

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE-IF



Disusun oleh:

REZKY FARREL

109082500203

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

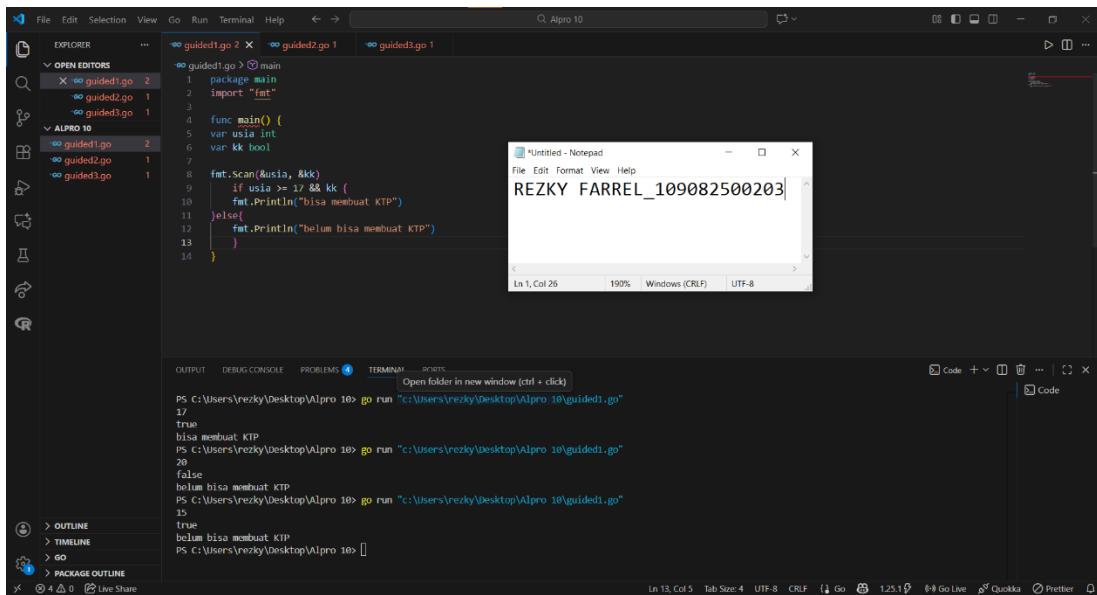
Source Code

```
package main
import "fmt"

func main() {
var usia int
var kk bool

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

Screenshot program :



Deskripsi program :

Program ini bertujuan memverifikasi syarat pembuatan KTP. Program ini bekerja dengan meminta input pengguna untuk dua variabel, yaitu usia (bertipe bilangan bulat) dan kk (bertipe boolean/benar-salah). Logika utamanya terletak pada pernyataan if yang menggunakan operator AND (&&), di mana program menuntut dua kondisi harus terpenuhi secara bersamaan agar hasilnya sukses: pengguna harus berusia minimal 17 tahun dan wajib memiliki Kartu Keluarga (bernilai true). Jika kedua syarat tersebut terpenuhi, program akan mencetak "bisa membuat KTP", atau sebaliknya

2. Guided 2

Source Code

```
package main

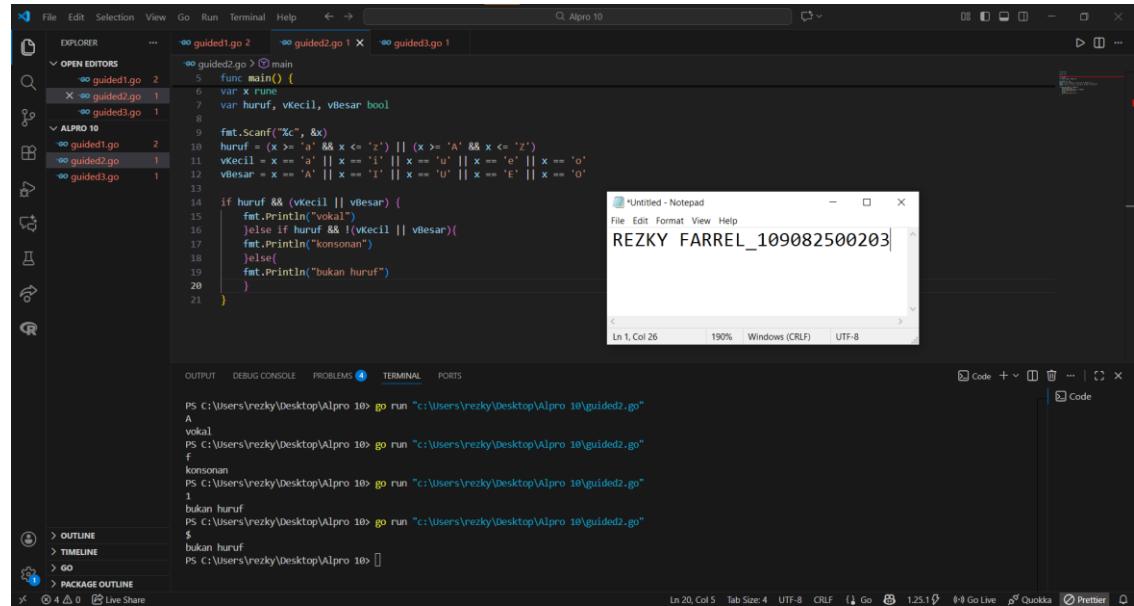
import "fmt"

