

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II  
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**Liya Khoirunnisa / 2311102124**

**IF-11-05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Golang adalah bahasa pemrograman baru yang dikembangkan Google oleh Robert Griesemer, Rob Pike, dan Ken Thompson pada tahun 2007 dan diperkenalkan ke publik pada tahun 2009. Gaya sintaks bahasa Golang dengan bahasa C dan C++ mirip.

Struktur program dalam bahasa pemrograman Go diantaranya :

- a. Package main  
Setiap program harus memiliki *package*. Pada setiap proyek wajib ada satu file bernama *main*. File yang berpackage *main*, akan dieksekusi pertama kali ketika program dijalankan
- b. Fungsi main  
Dalam sebuah proyek wajib ada file program yang berisikan fungsi yang bernama *main*. Fungsi tersebut harus berada di package yang sama dengan nama *main*.
- c. Import  
Digunakan untuk menggabung package lain ke dalam file program, agar isi package bisa digunakan. Salah satu package yang disediakan oleh Golang yaitu *fmt*.

Golang menyediakan command *go* untuk keperluan pengembangan aplikasi, diantaranya :

- a. Go run  
Digunakan untuk mengeksekusi file program. Command ini hanya bisa digunakan pada file yang menggunakan *package main*.
- b. Go test  
Digunakan untuk keperluan testing.
- c. Go build  
Digunakan untuk mengkompilasi file program yang menghasilkan executable pada folder yang aktif.
- d. Go install  
Go install memiliki fungsi yang sama dengan go build, namun setelah proses kompilasi selesai, dilanjutkan ke proses instalasi program.
- e. Go get  
Digunakan untuk mendownload package.

## II. GUIDED

1. Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berturut-turut adalah 'merah', 'kuning', 'hijau', dan 'ungu' selama 5 kali percobaan berulang.

Buatlah sebuah program yang menerima Input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan true apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan false untuk Telkom Um urutan warna lainnya.

### Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    // Urutan warna yang benar
    correctOrder := []string{"merah", "kuning", "hijau", "ungu"}

    // Membaca input untuk 5 percobaan
    reader := bufio.NewReader(os.Stdin)
    success := true

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)

        // Membaca input dari pengguna
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

        // Memisahkan input berdasarkan spasi
        colors := strings.Split(input, " ")

        // Mengecek apakah urutan warna sesuai
        for j := 0; j < 4; j++ {
            if colors[j] != correctOrder[j] {
                success = false
                break
            }
        }
    }
}
```

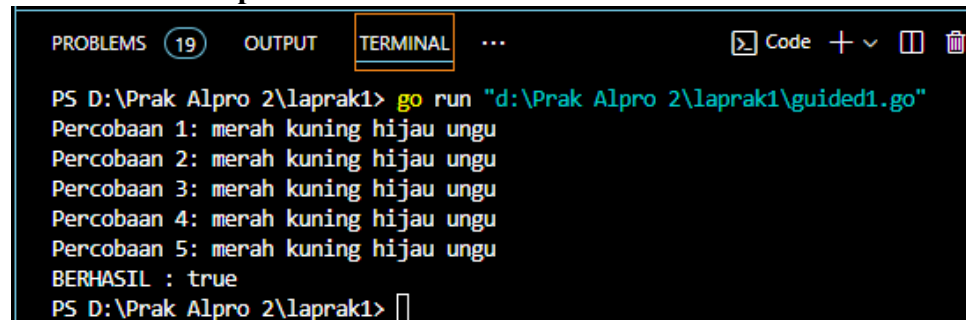
```

        // Jika ada percobaan yang tidak sesuai, keluar
dari loop
        if !success {
            break
        }
    }

    // Menampilkan hasil
    if success {
        fmt.Println("BERHASIL : true")
    } else {
        fmt.Println("BERHASIL : false")
    }
}

```

### Screenshoot Output



