

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II  
RIVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**SHAFa ADILA SANTOSO / 2311102158**

**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

### Struktur Program Go

Golang adalah bahasa pemrograman sumber terbuka (opensource) yang berguna untuk memudahkan pembuatan perangkat lunak yang sederhana, andal, dan efisien. Dalam bahasa pemrograman Go, terdapat dua buah komponen utama yang selalu ada dalam kerangka bahasa pemrograman, diantaranya:

1. Package main, nama paket khusus dalam bahasa pemrograman Go yang menunjukkan bahwa paket tersebut berisi kode untuk aplikasi yang dapat dieksekusi. Selain itu, package main ini merupakan cara untuk mengorganisir kode kedalam unit yang lebih
2. Func main(), fungsi utama yang akan dieksekusi ketika program dijalankan.

Tipe Dalam program Go terdapat proses yang melibatkan beberapa tahapan, seperti:

#### A. Koding

Koding merupakan menuliskan instruksi atau sumber code dalam bahasa pemrograman Go untuk melakukan atau mengerjakan perintah tertentu.

#### B. Kompilasi

Menerjemahkan sumber kode menjadi kode mesin yang nantinya dapat dijalankan langsung oleh komputer. Kemudian proses kompilasi ini akan diproses oleh kompilator. Kompilator ini berfungsi untuk menerjemahkan kode sumber yang dituliskan dalam bahasa pemrograman.

### Tipe Data dan Instruksi Dasar

#### 1. Data dan Variabel

Variabel adalah nama dari suatu lokasi di memori, yang data dengan tipe tertentu dapat disimpan.

#### 2. Instruksi Dasar

Instruksi dasar dalam pemrograman Go (Golang) adalah perintah-perintah yang digunakan untuk menjalankan operasi dasar dalam kode.

#### 3. Konstanta Simbolik

Konstanta dapat diberi nama untuk memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut. Seperti Pi untuk merepresentasikan konstanta phi.

## II. GUIDED

### 1. Guided 1

#### Sourcecode

```
package main

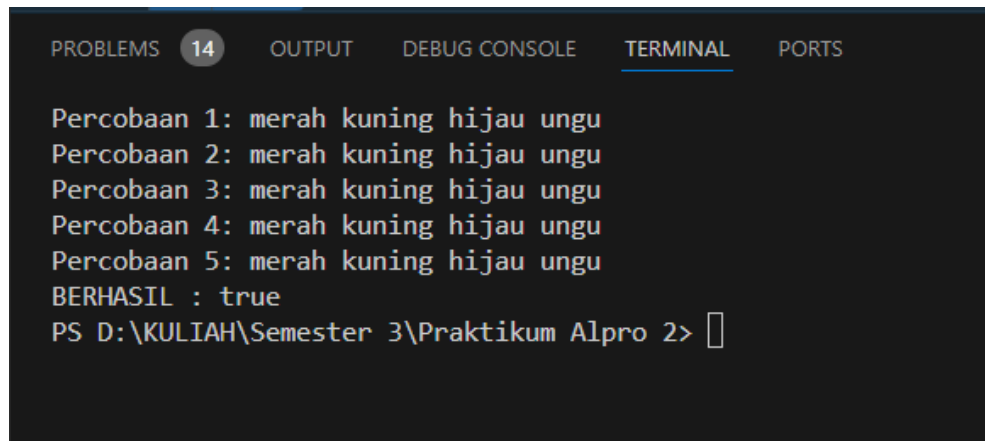
import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    correctOrder := []string{"merah", "kuning", "hijau",
    "ungu"}

    reader := bufio.NewReader(os.Stdin)
    success := true
    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

        colors := strings.Split(input, " ")
        for j := 0; j < 4; j++ {
            if colors[j] != correctOrder[j] {
                success = false
                break
            }
        }
        if !success {
            break
        }
    }
    if success {
        fmt.Println("BERHASIL : true")
    } else {
        fmt.Println("BERHASIL : false")
    }
}
```

#### Screenshoot Outpu

A screenshot of a Go IDE terminal window. The terminal has tabs for PROBLEMS, 14, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The output shows five test cases, each with the same input: 'merah kuning hijau ungu'. The output for each case is 'BERHASIL : true'. The prompt is 'PS D:\KULIAH\Semester 3\Praktikum Alpro 2>'.

```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

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:\KULIAH\Semester 3\Praktikum Alpro 2> 
```

### Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

### 2. Guided 2

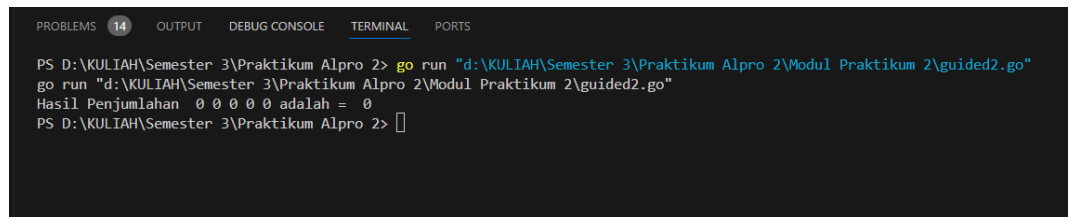
#### Sourcecode

```
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 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 2\guided2.go"
go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 2\guided2.go"
Hasil Penjumlahan 0 0 0 0 0 adalah = 0
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> █
```

## Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

### 3. Guided 3

#### Sourcecode

```
package main

import "fmt"

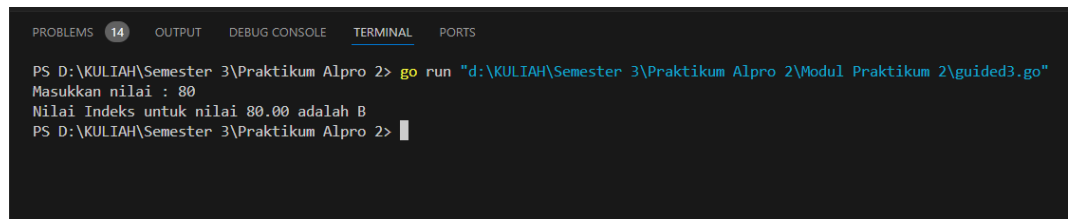
func main() {
    var nam float32
    var nmk string

    fmt.Print("Masukkan nilai : ")
    fmt.Scan(&nam)

    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "B"
    } else if nam > 65 {
        nmk = "C"
    } else if nam > 50 {
        nmk = "D"
    } else if nam > 40 {
        nmk = "E"
    } else {
        nmk = "F"
    }

    fmt.Printf("Nilai Indeks untuk nilai %.2f adalah\n%s\n", nam, nmk)
}
```

#### Screenshoot Outpu



```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 2\guided3.go"  
Masukkan nilai : 80  
Nilai Indeks untuk nilai 80.00 adalah B  
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> █
```

## Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

## 4. UNGUIDED

### 1. Soal Studi Case

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 disimpang ke dalam pita.

#### Sourcecode

```
package main
package main

import (
    "fmt"
    "strings"
)

// Fungsi untuk meminta input bunga hingga input
// "SELESAI" dimasukkan
func InputBunga_158(n int) (string, int) {
    var Bunga []string
    count := 0
    for i := 0; i < n; i++ {
        var bunga string
        fmt.Printf("Bunga %d: ", i+1)
        fmt.Scanln(&bunga)

        // Jika pengguna memasukkan "SELESAI", proses
        // input berhenti
        if strings.ToLower(bunga) == "selesai" {
            break
        }

        Bunga = append(Bunga, bunga)
        count++
    }
    return strings.Join(Bunga, " - "), count
}

func main() {
    var n int
    for {
        // Meminta input nilai N
        fmt.Print("Masukkan N (bilangan positif atau
ketik 'SELESAI' untuk berhenti): ")
        var input string
        fmt.Scanln(&input)

        // Jika input adalah "SELESAI", program akan
        // berhenti
        if strings.ToLower(input) == "selesai" {
```

```

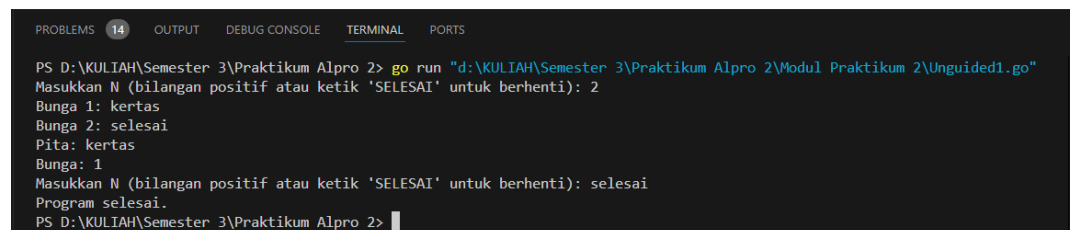
        fmt.Println("Program selesai.")
        break
    }

    // Mengonversi input ke integer
    fmt.Sscan(input, &n)

    if n > 0 {
        // Memanggil fungsi untuk memasukkan nama
        bunga berulang
        result, count := InputBunga_158(n)
        fmt.Printf("Pita: %s\n", result)
        fmt.Printf("Bunga: %d\n", count)
    } else {
        // Jika N = 0 atau input negatif
        fmt.Println("Pita: ")
        fmt.Println("Bunga: 0")
    }
}
}

