

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE-IF



Disusun oleh:

NADIFA AZKHIA SYARIF

109082530002

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 kk bool

    fmt.Print("Masukkan usia: ")
    fmt.Scan(&usia)

    fmt.Print("Apakah punya KK? (true/false): ")
    fmt.Scan(&kk)

    if usia >= 17 && kk {
        fmt.Println("bisa membuat KTP")
    } else {
        fmt.Println("belum bisa membuat KTP")
    }
}
```

Screenshoot program

The screenshot shows a Windows desktop with a Visual Studio Code window open. The workspace is named 'MINGGU10' and contains three files: 'soal2.go', 'soal3.go', and 'soal1.go'. The 'soal1.go' tab is active, displaying the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var usia int
7     var kk bool
8
9     fmt.Scan(&usia)
10    fmt.Println("Apakah punya KK? (true/false): ")
11    fmt.Scan(&kk)
12
13    if usia >= 17 && kk {
14        fmt.Println("bisa membuat KTP")
15    } else {
16        fmt.Println("belum bisa membuat KTP")
17    }
18}
19
20
```

To the right of the code editor, a terminal window shows the command-line interaction:

```
PS C:\Users\hp\OneDrive\Documents\minggu10\soal3> go run "c:\Users\hp\OneDrive\Documents\minggu10\soal3\soal1"
Masukkan usia: 17
Apakah punya KK? (true/false): true
bisa membuat KTP
PS C:\Users\hp\OneDrive\Documents\minggu10\soal3> cd "c:\Users\hp\OneDrive\Documents\minggu10\soal3"
PS C:\Users\hp\OneDrive\Documents\minggu10\soal3> go run "c:\Users\hp\OneDrive\Documents\minggu10\soal3\soal1"
Masukkan usia: 
```

A floating terminal window titled 'Nama Nad' shows the user input and output:

Name	NIM
Nadifa Azkha Syarif	109082530002

The status bar at the bottom of the VS Code window indicates the current file is 'soal1.go', the line is 21, column is 18, and the encoding is UTF-8.

Deskripsi program

Program ini meminta dua input dari pengguna, yaitu usia dalam bentuk angka dan status kepemilikan KK (Kartu Keluarga) dalam bentuk boolean true atau false. Setelah menerima input tersebut, program memeriksa apakah usia pengguna sudah 17 tahun atau lebih dan apakah pengguna memiliki KK. Jika kedua syarat itu terpenuhi, program akan menampilkan tulisan "*bisa membuat KTP*", tetapi jika salah satu syarat tidak terpenuhi, program menampilkan "*belum bisa membuat KTP*".

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var huruf string
    fmt.Scan(&huruf)

    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 if (huruf >= "a" && huruf <= "z") || (huruf >= "A"
&& huruf <= "Z") {

            fmt.Println("konsonan")

        } else {

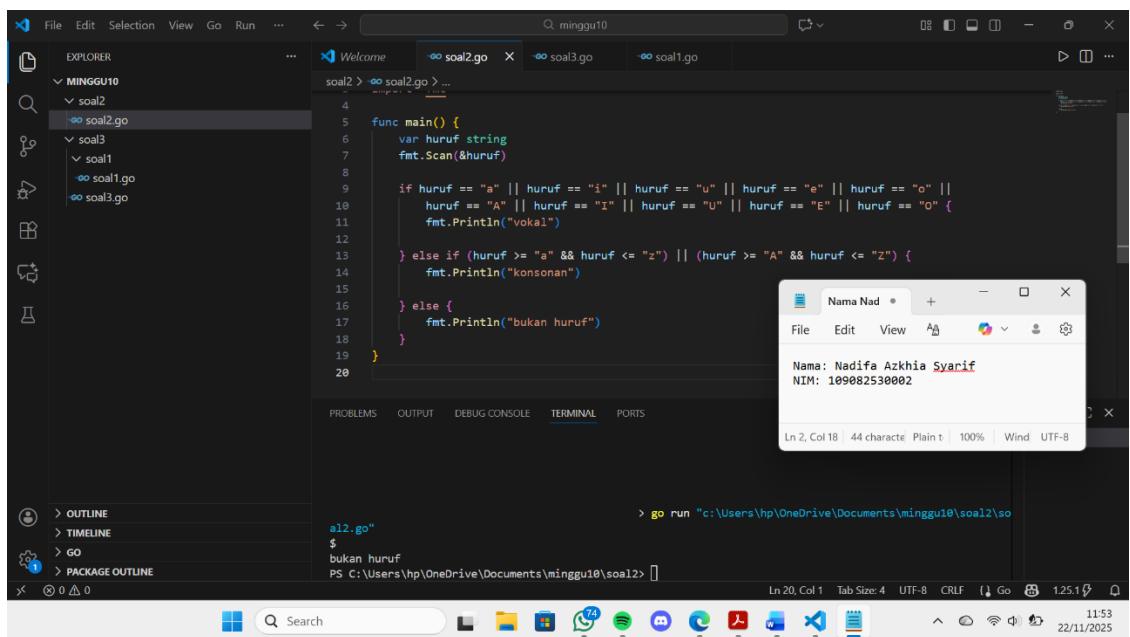
            fmt.Println("bukan huruf")

