSisa = sisa * 15

    }

    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.",
biayaSisa)

total = biayaKg + biayaSisa

fmt.Println("Total biaya: Rp.", total)

}

```

## Screenshot program

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, Find, and Copy/Paste. The main area displays a Go program:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var berat, kg, sisa int
8     var biayaKg, biayaSisa, total int
9
10    fmt.Print("Berat parcel (gram): ")
11    fmt.Scan(&berat)
12    kg = berat / 1000
13    sisa = berat % 1000
14    fmt.Println("Detail berat:", kg, "kg +", sisa, "gr")
15    biayaKg = kg * 10000
16
17    if sisa == 0 {
18        biayaSisa = 0
19    } else if sisa >= 500 {
20        biayaSisa = sisa * 5
21    } else {
22        biayaSisa = sisa * 15
23    }
24    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.", biayaSisa)
25
26    total = biayaKg + biayaSisa
27    fmt.Println("Total biaya: Rp.", total)
28

```

To the right of the code editor is a terminal window titled "LightClock". It shows the command "109082500138" being entered, followed by the output "S11F-13-07" and "ANINDYA RAHADITA YUMNAA". Below the terminal are status bars for the code editor and terminal.

```
C:\> Users > MSI > Documents > VSCode > MINGGU 10 > LAPRAK > soal 1 > -o soal 1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var berat, kg, sisa int
8     var biayaKg, biayaSisa, total int
9
10    fmt.Println("Berat parsel (gram): ")
11    fmt.Scan(&berat)
12
13    if berat <= 500 {
14        fmt.Println("Biaya pengiriman gratis!")
15        return
16    }
17
18    kg = berat / 1000
19    sisa = berat % 1000
20
21    if sisa == 0 {
22        biayaKg = kg * 10000
23        biayaSisa = 0
24    } else if sisa < 500 {
25        biayaKg = kg * 10000 + 15 * sisa
26        biayaSisa = 0
27    } else {
28        biayaKg = kg * 10000 + 5 * 500
29        biayaSisa = 5 * (sisa - 500)
30    }
31
32    total = biayaKg + biayaSisa
33
34    fmt.Println("Detail berat: ", kg, "kg +", sisa, "gr")
35    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.", biayaSisa)
36    fmt.Println("Total biaya: Rp.", total)
37
38}
39
40 PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 10/LAPRAK/soal 1/soal 1.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 10/LAPRAK/soal 1/soal 1.go"
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Users\MSI> go run "c:/Users/MSI/Documents/VSCode/MINGGU 10/LAPRAK/soal 1/soal 1.go"
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 113750
PS C:\Users\MSI>
```

### Deskripsi program

Program ini berfungsi sebagai **penghitung (kalkulator)** otomatis untuk menghitung **biaya pengiriman paket** berdasarkan beratnya dalam satuan gram. Pertama, program akan meminta untuk memasukkan **berat total parsel** yang akan dikirim (dalam gram). Begitu data berat diterima, program langsung **memecah berat tersebut** menjadi dua bagian: berat penuh dalam **kilogram (kg)** dan sisanya dalam **gram (gr)**. **Biaya Kilogram** dihitung dengan mengalikan berat penuh dalam kg dengan tarif dasar Rp10.000 per kg.

**Biaya Sisa Gram** dihitung dengan aturan berlapis:

- Jika sisa gramnya **nol** (beratnya pas per kg), biaya sisa gramnya gratis (**nol**).
- Jika sisa gramnya **500 gram atau lebih**, biaya dihitung Rp5 per gram.
- Jika sisa gramnya **kurang dari 500 gram**, biayanya lebih mahal, yaitu Rp15 per gram.

## 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 {

    nam = "A"

}

if nam > 72.5 {

    nam = "AB"

}

if nam > 65 {

    nam = "B"

}

if nam > 57.5 {

    nam = "BC"

}

if nam > 50 {

    nam = "C"

}

if nam > 40 {

    nam = "D"

} else if nam <= 40 {

    nam = "E"

}

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

```
}
```

## Screenshot program

The screenshot shows a VS Code interface with several tabs open. The active tab is 'soal 2.go' containing the following Go 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"
    }
    if nam > 72.5 {
        nmk = "AB"
    }
    if nam > 65 {
        nmk = "B"
    }
    if nam > 57.5 {
        nmk = "BC"
    }
    if nam > 50 {
        nmk = "C"
    }
    if nam > 40 {
        nmk = "D"
    } else if nam <= 40 {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}
```

To the right of the editor, a terminal window titled 'LightClock' is open, displaying the output of the program:

```
109082500138
S1IF-13-07
ANINDYA RAHADITA YUMNAA
```

The terminal also shows the status bar at the bottom.

This screenshot is similar to the previous one, but the Go code has been modified to include the final line of the program:

```
func main() {
    if nam > 72.5 {
        nmk = "AB"
    }
    if nam > 65 {
        nmk = "B"
    }
    if nam > 57.5 {
        nmk = "BC"
    }
    if nam > 50 {
        nmk = "C"
    }
    if nam > 40 {
        nmk = "D"
    } else if nam <= 40 {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}
```

The terminal window shows the same output as before:

```
109082500138
S1IF-13-07
ANINDYA RAHADITA YUMNAA
```

The status bar at the bottom of the terminal window indicates 'Ln 3, Col 24 : 48 caractere Plain t | 100% | Wind | UTF-8'.

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

- File Explorer:** Shows files like 'guided 1.go', 'guided 2.go', 'guided 3.go', 'soal 1.go', 'soal 2.go', and 'soal 3.go'.
- Editor:** Displays the code for 'soal 2.go' with several syntax errors underlined in red. The code defines a function 'main()' that maps a float64 value 'nam' to a string 'nmk' based on specific ranges.
- Terminal:** Shows the output of the program 'LightClock'. The output includes the date and time (109082500138, S1IF-13-07), the name (ANINDYA RAHADITA YUMNA), and the message "Nilai mata kuliah: " followed by a blank space.
- Problems:** A list of 12 errors related to float64 assignments to string constants.
- Status Bar:** Shows the current file is 'soal 2.go', line 32, column 45, with 48 characters, in Plain text mode, at 100% zoom, with Wind and UTF-8 encoding.

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

: Keluaran dari program tersebut tidak dapat berjalan sama sekali, karena program Go tidak akan menjalankan variabel float64 yang diisi dengan string. Kemudian untuk eksekusi program tersebut tidak memenuhi spesifikasi, yang dimana untuk float64 diisi dengan string "A", "AB", "B", "BC", "C", "D", "E".

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

: Beberapa kesalahan dari program tersebut adalah:

- 1) Tipe data yang salah (**float64** namun diisi dengan string)
- 2) Penempatan inputan variabel yang salah. Seharusnya **nam**, bukan **nmk**
- 3) Rantai kondisi tidak terhubung. Hanya dua baris terakhir yang menggunakan **else if**, sedangkan kondisi di atasnya menggunakan **if** terpisah. **else if** seharusnya terhubung dengan seluruh rantai kondisi, bukan hanya dua baris terakhir.
- 4) **if terpisah**, bukan **if – else if – else** yang saling menyambung. Akibatnya, semua kondisi yang benar akan dieksekusi, bukan hanya satu. Seperti pada kondisi, jika menginputkan 80.1 hasil akhirnya adalah "**D**", bukan "**A**". Karena semua kondisi dijalankan terus sampai yang paling bawah.

**Untuk alur program seharusnya:**

1. Program akan meminta untuk menginput nama mahasiswa
2. Program meminta input nilai mahasiswa dalam bentuk angka (float), disini float64

3. Program membandingkan nilai tersebut menggunakan **rantai kondisi yang terhubung** (if – else if – else)
  4. Hanya **satu kondisi** yang dieksekusi, yaitu kondisi pertama yang benar
  5. Program menampilkan kategori nilai berdasarkan rentang nilai tersebut
- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah ‘A’, ‘B’, dan ‘D’.
- : Berikut perbaikan dari program tersebut, beserta output yang diperoleh

```

File Edit Selection View Go Run ... ← → ⌂ Search
C: > Users > MSI > Documents > VSCode > MINGGU 10 > LAPRAK > soal 2 > soal 2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var nam float64
8     var nmk string
9     fmt.Println("Nilai akhir mata kuliah: ")
10    fmt.Scan(&nam)
11
12    if nam > 80 {
13        nmk = "A"
14    } else if nam > 72.5 {
15        nmk = "AB"
16    } else if nam > 65 {
17        nmk = "B"
18    } else if nam > 57.5 {
19        nmk = "BC"
20    } else if nam > 50 {
21        nmk = "C"
22    } else if nam > 40 {
23        nmk = "D"
24    } else {
25        nmk = "E"
26    }
27
28    fmt.Println("Nilai mata kuliah:", nmk)
29 }

```

LightClock \* + - □ ×

File Edit View

109982500138  
SIIIF-13-07  
ANINDYA RAHADITA YUMNA'A

Ln 3, Col 24 | 48 character Plain t | 100% | Wind | UTF-8

Ln 29, Col 2 | Spaces: 4 | UTF-8 | CRLF | Go | 1.25.1 | ⌂