```

PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\guided1.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL : true
PS D:\Prak Alpro 2\laprak1>

```

### Deskripsi Program

Program di atas dibuat untuk memverifikasi apakah pengguna memasukkan urutan warna yang benar atau tidak sebanyak 5 kali percobaan. Pengguna diminta untuk menginputkan urutan warna (merah, kuning, hijau, ungu) sesuai dengan urutan yang benar. Jika semua percobaan benar, maka mencetak “true”. Namun jika ada satu kesalahan, program mencetak “false”.

**\*Note:** Pada terminal terdapat 19 problem dikarenakan package dan func main digunakan juga di file lain.

### 2. Program penjumlahan

#### Sourcecode

```

package main

import "fmt"

func main(){

```

```

var a, b, c, d, e int
var hasil int
fmt.Scanln(&a, &b, &c, &d, &e)

hasil = a+b+c+d+e
fmt.Println("Hasil Penjumlahan ", a,b,c,d,e, "adalah = ", hasil)
}

```

### Screenshoot Output

```

PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\guided2.go"
1 2 3 4 5
Hasil Penjumlahan 1 2 3 4 5 adalah = 15
PS D:\Prak Alpro 2\laprak1>

```

### Deskripsi Program

Program di atas dibuat untuk menghitung penjumlahan lima bilangan bulat yang diinputkan pengguna. Setelah input diterima, program akan menjumlahkan lima bilangan yang telah diinputkan dan menampilkan hasilnya. Tipe data yang digunakan yaitu int.

**\*Note:** Pada terminal terdapat 16 problem dikarenakan package dan func main digunakan juga di file lain.

3. Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar nilai mata kuliah (NMK) sebagai berikut:

NAM	NMK
NAM > 80	A
72.5 < NAM <= 80	AB
65 < NAM <= 72.5	B
57.5 < NAM <= 65	BC
50 < NAM <= 57.7	C
40 < NAM <= 50	D
NAM <= 40	E

Jawablah pertanyaan-pertanyaan berikut:

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

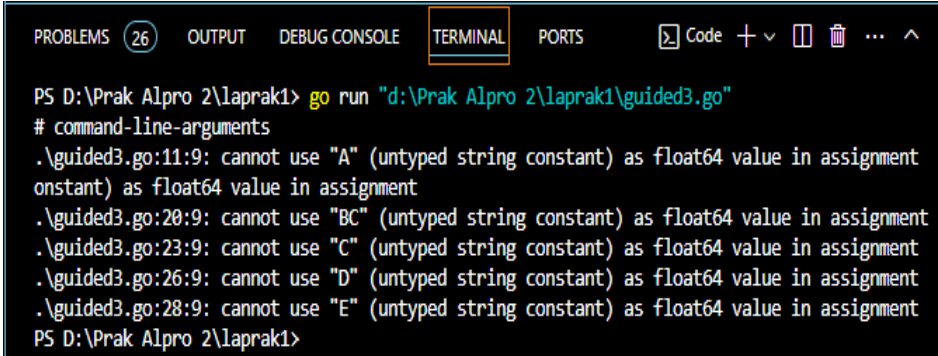
### Source code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&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)
}
```

### Screenshoot



```
PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\guided3.go"
# command-line-arguments
.\guided3.go:11:9: cannot use "A" (untyped string constant) as float64 value in assignment
.\guided3.go:20:9: cannot use "BC" (untyped string constant) as float64 value in assignment
.\guided3.go:23:9: cannot use "C" (untyped string constant) as float64 value in assignment
.\guided3.go:26:9: cannot use "D" (untyped string constant) as float64 value in assignment
.\guided3.go:28:9: cannot use "E" (untyped string constant) as float64 value in assignment
PS D:\Prak Alpro 2\laprak1>
```

## Deskripsi

Program di atas tidak dapat berjalan dengan baik karena kesalahan code. Sehingga tidak dapat menginputkan nam 80.1.

