

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 10**

**ELSE - IF**



**Disusun oleh:**

**SALSADILLA HANNY AZIZAH**

**109082500014**

**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 umur int
    var KK bool

    fmt.Print("Masukkan umur = ")
    fmt.Scan(&umur)

    fmt.Print("Status KK = ")
    fmt.Scan(&KK)

    if umur >= 17 && KK {
        fmt.Print("bisa membuat ktp")
    } else{
        fmt.Print("tidak bisa membuat ktp")
    }
}
```

#### Screenshoot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. On the left, the Explorer sidebar shows a project structure with files like 'guided1.go', 'guided2.go', and 'guided3.go'. The main editor area contains the following Go code:

```

package main
import "fmt"
func main() {
    var umur int
    var KK bool
    fmt.Println("Masukkan umur = ")
    fmt.Scan(&umur)
    fmt.Println("Status KK = ")
    fmt.Scan(&KK)

    if umur >= 17 && KK {
        fmt.Println("Bisa membuat ktp")
    } else{
        fmt.Println("tidak bisa membuat ktp")
    }
}

```

Below the editor is the Terminal tab, which shows the command `go run "d:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\meet9\guided1.go"` being run twice, followed by user input and the program's output: "Masukkan umur = 17" and "Status KK = true". The output "Bisa membuat ktp" is displayed.

To the right of the VS Code window is a Microsoft Edge browser window titled 'NIM 109082500014'. It displays the following text:

NIM : 109082500014  
Kelas : S1IF-13-02  
Nama : Salsadilla Hanny Azizah

At the bottom of the screen, the taskbar shows various pinned icons, and the system tray indicates the date and time as 17/11/2025 and 11:18:03.

## Deskripsi program

Program ini digunakan untuk menentukan apakah seseorang sudah memenuhi syarat untuk membuat KTP atau belum. Program meminta dua input dari pengguna, yaitu umur dan status KK. Umur diisi dengan angka, sedangkan status KK diisi dengan nilai boolean (true atau false) yang menunjukkan apakah orang tersebut sudah terdaftar dalam Kartu Keluarga. Setelah input diterima, program melakukan pengecekan. Jika umur yang dimasukkan sudah 17 tahun atau lebih, dan status KK bernilai true, maka program menampilkan bahwa orang tersebut bisa membuat KTP. Namun, jika salah satu dari dua syarat itu tidak terpenuhi misalnya umurnya belum 17 atau status KK-nya false maka hasilnya adalah tidak bisa membuat KTP. Dengan begitu, program ini secara sederhana mengecek kelayakan pembuatan KTP berdasarkan dua aturan utama: umur dan keanggotaan KK.

## 2. Guided 2

### Source Code

```

package main

import "fmt"

func main() {
    var huruf string

```

```

        fmt.Println("Masukkan = ")

        fmt.Scan(&huruf)

        if huruf < "A" || huruf > "z" {

            fmt.Println("bukan huruf")

        } else if huruf == "A" || huruf == "I" || huruf == "U" || huruf == "E" || huruf == "O" || huruf == "a" || huruf == "i" || huruf == "u" || huruf == "e" || huruf == "o" {

            fmt.Println("vokal")

        } else {

            fmt.Println("konsonan")

        }

    }

```

## Screenshot program

The screenshot shows a Windows desktop environment. On the left is a code editor window titled 'guided2.go' with the following Go code:

```

package main
import "fmt"
func main() {
    var huruf string
    fmt.Println("Masukkan = ")
    fmt.Scan(&huruf)
    if huruf < "A" || huruf > "z" {
        fmt.Println("bukan huruf")
    } else if huruf == "A" || huruf == "I" || huruf == "U" || huruf == "E" || huruf == "O" || huruf == "a" || huruf == "i" || huruf == "u" || huruf == "e" || huruf == "o" {
        fmt.Println("vokal")
    } else {
        fmt.Println("konsonan")
    }
}

```

On the right is a terminal window titled 'NIM 109C' showing the output of the program:

```

NIM : 109082500014
Kelas : S1IF-13-02
Nama : Salsadilla Hanny Azizah

```

The taskbar at the bottom includes icons for File Explorer, Task View, Start, Search, Taskbar settings, and various pinned applications.

## Deskripsi program :

Program ini digunakan untuk mengecek apakah input dari pengguna merupakan huruf vokal, huruf konsonan, atau bahkan bukan huruf sama sekali. Pengguna memasukkan sebuah karakter. Nilai tersebut disimpan dalam variabel huruf. Setelah itu, program melakukan pengecekan awal untuk memastikan apakah karakter yang dimasukkan benar-benar huruf. Caranya adalah dengan membandingkan nilai input dengan rentang "A" sampai "z". Jika berada di luar rentang itu, maka program langsung menampilkan bahwa input bukan huruf. Kalau inputnya termasuk huruf, program lanjut mengecek apakah huruf tersebut vokal. Daftar huruf vokal yang dicek mencakup huruf besar dan kecil: A, I, U, E, O, a, i, u, e, o. Jika cocok dengan salah satu dari itu, program menampilkan "vokal". Jika input sudah jelas huruf tapi bukan termasuk vokal, maka otomatis dianggap sebagai huruf konsonan. Karena itu, pada bagian terakhir program menampilkan "konsonan".

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var bilangan, ribuan, ratusan, puluhan, satuan int

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

    ribuan = (bilangan / 1000)
    ratusan = (bilangan / 100) % 10
    puluhan = (bilangan / 10 ) % 10
    satuan = (bilangan % 10)

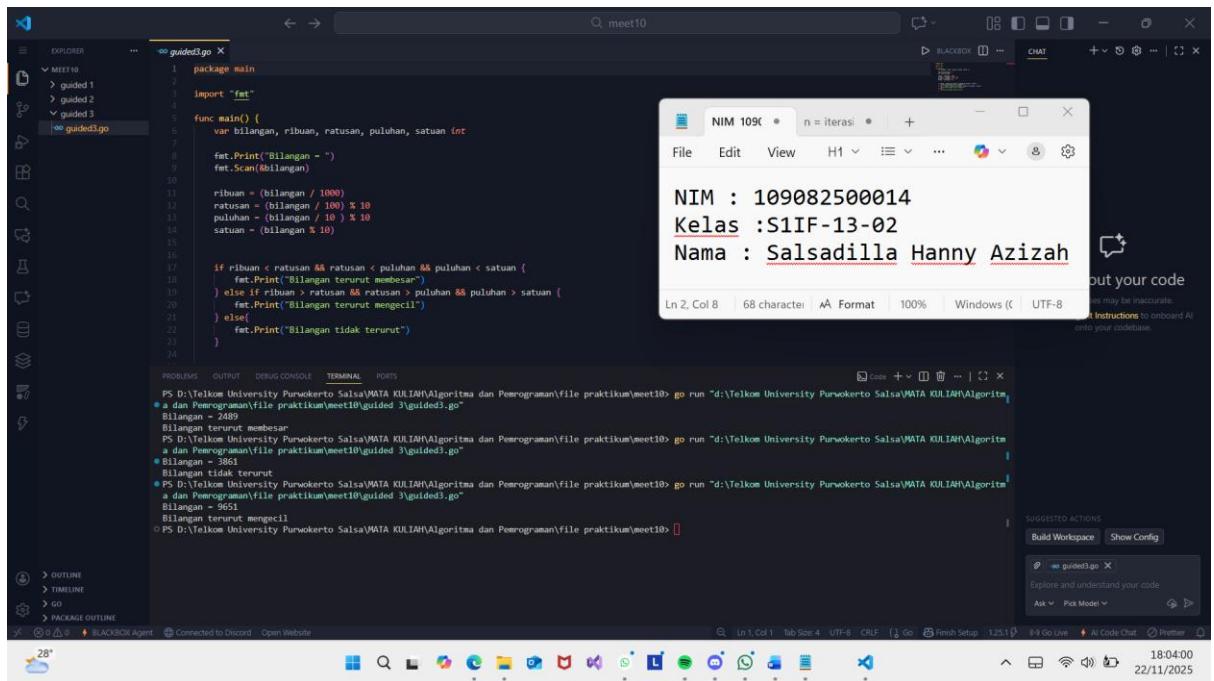
    if ribuan < ratusan && ratusan < puluhan && puluhan
    < satuan {
        fmt.Println("Bilangan terurut membesar")
    } else if ribuan > ratusan && ratusan > puluhan &&
    puluhan > satuan {
        fmt.Println("Bilangan terurut mengecil")
```

```

        } else{
            fmt.Println("Bilangan tidak terurut")
        }
    }
}