        }

    }

```

Screenshot program



Deskripsi program

Program ini berfungsi untuk menentukan apakah karakter yang dimasukkan termasuk huruf vokal, huruf konsonan, atau bukan huruf. Program membaca input berupa sebuah string yang seharusnya hanya berisi satu huruf. Setelah input diterima,

program memeriksa apakah karakter tersebut merupakan huruf vokal, yaitu a, i, u, e, o, baik dalam bentuk huruf kecil maupun huruf besar. Jika cocok dengan salah satu huruf tersebut, maka program menampilkan "vokal". Jika karakter bukan vokal, program kemudian mengecek apakah karakter berada dalam rentang huruf alfabet, yaitu dari 'a' sampai 'z' atau dari 'A' sampai 'Z'. Jika berada dalam rentang itu, maka karakter termasuk huruf konsonan dan program menampilkan "konsonan". Apabila karakter tidak memenuhi kedua kondisi sebelumnya, berarti input tersebut bukan huruf, misalnya berupa angka atau simbol. Dalam kondisi ini, program akan menampilkan "bukan huruf".

3. Guided 3

Source Code

```
package main

import (
    "fmt"
)

func main() {
    var n int
    fmt.Print("Bilangan: ")
    fmt.Scanln(&n)

    d1 := n / 1000
    d2 := (n / 100) % 10
    d3 := (n / 10) % 10
    d4 := n % 10

    if d1 < d2 && d2 < d3 && d3 < d4 {
```

```

        fmt.Printf("Digit pada bilangan %d terurut
membesar\n", n)

    } else if d1 > d2 && d2 > d3 && d3 > d4 {

        fmt.Printf("Digit pada bilangan %d terurut
mengecil\n", n)