```

File Edit Selection View Go Run ... ← → ⌂ Search
C: > Users > MSI > Documents > VSCode > MINGGU 10 > LAPRAK > soal 2 > soal 2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var nam float64
8     var nmk string
9     fmt.Println("Nilai akhir mata kuliah: ")
10    fmt.Scan(&nam)
11
12    if nam > 80 {
13        nmk = "A"
14    } else if nam > 72.5 {
15        nmk = "AB"
16    } else if nam > 65 {
17        nmk = "B"
18    } else if nam > 57.5 {
19        nmk = "BC"
20    } else if nam > 50 {
21        nmk = "C"
22    } else if nam > 40 {
23        nmk = "D"
24    } else {
25        nmk = "E"
26    }
27
28    fmt.Println("Nilai mata kuliah:", nmk)
29 }

```

LightClock \* + - □ ×

File Edit View

109982500138  
SIIIF-13-07  
ANINDYA RAHADITA YUMNA'A

Ln 3, Col 24 | 48 character Plain t | 100% | Wind | UTF-8

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\soal 2\soal 2.go"  
Nilai akhir mata kuliah: 93.5  
Nilai mata kuliah: A  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\soal 2\soal 2.go"  
Nilai akhir mata kuliah: 70.6  
Nilai mata kuliah: B  
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\soal 2\soal 2.go"  
Nilai akhir mata kuliah: 49.5  
Nilai mata kuliah: D  
PS C:\Users\MSI> []

Ln 29, Col 2 | Spaces: 4 | UTF-8 | CRLF | Go | 1.25.1 | ⌂

### Deskripsi program

Program ini berfungsi sebagai **konverter nilai otomatis** untuk mengubah nilai akhir angka menjadi **nilai huruf** (seperti A, B, C, dst.). Program akan meminta untuk memasukkan **Nilai Akhir Mata Kuliah** berupa angka (berkemungkinan juga memiliki koma). Setelah angka dimasukkan, program langsung **menguji nilai** tersebut berdasarkan serangkaian aturan secara berurutan: jika nilainya di atas 80 dapat "A"; jika di atas 72.5 dapat "AB"; di atas 65 dapat "B", dan seterusnya, dengan batas terendah di atas 40 masih dapat "D". Jika nilai tersebut bahkan tidak mencapai batas 40, maka nilai mutu yang diberikan adalah "E". Di akhir proses, program akan menampilkan **nilai huruf** yang sesuai dengan nilai angka yang telah dimasukkan.

### 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)
    }
}

```

## Screenshoot program

The screenshot shows the Visual Studio Code interface. The code editor on the left contains Go code for finding factors and checking primality. A terminal window on the right displays the output of the program for the number 12.

```

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)
}

```

Terminal Output:

```

12
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false

```

This screenshot shows the same VS Code environment. The terminal window at the bottom displays the command `go run` followed by the path to the Go file, and then the program's output for the number 7.

```

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\soal 3\soal 3.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\MINGGU 10\LAPRAK\soal 3\soal 3.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Users\MSI>

```

## Deskripsi program

Program dirancang untuk **mencari semua faktor pembagi dari sebuah bilangan** yang dimasukkan, dan sekaligus **menentukan apakah bilangan tersebut adalah bilangan prima**. Cara kerjanya dimulai dengan meminta untuk memasukkan sebuah angka

bulat. Setelah angka diinput, program akan memeriksa setiap angka mulai dari 1 hingga bilangan itu sendiri. Setiap kali sebuah angka ditemukan bisa **membagi habis** bilangan tanpa sisa (yaitu faktornya), program akan mencetak angka faktor tersebut dan mencatatnya ke dalam sebuah penghitung (jumlahFaktor). Setelah semua angka selesai diperiksa, program akan **menentukan status prima**: jika total faktor yang ditemukan **hanya ada dua** (yaitu 1 dan bilangan itu sendiri), maka bilangan tersebut dinyatakan sebagai "Prima: true"; jika jumlah faktornya lebih atau kurang dari dua, maka hasilnya "Prima: false".