```

## Screenshot program



### Deskripsi program :

Program ini dibuat untuk mengecek apakah digit-digit pada sebuah bilangan empat angka tersusun membesar, mengecil, atau tidak berurutan sama sekali. Setelah pengguna memasukkan sebuah bilangan, program memisahkan digitnya menjadi ribuan, ratusan, puluhan, dan satuan menggunakan operasi pembagian dan modulus. Setelah setiap digit berhasil diambil, program mulai membandingkan. Kalau digit ribuan < ratusan < puluhan < satuan, maka bilangan tersebut dianggap terurut membesar. Kalau digitnya justru ribuan > ratusan > puluhan > satuan, maka program menampilkan bahwa bilangan terurut mengecil. Selain dua kondisi itu, sisanya dianggap tidak terurut. Dengan ini, program bisa menentukan pola susunan digit dari bilangan yang dimasukkan.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var berat, biaya, total int

    fmt.Println("Berat Parsel (Gram) = ")
    fmt.Scan(&berat)

    kg := berat / 1000
    sisa := berat % 1000

    biayaKg := kg * 10000

    if sisa >= 500 {
        biaya = sisa * 5
    } else {
        biaya = sisa * 15
    }

    if kg > 10 {
        total = biayaKg
    } else {
        total = biayaKg + biaya
    }

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

```

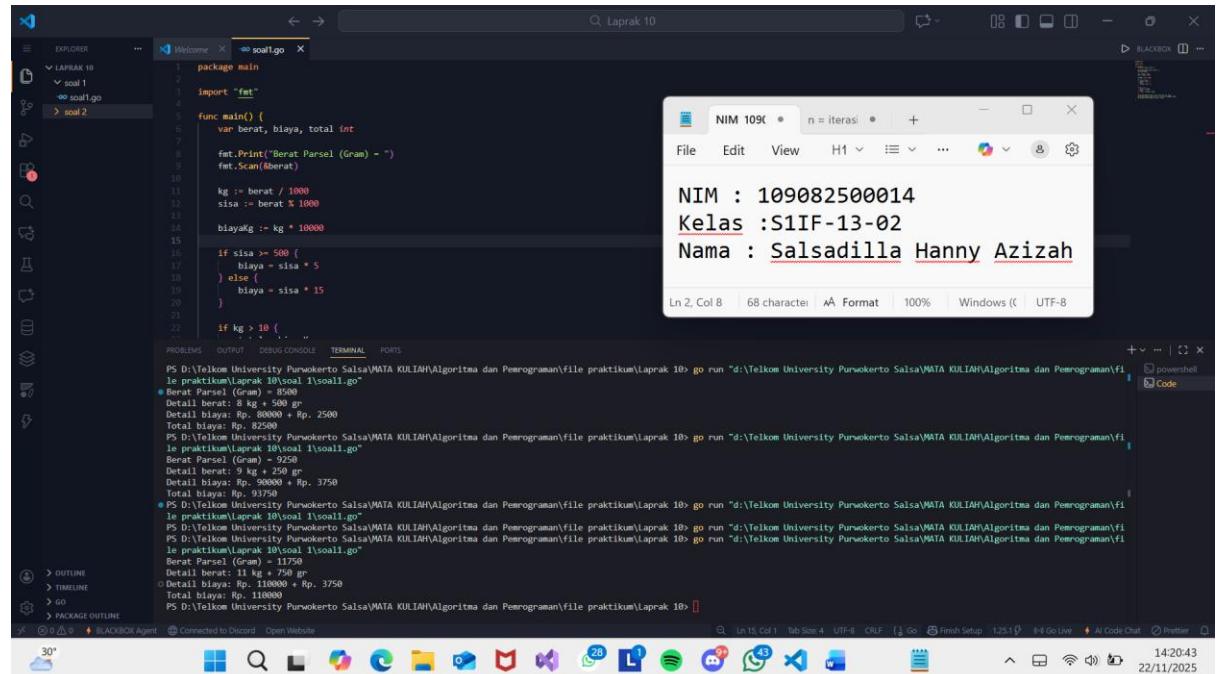
        fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n",
        biayaKg, biaya)

        fmt.Printf("Total biaya: Rp. %d\n", total)

    }