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 :



Deskripsi program :

Program berfungsi sebagai alat klasifikasi karakter sederhana yang menentukan apakah input pengguna merupakan huruf vokal, konsonan, atau bukan huruf sama sekali. Prosesnya dimulai

dengan membaca satu karakter dari papan ketik, lalu program mengevaluasi karakter tersebut menggunakan variabel logika untuk memeriksa dua hal utama: apakah karakter tersebut termasuk abjad (huruf a-z atau A-Z) dan apakah karakter tersebut merupakan huruf vokal (a, i, u, e, o, baik huruf kecil maupun kapital).

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 :

The screenshot shows a Go development environment with several windows:

- Editor:** Displays three files: `guided1.go`, `guided2.go`, and `guided3.go`. The code in `guided3.go` is as follows:

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var n int
9     fmt.Println("Bilangan: ")
10    fmt.Scanln(&n)
11
12    d1 := n / 1000
13    d2 := (n / 100) % 10
14    d3 := (n / 10) % 10
15    d4 := n % 10
16
17    if d1 < d2 && d2 < d3 && d3 < d4 {
18        fmt.Printf("Digit pada bilangan %d terurut membesar\n", n)
19    } else if d1 > d2 && d2 > d3 && d3 > d4 {
20        fmt.Printf("Digit pada bilangan %d terurut mengecil\n", n)
21    } else {
22        fmt.Println("Digit pada bilangan %d tidak terurut\n", n)
23    }
24 }

```

- Terminal:** Shows command-line interactions with the program. It runs the program twice with inputs 2489 and 3861, printing the results: "Digit pada bilangan 2489 terurut membesar" and "Digit pada bilangan 3861 tidak terurut".
- Output:** Shows the status bar with information like line 6, column 1, tab size 4, and encoding UTF-8.
- File Explorer:** Shows the project structure with files `guided1.go`, `guided2.go`, and `guided3.go`.
- CodeLens:** Shows the number of changes (1) for each file.
- Quick Fix:** Shows a tooltip for the `fmt.Println` line in `guided3.go`.

Deskripsi program :

Program ini merupakan menganalisis pola urutan angka dalam sebuah bilangan empat digit. Prosesnya dimulai dengan memecah bilangan input menjadi empat bagian terpisah (ribuan, ratusan, puluhan, dan satuan) menggunakan operasi matematika pembagian dan modulus (sisa bagi). Setelah digit-digit tersebut dipisahkan, program melakukan perbandingan logika secara berurutan antar digit. Jika ditemukan bahwa digit pertama lebih kecil dari digit kedua dan seterusnya hingga akhir, program akan mengidentifikasinya sebagai bilangan yang "terurut membesar". Sebaliknya, jika digit pertama lebih besar dari digit berikutnya hingga akhir, maka dianggap "terurut mengecil". Apabila susunan angkanya tidak konsisten naik atau turun, program akan menyimpulkan bahwa bilangan tersebut "tidak terurut".

TUGAS

1. Tugas 1

Source code