```

## Screenshoot Output



```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 2\Unguided1.go"
Masukkan N (bilangan positif atau ketik 'SELESAI' untuk berhenti): 2
Bunga 1: kertas
Bunga 2: selesai
Pita: kertas
Bunga: 1
Masukkan N (bilangan positif atau ketik 'SELESAI' untuk berhenti): selesai
Program selesai.
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>

```

## Deskripsi Program

Program di atas meminta pengguna untuk memasukkan beberapa nama bunga hingga mencapai jumlah tertentu atau sampai pengguna mengetik "SELESAI" untuk menghentikan proses. Pertama, program akan meminta pengguna untuk memasukkan sebuah angka positif (N), yang menunjukkan jumlah bunga yang akan dimasukkan. Jika pengguna memasukkan kata "SELESAI", program akan berhenti. Setelah itu, pengguna akan diminta untuk memasukkan nama-nama bunga satu per satu. Jika pengguna mengetik "SELESAI" di tengah-tengah pengisian nama bunga, proses input berhenti lebih awal. Nama-nama bunga yang dimasukkan kemudian digabung menjadi sebuah pita teks yang dipisahkan oleh tanda "-", dan program akan menampilkan jumlah bunga yang berhasil dimasukkan. Jika nilai N yang dimasukkan adalah nol atau



negatif, program hanya akan menampilkan pita kosong dan jumlah bunga sebagai 0.

## 2. Soal Studi Case

Buatlah program pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing – masing isi kantong terpal. Program akan yterus meminta input bilangan tersebut hingga salah satu kantong terpal berisi 9 kg atau lebih.

### Sourcecode

```
package main

import "fmt"

func main() {
    var beratKiri, beratKanan float64

    for {
        // Meminta input berat belanjaan di kedua
        kantong
        fmt.Print("Masukkan berat belanjaan di kedua
        kantong (kiri kanan): ")
        fmt.Scanln(&beratKiri, &beratKanan)

        // Menampilkan total berat kantong
        totalBerat := beratKiri + beratKanan
        fmt.Printf("Total berat kantong: %.2f kg\n",
        totalBerat)

        // Logika pengecekan apakah sepeda motor oleng
        (true jika selisih berat terlalu besar)
        var oleng bool

        // Cek selisih berat
        selisihBerat := beratKiri - beratKanan
        if selisihBerat < 0 {
            selisihBerat = -selisihBerat // Ambil nilai
            absolut
        }

        if selisihBerat >= 9 { // Tambahan logika untuk
            selisih lebih dari sama dengan 9 kg
            oleng = true
        } else if beratKiri > beratKanan && beratKiri >=
            1.5*beratKanan {
            oleng = true
        } else if beratKanan > beratKiri && beratKanan
            >= 1.5*beratKiri {
            oleng = false
        } else {
```

```

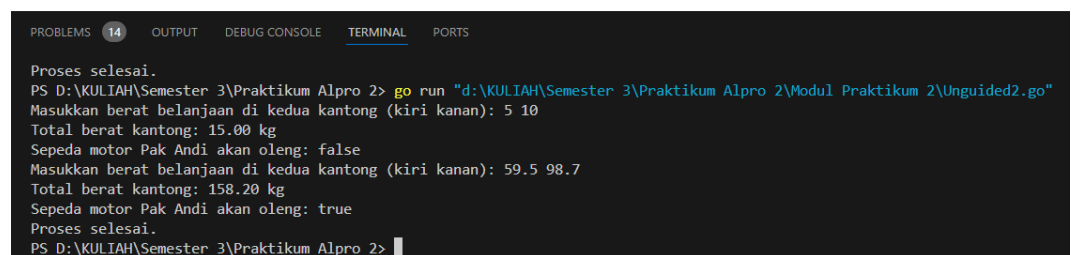
        oleng = false
    }

    // Menampilkan hasil apakah sepeda motor oleng
    atau tidak
    fmt.Printf("Sepeda motor Pak Andi akan oleng:
    %t\n", oleng)

    // Memeriksa jika salah satu kantong beratnya
    lebih dari atau sama dengan 9 kg
    if totalBerat > 150 {
        fmt.Println("Proses selesai.")
        break
    }
}
}