**\*Note:** Pada terminal terdapat 26 problem dikarenakan package dan func main digunakan juga di file lain.

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

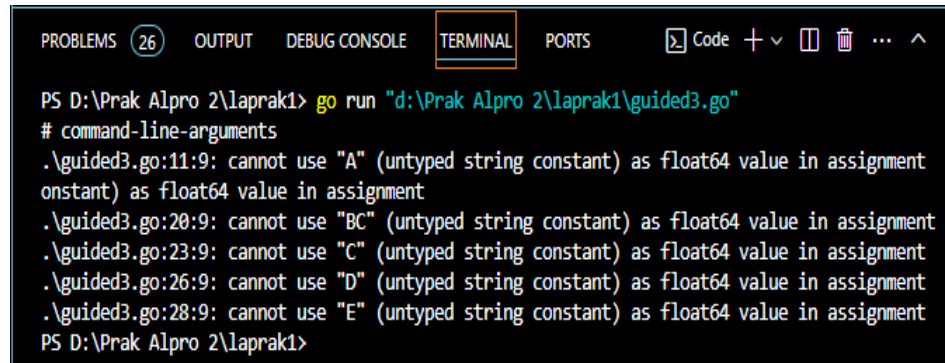
## Source code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&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)
}
```

## Screenshoot

A screenshot of a Go IDE's terminal window. The terminal shows the command 'go run "d:\Prak Alpro 2\laprak1\guided3.go"' being executed. Below the command, several error messages are displayed, all stating 'cannot use [string value] (untyped string constant) as float64 value in assignment'. The errors are for variables 'A', 'BC', 'C', 'D', and 'E' at lines 11, 20, 23, 26, and 28 respectively. The terminal window has tabs for 'PROBLEMS' (26), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (selected), and 'PORTS'.

```
PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\guided3.go"
# command-line-arguments
.\guided3.go:11:9: cannot use "A" (untyped string constant) as float64 value in assignment
onstant) as float64 value in assignment
.\guided3.go:20:9: cannot use "BC" (untyped string constant) as float64 value in assignment
.\guided3.go:23:9: cannot use "C" (untyped string constant) as float64 value in assignment
.\guided3.go:26:9: cannot use "D" (untyped string constant) as float64 value in assignment
.\guided3.go:28:9: cannot use "E" (untyped string constant) as float64 value in assignment
PS D:\Prak Alpro 2\laprak1>
```

### Deskripsi

Kesalahan pada program di atas yaitu pada penggunaan variabel nam bertipe data float64 yang seharusnya untuk menyimpan nilai angka (nilai akhir mata kuliah), justru ditetapkan sebagai nilai huruf A, AB, B, BC, C, D, E. Hal ini menyebabkan tidak kesesuaian penggunaan tipe data.

**\*Note:** Pada terminal terdapat 26 problem dikarenakan package dan func main digunakan juga di file lain.

- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

### Source code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string


    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nam)

    // Menggunakan else if untuk memastikan hanya satu
    nilai yang diassign
    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 Output



```
PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - []
```

```
PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\guided3.go"  
Nilai akhir mata kuliah: 93.5  
Nilai mata kuliah: A  
PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\guided3.go"  
Nilai akhir mata kuliah: 70.6  
Nilai mata kuliah: B  
PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\guided3.go"  
Nilai akhir mata kuliah: 49.5  
Nilai mata kuliah: D  
PS D:\Prak Alpro 2\laprak1> 
```

### Deskripsi Program

Program di atas dibuat untuk mengkonversi nilai numerik ke nilai huruf. Pengguna diminta untuk memasukkan nilai numerik. Lalu program akan membandingkan dengan kriteria yang ada. Setelah itu nilai huruf dicetak ke layar. Perbaikan yang dilakukan yaitu dengan mengubah variabel yang menyimpan nilai huruf dengan nmk dan mengubah if menjadi `else if` agar lebih efektif.

**\*Note:** Pada terminal terdapat 19 problem dikarenakan package dan func main digunakan juga di file lain. Dan maaf kak, codingan `guided3` saya ubah karena ada ketidaksesuaian nilai huruf dengan di modul.

### III. UNGUIDED

1. Suatu pita (string) berisi kumpulan nama-nama bunga yang dipisahkan oleh spasi dan '-', contoh pita diilustrasikan seperti berikut ini.

Pita: mawar-melati-tulip-teratai-kamboja-anggrek

Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak nol) N, kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita.

(Petunjuk: gunakan operasi penggabungan string dengan operator "+").

Tampilkan isi pita setelah proses input selesai.

Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetikkan 'SELESAI'. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita

#### Source code