```

## Screenshot program



### Deskripsi program :

Pada program ini saya membuat perhitungan biaya pengiriman berdasarkan berat parsel. Variabel yang digunakan yaitu berat, kg, sisa, biayaKG, biayaSisa, dan totalBiaya. Variabel berat menyimpan input berat dalam gram. Setelah itu berat dipisahkan menjadi kg dan sisa gram. Perhitungan biayanya memakai tarif Rp10.000 per kg. Untuk sisa gram, jika sisanya 500 gram atau lebih maka dihitung Rp5 per gram, sedangkan kalau kurang dari 500 gram tarifnya Rp15 per gram. Namun kalau total berat lebih dari 10 kg, sisa gram tidak dikenakan biaya sama sekali. Program menampilkan berat awal, pembagian berat menjadi kg dan gram, biaya berdasarkan kg, biaya dari sisa gram, serta total biaya pengiriman. Program ini membantu menghitung biaya secara otomatis sesuai aturan yang berlaku.

## 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 if nam <= 40 {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}
```

## Screenshot program

The screenshot shows a Windows desktop environment. In the center, a Microsoft Edge browser window is open with the following content:

```
NIM : 109082500014  
Kelas : S1IF-13-02  
Nama : Salsadilla Hanny Azizah
```

Below the browser, the taskbar displays several pinned icons, including File Explorer, Microsoft Edge, FileZilla, and others. The system tray shows the date and time as 22/11/2025 and 15:46:33.

On the left, the Visual Studio Code interface is visible, showing a file named `soal2.go` with the following code:

```
package main  
import "fmt"  
func main() {  
    var nam float64  
    var nmk string  
    fmt.Println("Nilai akhir mata kuliah: ")  
    fmt.Scan(&nam)  
    if nam >= 90 {  
        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 tab in VS Code shows command-line output related to the execution of the Go program.

### Deskripsi program:

1. Jika nilai yang dimasukkan adalah 80.1, program tidak memberikan hasil yang benar. Variabel yang seharusnya menampilkan output tidak pernah diisi, dan logika programnya sudah salah sejak awal, sehingga tidak sesuai dengan spesifikasi soal.
2. Program memiliki beberapa kesalahan yang membuat hasilnya tidak sesuai. Kesalahan pertama ada pada variabel `nam`, yang seharusnya menyimpan angka tetapi justru diisi dengan teks di dalam kondisi. Hal ini membuat program tidak valid dan tidak bisa berjalan sebagaimana mestinya. Kesalahan kedua terletak pada penggunaan banyak `if` yang berdiri sendiri tanpa `else if`. Dengan struktur ini, sebuah nilai bisa masuk ke beberapa kondisi sekaligus, sehingga urutan pengecekannya tidak terkontrol. Selain itu, variabel `nmk` yang seharusnya menjadi tempat untuk menyimpan hasil akhir justru tidak pernah diisi, sehingga ketika dicetak hasilnya tetap kosong. Seharusnya alur program dimulai dengan membaca nilai angka yang dimasukkan pengguna, lalu mengeceknya satu per satu menggunakan `if – else if` supaya hanya satu kondisi yang dipilih. Setelah itu, hasil pengecekan disimpan ke variabel yang benar, dan baru ditampilkan ke layar. Dengan alur seperti ini, program akan bekerja sesuai aturan dan menghasilkan output yang benar.
3. Saya sudah perbaiki code programnya. Saya menginputkan 93.5; 70.6; dan 49.5. Keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main () {
    var b int

    fmt.Print("Bilangan = ")
    fmt.Scan(&b)

    fmt.Print("Faktor = ")

    for f := 1; f <= b; f++ {

        if b%f == 0 {
            fmt.Print(f, " ")
        } else {
        }
    }
}
```

```
package main

import "fmt"