```

## Screenshoot Output



```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Proses selesai.
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 2\Unguided2.go"
Masukkan berat belanja di kedua kantong (kiri kanan): 5 10
Total berat kantong: 15.00 kg
Sepeda motor Pak Andi akan oleng: false
Masukkan berat belanja di kedua kantong (kiri kanan): 59.5 98.7
Total berat kantong: 158.20 kg
Sepeda motor Pak Andi akan oleng: true
Proses selesai.
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>

```

## Deskripsi Program

Program ini meminta pengguna untuk memasukkan berat belanjaan yang ditempatkan di dua kantong (kiri dan kanan) pada sepeda motor Pak Andi. Setelah input berat dari kedua kantong diberikan, program menghitung dan menampilkan total berat belanjaan. Kemudian, program mengecek apakah sepeda motor akan oleng berdasarkan perbandingan berat di antara kedua kantong. Jika selisih berat antara kantong kiri dan kanan lebih dari 9 kg, atau jika salah satu kantong beratnya 1,5 kali lebih besar dari kantong lainnya, maka motor dianggap oleng. Program terus meminta input hingga total berat belanjaan melebihi 150 kg, yang akan menghentikan program dengan menampilkan pesan "Proses selesai."

### 3. Soal Studi Case

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

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

Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

Nilai K = <u>100</u> Nilai f(K) = 1.0000061880
---

$\sqrt{2}$  merupakan bilangan irasional. Meskipun demikian, nilai tersebut dapat dihampiri dengan rumus berikut:

$$\sqrt{2} = \prod_{k=0}^{\infty} \frac{(4k+2)^2}{(4k+1)(4k+3)}$$

Modifikasi program sebelumnya yang menerima input integer  $K$  dan menghitung  $\sqrt{2}$  untuk  $K$  tersebut. Hampiran  $\sqrt{2}$  dituliskan dalam ketelitian 10 angka di belakang koma.

Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

1	Nilai K = <u>10</u> Nilai akar 2 = 1.4062058441
2	Nilai K = <u>100</u> Nilai akar 2 = 1.4133387072
3	Nilai K = <u>1000</u> Nilai akar 2 = 1.4141252651

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung akar 2 berdasarkan nilai K
func HitungAkar2(K int) float64 {
    akar2 := 1.0 // Nilai awal

    // Hitung produk dari deret untuk setiap k dari 0
    sampai K
```

```

    for k := 0; k <= K; k++ {
        numerator := math.Pow(float64(4*k+2), 2)
        denominator := float64((4*k + 1) * (4*k + 3))
        akar2 *= numerator / denominator
    }
    return akar2
}

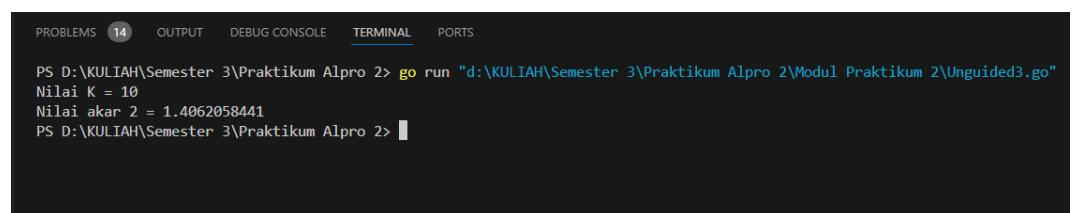
func main() {
    var K int

    // Input nilai K dari pengguna
    fmt.Print("Nilai K = ")
    fmt.Scan(&K)