    } else {

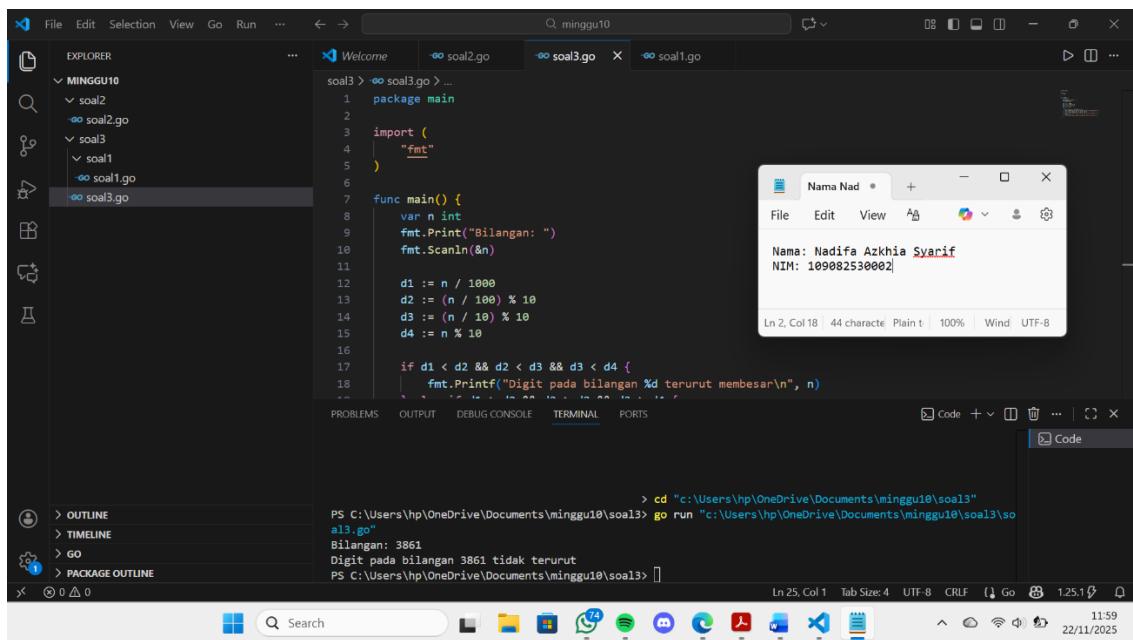
        fmt.Printf("Digit pada bilangan %d tidak
terurut\n", n)

    }

}

```

Screenshot program



Deskripsi program

Program ini digunakan untuk mengecek apakah digit-digit pada sebuah bilangan empat angka tersusun membesar, mengecil, atau tidak terurut. Bilangan empat digit dimasukkan ke dalam variabel `n`. Setelah itu, setiap digit dipisahkan menggunakan operasi pembagian dan sisa bagi. Digit ribuan disimpan pada `d1`, ratusan pada `d2`, puluhan pada `d3`, dan satuan pada `d4`.

Setelah setiap digit berhasil dipisahkan, program melakukan pengecekan berurutan. Kondisi pertama memeriksa apakah $d_1 < d_2 < d_3 < d_4$. Jika kondisi ini terpenuhi, berarti digit-digitnya tersusun membesar dan program menampilkan pesan bahwa digit bilangan tersebut terurut membesar. Kondisi kedua memeriksa apakah $d_1 > d_2 > d_3 > d_4$. Jika terpenuhi, digit-digitnya tersusun mengecil dan program menampilkan pesan terurut mengecil. Jika tidak memenuhi kedua kondisi tersebut, berarti digit-digit tidak membentuk pola urutan tertentu sehingga program menampilkan pesan tidak terurut.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

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

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

    biayaKg := kg * 10000
    var biayaSisa int

    if kg > 10 {
        biayaSisa = 0
    } else {
```

```

if sisa >= 500 {

    biayaSisa = sisa * 5

} else if sisa > 0 {

    biayaSisa = sisa * 15

} else {

    biayaSisa = 0

}

}

total := biayaKg + biayaSisa

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

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

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

}

```

Screenshot program

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

- File Explorer:** Shows a project structure under "MINGGU10" with files: soal2.go, soal3.go, soal1.go, tugas1.go, and soal3.go.
- Code Editor:** The main editor window displays the following Go code:

```

package main

import "fmt"

func main() {
    var berat int
    fmt.Println("Berat parcel (gram): ")
    fmt.Scan(&berat)

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

    biayakg := kg * 10000
    var biayaSisa int

    if kg > 10 {
        biayaSisa = 0
    } else {
        biayaSisa = ...
    }
}


```

- Terminal:** The terminal window shows the command `cd "c:\Users\hp\OneDrive\Documents\minggu10\soal3\tugas1> go run "c:\Users\hp\OneDrive\Documents\minggu10\soal3\tugas1\go"`. The output is:

```

Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750

```
- Modal Window:** A small modal window titled "Nama Nad" is open, displaying the text "Nama: Nadifa Azkhia Syarif" and "NIM: 109082530002".