```
/*Liya Khoirunnisa - 2311102124*/

package main

import (
    "bufio" // Membaca input dari pengguna
    "fmt"    // Menampilkan teks ke layar
    "os"     // Mendapatkan input dari terminal
    "strings" // Mengolah teks
)

func main() {
    // Membaca input
    reader := bufio.NewReader(os.Stdin)

    // Deklarasi pita untuk menyimpan daftar bunga
    var pita []string
    nomorBunga := 1

    // Perulangan inputan bunga
    for {
        fmt.Printf("Bunga %d: ", nomorBunga)
        bunga, _ := reader.ReadString('\n')
        bunga = strings.TrimSpace(bunga)

        // Perulangan akan berhenti jika input "selesai"
        if strings.ToLower(bunga) == "selesai" {
            break
        }

        // Jika tidak maka disimpan di pita
        pita = append(pita, bunga)
        nomorBunga++
    }
}
```

```
// Menampilkan hasil
if len(pita) > 0 {
    fmt.Printf("Pita: %s -\n", strings.Join(pita, " -
"))
    fmt.Printf("Bunga: %d\n", len(pita))
} else {
    fmt.Println("Pita: ")
    fmt.Println("Bunga: 0")
}
}
```

### Screenshoot Output

```
PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\unguided1.go"
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Bunga 4: SELESAI
Pita: Kertas - Mawar - Tulip -
Bunga: 3
PS D:\Prak Alpro 2\laprak1>
```

### Deskripsi Program

Program di atas dibuat untuk mengumpulkan daftar nama bunga dari inputan pengguna sampai pengguna mengetikkan kata “selesai”. Algoritma yang digunakan adalah pengulangan untuk menerima input dari pengguna dan menyimpannya dalam “pita”. Setelah itu, program menampilkan semua bunga yang telah dicatat beserta dengan jumlahnya. Tipe data yang digunakan yaitu string dan int.

**\*Note:** Pada terminal terdapat 19 problem dikarenakan package dan func main digunakan juga di file lain.

- Setiap hari Pak Andi membawa banyak barang belanjaan dari pasar dengan mengendarai sepeda motor. Barang belanjaan tersebut dibawa dalam kantong terpal di kiri-kanan motor. Sepeda motor tidak akan oleng jika selisih berat barang di kedua kantong sisi tidak lebih dari 9 kg. Buatlah program Pak Andi yang menerima Input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebut hingga salah satu kantong terpal berisi 9 kg atau lebih. Pada modifikasi program tersebut, program akan menampilkan true jika selisih kedua isi kantong lebih dari atau sama dengan 9 kg. Program berhenti

memproses apabila total berat isi kedua kantong melebihi 150 kg atau salah satu kantong beratnya negatif.

### Source code

```
/*Liya Khoirunnisa - 2311102124*/

package main

import "fmt" // Berfungsi untuk menampilkan teks ke layar

func main() {
    var kantong1, kantong2 float64

    for {
        // Input berat belanjaan dari kedua kantong
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&kantong1, &kantong2)

        // Cek jika salah satu kantong beratnya negatif atau total berat > 150
        if kantong1 < 0 || kantong2 < 0 || (kantong1+kantong2) > 150 {
            fmt.Println("Proses selesai")
            break
        }

        // Hitung apakah sepeda motor oleng atau tidak
        if kantong1 > kantong2 {
            if kantong1-kantong2 >= 9 {
                fmt.Println("Sepeda motor pak Andi akan oleng: true")
            } else {
                fmt.Println("Sepeda motor pak Andi akan oleng: false")
            }
        } else {
            if kantong2-kantong1 >= 9 {
                fmt.Println("Sepeda motor pak Andi akan oleng: true")
            } else {
                fmt.Println("Sepeda motor pak Andi akan oleng: false")
            }
        }
    }
}
```

### Screenshoot Output