    // Hitung dan tampilkan hasil dengan 10 angka di
    belakang koma
    hasil := HitungAkar2(K)
    fmt.Printf("Nilai akar 2 = %.10f\n", hasil)
}

```

## Screenshoot Output



```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 2\Unguided3.go"
Nilai K = 10
Nilai akar 2 = 1.4062058441
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>

```

## Deskripsi Program

Program di atas bertujuan untuk menghitung nilai aproksimasi akar kuadrat dari 2 ( $\sqrt{2}$ ) berdasarkan nilai input K yang diberikan pengguna. Algoritmanya menggunakan deret khusus yang menghitung produk dari beberapa suku, di mana setiap suku dihitung dari perbandingan kuadrat penyebut dan pembilang yang mengikuti pola tertentu. Program dimulai dengan nilai awal  $\text{akar2} = 1.0$ , kemudian secara berulang mengalikan hasil produk dari suku-suku tersebut untuk setiap k dari 0 hingga K. Setiap suku dalam deret dihitung dengan rumus seperti pada soal. Setelah perhitungan selesai, hasil akhir ditampilkan dengan presisi 10 angka di belakang koma.

#### 4. Soal Studi Case

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 1 kg) digratiskan biayanya apabila total berat ternyata lebih dari 10 kg.

#### Sourcecode

```
package main

import (
    "fmt"
)

func HitungBiayaPos_158(beratGram int) int {
    // Konstanta biaya dasar
    const biayaPerKg = 10000
    const biayaPerGramTambahan = 5
    const biayaPerGramKurang = 15

    // Konversi berat ke kg dan gram
    beratKg := beratGram / 1000
    sisaGram := beratGram % 1000

    // Jika berat total lebih dari 10 kg, biaya sisa
    gram digratiskan
    if beratKg >= 10 {
        sisaGram = 0
    }

    // Hitung biaya berdasarkan kondisi
    totalBiaya := beratKg * biayaPerKg
    if sisaGram > 500 {
        // Jika sisa gram lebih dari 500g, biaya Rp. 15
        per gram
        totalBiaya += sisaGram * biayaPerGramKurang
    } else if sisaGram > 0 {
        // Jika sisa gram kurang dari atau sama dengan
        500g, biaya Rp. 5 per gram
        totalBiaya += sisaGram * biayaPerGramTambahan
    }

    return totalBiaya
}
```

```

func main() {
    // Contoh 1
    beratParcel1 := 8500
    fmt.Printf("Contoh #1\nBerat parcel: %d gram\nTotal
biaya: Rp. %d\n\n", beratParcel1, HitungBiayaPos_158
(beratParcel1))

    // Contoh 2
    beratParcel2 := 9250
    fmt.Printf("Contoh #2\nBerat parcel: %d gram\nTotal
biaya: Rp. %d\n\n", beratParcel2, HitungBiayaPos_158
(beratParcel2))

    // Contoh 3
    beratParcel3 := 11750
    fmt.Printf("Contoh #3\nBerat parcel: %d gram\nTotal
biaya: Rp. %d\n", beratParcel3, HitungBiayaPos_158
(beratParcel3))
}

```

## Screenshoot Output

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Contoh #1
Berat parcel: 8500 gram
Total biaya: Rp. 82500

Contoh #2
Berat parcel: 9250 gram
Total biaya: Rp. 91250

Contoh #3
Berat parcel: 11750 gram
Total biaya: Rp. 110000
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>

```

## Deskripsi Program

Program di atas menghitung biaya pengiriman pos berdasarkan berat barang dalam gram. Pertama, berat parcel diubah menjadi kilogram dan gram. Jika berat barang mencapai 10 kilogram atau lebih, sisa

gramnya tidak dihitung, artinya biaya tambahan untuk gram dibebaskan. Untuk barang yang beratnya kurang dari 10 kg, biaya dihitung dengan dua kategori: jika sisa gram lebih dari 500g, setiap gram dikenakan biaya Rp. 15, sedangkan jika kurang dari atau sama dengan 500g, setiap gram dikenakan biaya Rp. 5. Program ini kemudian menampilkan contoh perhitungan untuk tiga berat parcel berbeda: 8500 gram, 9250 gram, dan 11750 gram, dengan hasil biaya pengiriman yang sesuai.