Deskripsi program

Program ini menghitung biaya kirim berdasarkan berat parsel dalam gram. Berat dibagi menjadi dua bagian, yaitu kilogram (berat / 1000) dan sisa gram (berat % 1000). Biaya untuk kilogram dihitung dengan mengalikan kilogram dengan 10.000 rupiah. Setelah itu, biaya tambahan dihitung dari sisa gram. Jika kilogram lebih dari 10, sisa gram tidak dikenakan biaya. Jika tidak lebih dari 10, sisa gram dikenakan tarif 5 rupiah per gram bila sisa 500 gram atau lebih, dan tarif 15 rupiah per gram bila kurang dari 500 gram. Terakhir, biaya kilogram dan biaya tambahan dijumlahkan. Hasil akhirnya menampilkan detail berat, detail biaya, dan total biaya.

2. Tugas 2

a. Jika nam diberikan 80.1, apa keluarannya? Apakah sesuai spesifikasi?

Keluaran dari program adalah tidak ada nilai huruf (kosong), karena variabel nmk tidak pernah diberikan nilai. Program tidak sesuai spesifikasi karena nilai NAM seharusnya menghasilkan NMK, tetapi program tidak mengeluarkan hasil yang benar.

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

Kesalahan program:

1. Variabel nam (angka) diubah menjadi string seperti "A", "B", sehingga tipe data menjadi tidak konsisten.
2. Percabangan menggunakan banyak if terpisah, bukan else if, sehingga semua kondisi tetap diperiksa.
3. Variabel nmk tidak pernah diisi sehingga hasil akhirnya kosong.
4. Interval penilaian tidak mengikuti aturan tabel penilaian yang diberikan.

Alur program yang seharusnya:

- Membaca nilai NAM.
- Menentukan NMK berdasarkan interval dengan struktur if – else if – else.
- Menyimpan hasil penilaian pada variabel nmk.
- Menampilkan NMK tersebut.

c. Program yang sudah diperbaiki serta hasil uji 93.5, 70.6, dan 49.5

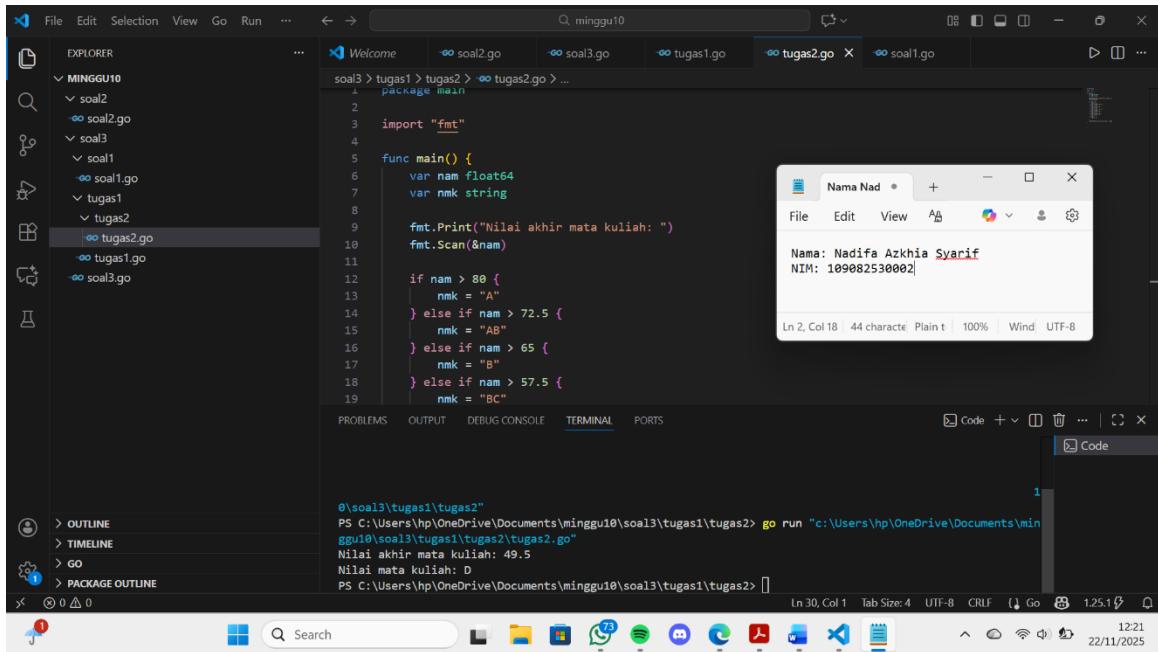
Source code

```
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



3. Tugas 3

Source code