func main () {
    var b int

    fmt.Print("Bilangan = ")
    fmt.Scan(&b)
```

```

fmt.Println("Faktor = ")
jumlahFaktor := 0

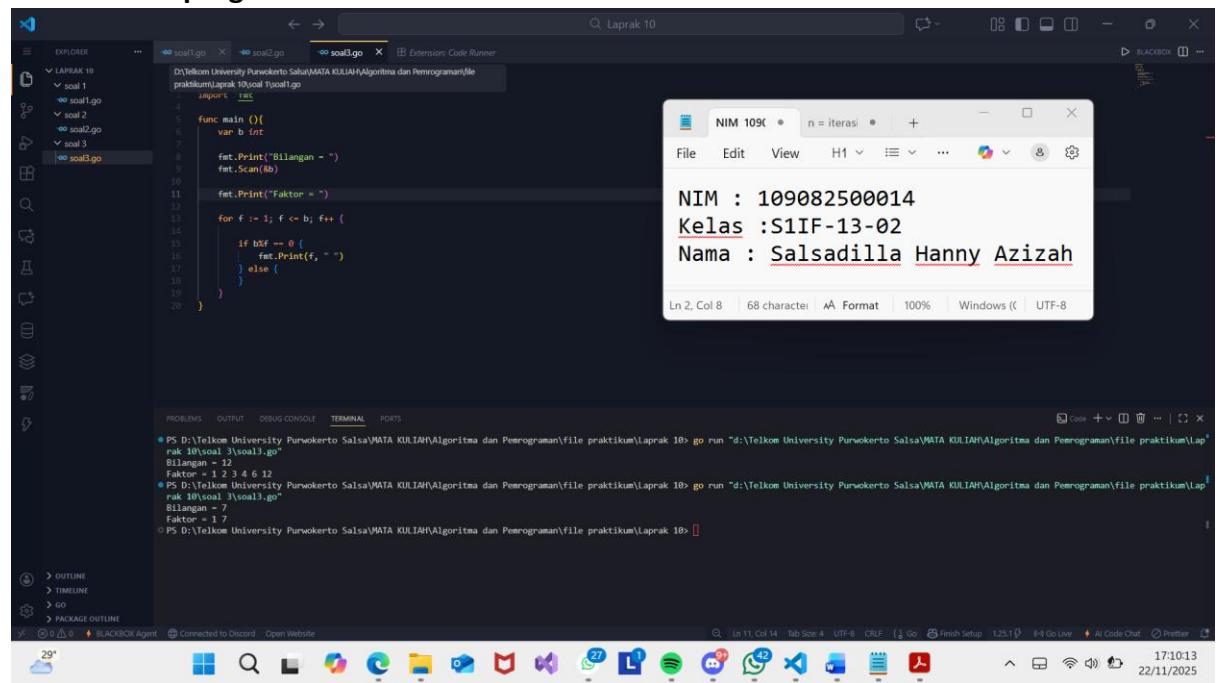
for f := 1; f <= b; f++ {

    if b%f == 0 {
        fmt.Print(f, " ")
        jumlahFaktor++
    } else {
    }
}
prima := jumlahFaktor == 2

fmt.Println("\nPrima:", prima)
}

```

## Screenshoot program



```

func main () {
    var b int
    fmt.Println("Bilangan = ")
    fmt.Scan(&b)

    fmt.Println("Faktor = ")
    jumlahFaktor := 0
    for f := 1; f <= b; f++ {
        if b%f == 0 {
            fmt.Println(f, " ")
            jumlahFaktor++
        } else {
        }
    }
    prima := jumlahFaktor == 2
    fmt.Println("\nPrima:", prima)
}

PS D:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10> go run "d:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10\soal 3\soal3.go"
● PS D:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10> go run "d:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10\soal 3\soal3.go"
Bilangan = 7
Faktor = 1 7
Prima: true
● PS D:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10> go run "d:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10\soal 3\soal3.go"
Bilangan = 12
Faktor = 1 2 3 4 6 12
Prima: false
● PS D:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10> go run "d:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10\soal 3\soal3.go"
Bilangan = 7
Faktor = 1 7
Prima: true
○ PS D:\Telkom University Purwokerto Salsa\MATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laprak 10>

```

### Deskripsi program :

Program ini dibuat untuk menerima sebuah bilangan bulat positif b dari pengguna. Setelah angka dimasukkan, program langsung mulai ngecek satu per satu angka dari 1 sampai b buat nyari mana saja yang termasuk faktor. Cara ngeceknya gampang: kalau b dibagi suatu angka dan hasilnya tidak ada sisa, berarti angka itu faktor, dan langsung ditampilkan pada output. Setelah semua faktor ketemu dan ditampilkan dalam satu baris, program lanjut untuk menghitung berapa banyak faktor yang ada. Jumlah faktor ini dipakai buat nentuin apakah b itu bilangan prima atau bukan. Logikanya juga sederhana: kalau faktornya cuma dua, yaitu 1 dan b sendiri, berarti b digolongkan sebagai bilangan prima. Tapi kalau faktornya lebih dari dua, program akan menyatakan bahwa angka tersebut bukan bilangan prima. Pada program ini tidak cuma menampilkan faktor-faktor dari sebuah bilangan, tapi juga sekaligus untuk mengetahui status “prima atau bukan” dari angka yang dimasukkan.