## 5. Soal Studi Case

Sebuah bilangan bulat  $b$  memiliki factor bilangan  $f > 0$  jika  $f$  habis membagi  $b$ . contoh : 2 merupakan factor 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 factor dari bilangan tersebut! Bilangan bulat  $b > 0$  merupakan bilangan prima  $p$  jika dan hanya jika memiliki persis dua factor bilangan saja, yaitu 1 dan dirinya sendiri. Lanjutkan program sebelumnya, setelah menerima masukan sebuah bilangan bulat  $b > 0$ . Program tersebut mencari dan menampilkan semua factor bilangan tersebut. Kemudian, program menentukan apakah  $b$  merupakan bilangan prima.

### Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk mencari faktor dari bilangan b
func CariFaktor(b int) []int {
    var faktor []int
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            faktor = append(faktor, i)
        }
    }
    return faktor
}

// Fungsi untuk memeriksa apakah bilangan prima
func ApakahPrima(b int) bool {
    faktor := CariFaktor(b)
    // Bilangan prima hanya memiliki 2 faktor: 1 dan
    dirinya sendiri
    return len(faktor) == 2
}

func main() {
```

```

var b int

// Meminta input bilangan bulat b lebih dari 1
fmt.Print("Masukkan bilangan bulat b > 1: ")
fmt.Scan(&b)

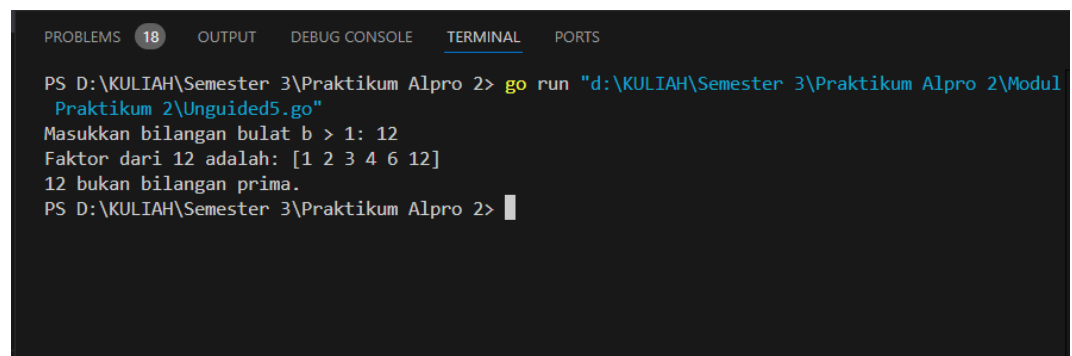
if b <= 1 {
    fmt.Println("Bilangan harus lebih besar dari
1.")
    return
}

// Mencari dan menampilkan semua faktor dari
bilangan b
faktor := CariFaktor(b)
fmt.Printf("Faktor dari %d adalah: %v\n", b, faktor)

// Menentukan apakah b merupakan bilangan prima
if ApakahPrima(b) {
    fmt.Printf("%d adalah bilangan prima.\n", b)
} else {
    fmt.Printf("%d bukan bilangan prima.\n", b)
}
}

```

## Screenshoot Output



```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul
Praktikum 2\Unguided5.go"
Masukkan bilangan bulat b > 1: 12
Faktor dari 12 adalah: [1 2 3 4 6 12]
12 bukan bilangan prima.
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>

```

## Deskripsi Program

Program di atas menerima input bilangan bulat  $(b)$  yang lebih besar dari 1, kemudian mencari dan menampilkan semua faktor dari bilangan tersebut. Pertama, fungsi `CariFaktor` memeriksa setiap angka dari 1 hingga  $(b)$  untuk menentukan apakah angka tersebut merupakan faktor, yaitu apakah  $(b)$  habis dibagi angka tersebut. Faktor-faktor ini disimpan dalam sebuah slice dan ditampilkan. Selanjutnya, fungsi



`ApakahPrima` digunakan untuk memeriksa apakah  $(b)$  adalah bilangan prima. Bilangan prima hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri. Jika  $(b)$  hanya memiliki dua faktor, maka  $(b)$  dinyatakan sebagai bilangan prima, jika tidak, program menyatakan bahwa  $(b)$  bukan bilangan prima.

## **DAFTAR PUSTAKA**

Asisten Praktikum, "Modul 2 Review Struktur Kontrol ", Learning Management System, 2024.