```
package main

import "fmt"

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

    fmt.Print("Faktor: ")

    hitungFaktor := 0

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

```

        }

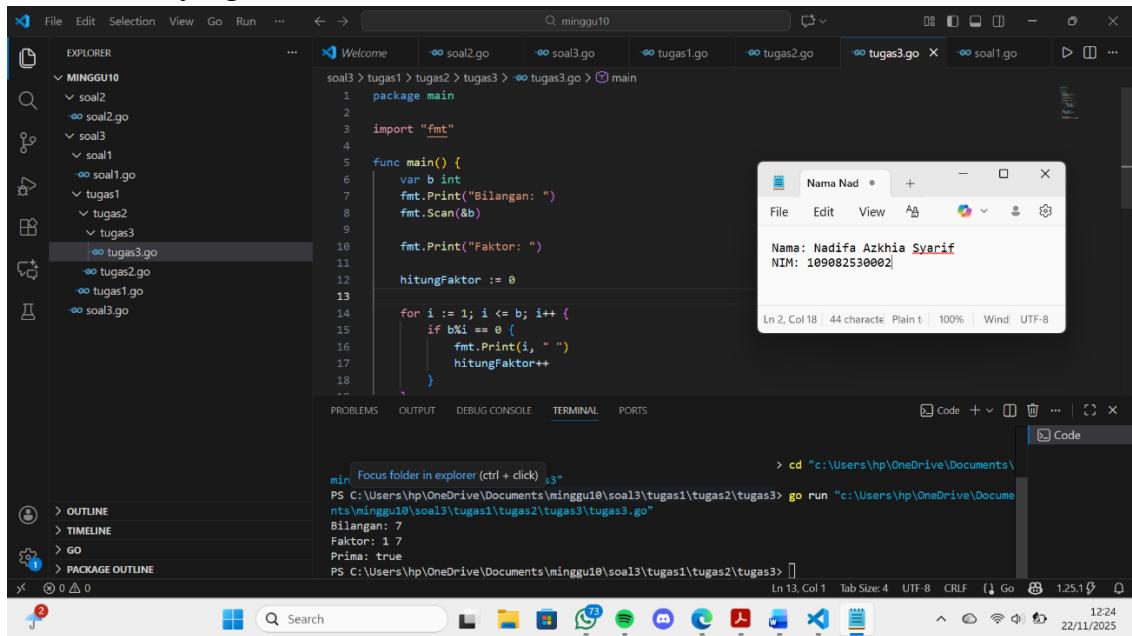
    }

    prima := false
    if hitungFaktor == 2 {
        prima = true
    }

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

```

Screenshot program



Deskripsi program

Program ini digunakan untuk menampilkan semua faktor dari sebuah bilangan bulat b (dengan $b > 1$), kemudian menentukan apakah bilangan tersebut merupakan bilangan prima. Program dimulai dengan membaca nilai b . Setelah itu, program melakukan perulangan dari 1 sampai b untuk memeriksa setiap nilai i . Jika b habis dibagi i (artinya $b \% i == 0$), maka i adalah faktor dari b dan angka tersebut ditampilkan. Program juga menghitung berapa banyak faktor yang ditemukan. Sebuah bilangan dikatakan prima apabila memiliki tepat dua faktor, yaitu 1 dan dirinya sendiri. Karena itu, setelah proses pencarian faktor selesai, program memeriksa apakah jumlah faktor sama dengan dua. Jika ya, nilai variabel prima

menjadi true; jika tidak, nilainya false. Terakhir, program menampilkan daftar faktor dan status apakah bilangan tersebut prima.