```
PROBLEMS (19) OUTPUT TERMINAL ... Code + - [ ] [X]

PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\unguided2.go"
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai
PS D:\Prak Alpro 2\laprak1> [ ]
```

### Deskripsi Program

Program di atas dibuat untuk memeriksa apakah sepeda motor Pak Andi akan oleng jika belanjaan diletakkan di dua kantong. Program meminta input dari berat kedua kantong dalam bentuk gram, lalu dicek apakah berat salah satu negatif atau total berat melebihi 150 Kg. Jika melebihi 150 kg maka program akan berhenti. Jika berat valid, program akan mengecek keseimbangan dengan membandingkan selisih berat kedua kantong. Jika selisihnya lebih dari atau sama dengan 9 Kg, maka sepeda motor akan oleng.

**\*Note:** Pada terminal terdapat 19 problem dikarenakan package dan func main digunakan juga di file lain.

3. 
$$f(k) = \frac{(4k+2)^2}{(4k+1)(4k+3)}$$

Buatlah sebuah program yang menerima input sebuah bilangan sebagai K, kemudian menghitung dan menampilkan nilai f(K) sesuai persamaan di atas

### Sourcecode

```
/*Liya Khoirunnisa - 2311102124*/

package main

import (
    "fmt" // Untuk input atau output
    "math" // Untuk operasi matematika
)

// Fungsi untuk menghitung f(k)
func f(k float64) float64 {
    pembilang := math.Pow(4*k+2, 2)
    penyebut := (4*k + 1) * (4*k + 3)
    return pembilang / penyebut
}
```

```

}

func main() {
    var k float64
    fmt.Print("Nilai K = ") // Mencetak ke layar untuk
    meminta input
    fmt.Scanln(&k) // Membaca input
    hasil := f(k) // Memanggil fungsi
    fmt.Printf("Nilai f(K) = %.10f\n", hasil) // Mencetak
    nilai
}

```

### Screenshoot Output

```

PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\unguided3.go"
Nilai K = 100
Nilai f(K) = 1.0000061880
PS D:\Prak Alpro 2\laprak1>

```

### Deskripsi Program

Program di atas dibuat untuk menghitung nilai fungsi  $f(K)$  berdasarkan input  $K$  dari pengguna, dengan rumus yang ada. Setelah pengguna memasukkan  $K$ , program akan menghitung berdasarkan rumus yang ada dan akan mencetak hasilnya dengan 10 angka di belakang koma. Program ini menggunakan import *fmt* dan *math*.

**\*Note:** Pada terminal terdapat 19 problem dikarenakan package dan func main digunakan juga di file lain.

- PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program Biaya Pos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut!

Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

### Sourcecode

```

/*Liya Khoirunnisa - 2311102124*/

package main

import (
    "fmt" // Untuk input atau output

```

```

)

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

    // Menghitung berat total dalam kg dan gram
    beratKg := beratGram / 1000
    sisaGram := beratGram % 1000

    // Menghitung biaya kirim
    biayaPerKg := 10000
    biayaSisa := 0

    // Cek
    if beratKg > 10 {
        biayaSisa = 0
    } else if sisaGram >= 500 {
        biayaSisa = sisaGram * 5
    } else {
        biayaSisa = sisaGram * 15
    }

    // Menghitung total biaya kirim
    totalBiaya := (beratKg * biayaPerKg) + biayaSisa

    // Output detail berat dan biaya
    fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg,
sisaGram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n",
beratKg*biayaPerKg, biayaSisa)
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}

```

### Screenshoot Output