```
package main
import "fmt"
func main() {
    var berat, kg, lebih, biayaLebih, biayaKg int
    fmt.Println("Berat parsel (gram): ")
    fmt.Scan(&berat)

    kg = berat / 1000
    lebih = berat % 1000

    biayaKg = kg * 10000
    biayaLebih = 0

    if berat > 10000 {
        biayaLebih = 0
    } else if lebih >= 500 {
        biayaLebih = lebih * 5
    } else if lebih > 0 {
        biayaLebih = lebih * 15
    } else {
        biayaLebih = 0
    }

    total := biayaKg + biayaLebih

    fmt.Println("Detail berat:", kg, "kg +", lebih, "gr")
    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.", biayaLebih)
    fmt.Println("Total biaya: Rp.", total)
}
```

Screenshot program :

The screenshot shows the Visual Studio Code (VS Code) interface. On the left, the Explorer sidebar displays a file tree with several Go files: guided1.go, guided2.go, guided3.go, soal1.go (selected), soal2.go, and soal3.go. Below the file tree, there's a section for 'ALPRO 10' containing the same files. The main editor area shows the content of soal1.go:

```

package main
import "fmt"
func main() {
    var berat, kg, lebih, biayalebih, biaya int
    fmt.Println("berat parcel (gram): ")
    fmt.Scan(&berat)
    kg = berat / 1000
    lebih = berat % 1000
    biayaKg = kg * 10000
    biayalebih = 0
    if berat > 10000 {
        biayalebih = 0
    } else if lebih >= 500 {
        biayalebih = lebih * 5
    } else if lebih > 0 {
        biayalebih = lebih * 15
    } else {
        biayalebih = 0
    }
}

```

The terminal at the bottom shows the output of running the program:

```

PS C:\Users\rezy\Desktop\Alpro 10> go run "c:/Users/rezy/Desktop/Alpro 10/soal1.go"
berat parcel (gram): 8500
Detail berat: 8 Kg + 500 gr
Detail Biaya: Rp. 80000 + Rp. 2500
Total Biaya: Rp. 82500
PS C:\Users\rezy\Desktop\Alpro 10>

```

Deskripsi program :

Program ini berfungsi sebagai kalkulator biaya pengiriman paket otomatis yang menghitung tarif berdasarkan total berat dalam satuan gram. Prosesnya dimulai dengan memisahkan berat input menjadi dua komponen: berat utama dalam kilogram (kg) dan sisa berat dalam gram, di mana setiap 1 kg dikenakan tarif dasar Rp 10.000. Logika unik program ini terletak pada perhitungan biaya untuk "sisa gram" atau kelebihan beratnya, yang ditentukan melalui serangkaian syarat bertingkat. Jika total berat paket melebihi 10 kg, maka biaya untuk sisa gram digratiskan sepenuhnya. Namun, jika beratnya di bawah 10 kg, program akan memeriksa jumlah sisa gram tersebut: apabila sisanya 500 gram atau lebih, tarifnya lebih murah (dikali Rp 5 per gram), sedangkan jika sisanya kurang dari 500 gram, tarifnya menjadi lebih mahal (dikali Rp 15 per gram).

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 Go development environment. In the main window, several files are listed in the Explorer sidebar: guided1.go, guided2.go, guided3.go, soal1.go, soal2.go, soal3.go, and ALPRO 10. The file soal2.go is open in the editor, displaying the following Go code:

```
package main
import "fmt"
func main() {
    var nam float64
    var nmk string
    fmt.Println("Nilai akhir mata kuliah: ")
    nm.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"
    }
}
```

The terminal window below shows the execution of the program:

```
Nilai akhir mata kuliah: 70,6
Nilai mata kuliah: A
PS C:\Users\rezky\Desktop\Alpro 10> go run "c:\Users\rezky\Desktop\Alpro 10\soal2.go"
PS C:\Users\rezky\Desktop\Alpro 10> go run "c:\Users\rezky\Desktop\Alpro 10\soal2.go"
Nilai akhir mata kuliah: 70,6
Nilai mata kuliah: B
PS C:\Users\rezky\Desktop\Alpro 10> go run "c:\Users\rezky\Desktop\Alpro 10\soal2.go"
Nilai akhir mata kuliah: 49,5
Nilai mata kuliah: D
PS C:\Users\rezky\Desktop\Alpro 10>
```

Deskripsi program

Program ini dirancang sebagai sistem konversi nilai akademik yang bertugas menerjemahkan nilai angka (kuantitatif) menjadi nilai huruf (kualitatif) berdasarkan standar kelulusan tertentu. Prosesnya diawali dengan meminta pengguna memasukkan nilai akhir mata kuliah yang dapat berupa bilangan desimal, berkat penggunaan tipe data float64. Nilai tersebut kemudian dievaluasi melalui serangkaian gerbang logika if-else yang disusun secara hierarkis dari nilai tertinggi ke terendah. Program akan memeriksa apakah nilai tersebut memenuhi ambang batas tertentu—mulai dari nilai 80 ke atas untuk grade "A", 72.5 ke atas untuk "AB", hingga batas-batas di bawahnya seperti 65 untuk "B" dan 40 untuk "D". Begitu nilai input cocok dengan salah satu kriteria rentang tersebut, program akan menetapkan huruf mutu yang sesuai dan menampilkannya sebagai hasil akhir evaluasi.

Tugas 3

Source code

```
package main

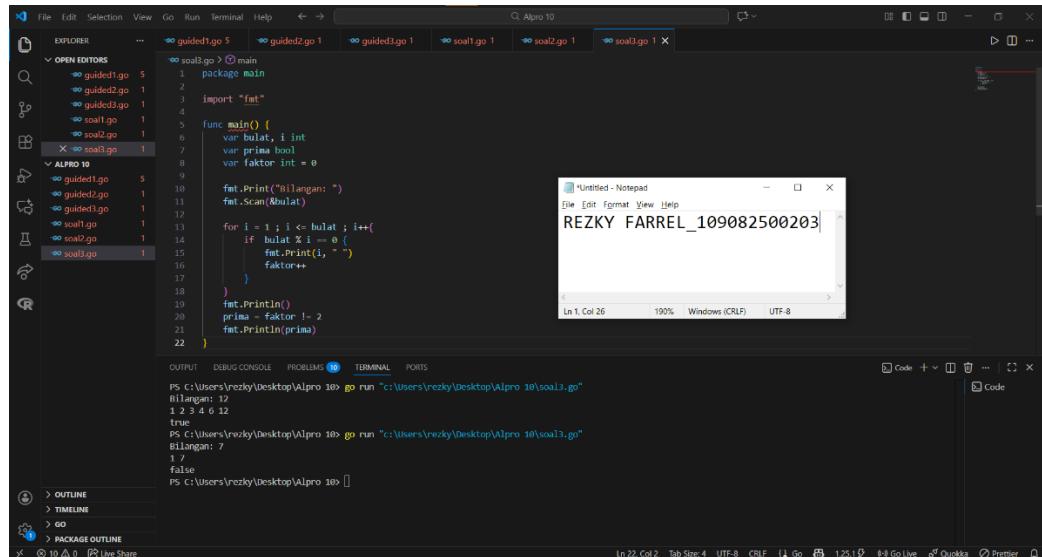
import "fmt"

func main() {
    var bulat, i int
    var prima bool
    var faktor int = 0

    fmt.Println("Bilangan: ")
    fmt.Scan(&bulat)

    for i = 1 ; i <= bulat ; i++{
        if bulat % i == 0 {
            fmt.Print(i, " ")
            faktor++
        }
    }
    fmt.Println()
    prima = faktor != 2
    fmt.Println(prima)
}
```

Screenshot program :



Deskripsi program

Program ini bertujuan sebagai pencari faktor bilangan sekaligus penentu karakteristik bilangan tersebut dengan menggunakan logika perbandingan yang cukup unik. Setelah pengguna memasukkan sebuah bilangan bulat, program melakukan perulangan untuk menguji satu per satu angka dari 1 hingga bilangan

tersebut; setiap angka yang terbukti bisa membagi habis input akan langsung ditampilkan di layar dan dihitung ke dalam variabel penghitung faktor. Langkah terakhir adalah penentuan status boolean, di mana program menetapkan hasil true apabila jumlah faktor yang ditemukan tidak sama dengan dua. Hal ini menghasilkan output yang berkebalikan dari ekspektasi nama variabelnya: bilangan prima (yang secara definisi pasti memiliki tepat dua faktor) justru akan menghasilkan output false, sedangkan bilangan selain prima (seperti bilangan komposit yang memiliki banyak faktor) akan menghasilkan output true.