```

PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\unguided4.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\Prak Alpro 2\laprak1> 

```

### Deskripsi Program

Program di atas dibuat untuk menghitung biaya kirim parsel berdasarkan beratnya. Pengguna diminta menginputkan berat dalam gram, lalu program menghitung berat dalam kg dan sisa gram. Biaya per kg adalah Rp 10.000, dan sisa gram dihitung berdasarkan kondisi tertentu. Program kemudian

menghitung total biaya kirim dan menampilkan rincian berat dan biaya kepada pengguna.

**\*Note:** Pada terminal terdapat 19 problem dikarenakan package dan func main digunakan juga di file lain.

5. Sebuah bilangan bulat  $b$  memiliki faktor bilangan  $f > 0$  jika  $f$  habis membagi  $b$ . Contoh: 2 merupakan faktor dari bilangan 6 karena 6 habis dibagi 2. Buatlah program yang menerima input sebuah bilangan bulat  $b$  dan  $b > 1$ . Program harus dapat mencari dan menampilkan semua faktor dari bilangan tersebut! Bilangan bulat  $b > 0$  merupakan bilangan prima  $p$  jika dan hanya jika memiliki persis dua faktor bilangan saja, yaitu 1 dan dirinya sendiri. Lanjutkan program sebelumnya. Setelah menerima masukan sebuah bilangan bulat  $b > 0$ . Program tersebut mencari dan menampilkan semua faktor bilangan tersebut. Kemudian, program menentukan apakah  $b$  merupakan bilangan prima.

### Sourcecode

```
/*Liya Khoirunnisa - 2311102124*/

package main

import (
    "fmt"
)

func main() {
    // Deklarasi variabel
    var bilangan int
    fmt.Print("Bilangan: ") // Meminta input pengguna
    fmt.Scan(&bilangan) // Membaca input pengguna

    // Memeriksa bilangan
    if bilangan <= 1 {
        fmt.Println("Bilangan harus lebih besar dari 1.")
        return // Menghentikan program
    }

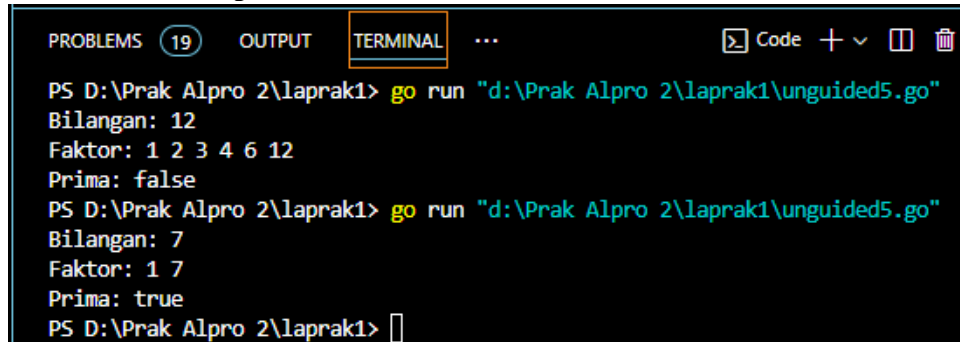
    fmt.Print("Faktor: ")
    jumlahFaktor := 0 // Variabel jumlah faktor
    for i := 1; i <= bilangan; i++ {
        if bilangan%i == 0 {
            fmt.Printf("%d ", i) // Menampilkan faktor
            jumlahFaktor++ // Menambah jumlah faktor
        }
    }
    fmt.Println()

    // Mengecek bilangan prima
```



```
    prima := jumlahFaktor == 2
    fmt.Printf("Prima: %v\n", prima)
}
```

### Screenshoot Output



```
PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\unguided5.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Prak Alpro 2\laprak1> go run "d:\Prak Alpro 2\laprak1\unguided5.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Prak Alpro 2\laprak1> 
```

### Deskripsi Program

Program di atas dibuat untuk menentukan faktor dan mengecek bilangan prima. Pengguna diminta menginputkan bilangan, lalu program akan menentukan faktor bilangan tersebut dan menentukan apakah bilangan tersebut bilangan prima atau bukan.

**\*Note:** Pada terminal terdapat 19 problem dikarenakan package dan func main digunakan juga di file lain.

## DAFTAR PUSTAKA

- [1] Agung, N. (2017). *Dasar Pemrograman Golang*. Dinkes Provinsi Jatim. Tersedia dari <https://dinkes.jatimprov.go.id/userfile/dokumen/Dasar%20Pemrograman%20Golang.pdf>