LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL II

REVIEW STRUKTUR KONTROL

Disusun Oleh :

Naya Putwi Setiasih / 2311102155

S1 11 IF - 05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Golang (Go) merupakan bahasa pemrograman yang dirancang oleh Google dengan fokus pada kecepatan eksekusi, efisien, dan kesederhanaan.

- Struktur Program Go
  Kerangka program dalam bahasa pemrograman Go yaitu :
  - Package main merupakan penanda bahwa file ini berisi program utama.
  - Func main() berisi kode utama dari sebuah program Go
- Tipe Data dan Instruksi Dasra
  - Data dan Vaariabel
    Variabel adalah nama dari suatu lokasi di meori, yang data dengan tipe tertentu dapat disimpan. Nama variabel dimulai dengan huruf dan dapat diikuti dengan sejumlah huruf, angka, atau garisbawah.
  - Instruksi Dasar
  - Konstanta Simbolik
    Konstanta dapat diberi nama untuk memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut.
- Struktur Kontrol Perulangan
  Go hanya mempunyai kata kunci for untuk semua jenis perulangan dalam notasi algoritma. Dalam konsep pemrograman, setiap rancangan algoritma harus memenuhi syarat satu pintu masuk atau satu pintu keluar.
  - Bentuk While-Loop
    Bentuk while-loop memastikan setiap kali memasuki loop, ada kondisi yang harus dipenuhi (benar/true). Saat keluar dari loop, maka nilai kondisi tersebut pasti false/salah.
  - Bentuk Repeat-Until
    Bentuk repeat-until diperulangan dilakukan terus menerus sampai kondisi terpenuhi.
- Struktur Kontrol Percabangan
  - Bentuk if-else
    Bentuk if-else dalam bahasa Go merupakan satu instruksi if-else-if.
  - Bentuk Switch-Case
    Dalam bahasa Go terdapat dua bentuk switch-case. Bentuk ekspresi pada perintah switch dan nilai ditulis setiap label case. Bentuk yang kedua mempunyai switch tanpa ekspresi, tetapi setiap case boleh berisi boolean.

II. GUIDED

1.

Soal Studi Case

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

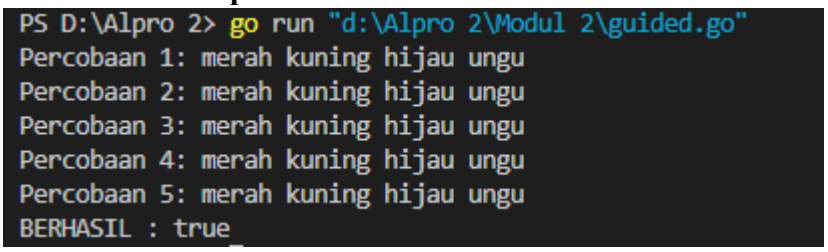
Buatlah sebuah program yang menerima input berupa warna dari 4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan true apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna lainnya.

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 Output



Deskripsi Program

Program ini ditulis dalam bahasa pemrograman Go dan bertujuan untuk menguji kemampuan pengguna dalam memasukkan urutan warna yang benar. Urutan warna yang harus dimasukkan adalah merah, kuning, hijau, dan ungu. Program meminta pengguna untuk melakukan hingga lima percobaan. Pada setiap percobaan, pengguna diminta untuk memasukkan empat warna yang dipisahkan oleh spasi. Setelah menerima input, program akan memvalidasi apakah urutan yang dimasukkan sesuai dengan urutan yang benar. Jika ada ketidakcocokan, program akan menghentikan proses dan mengeluarkan pesan "BERHASIL : false". Namun, jika pengguna berhasil memasukkan urutan yang benar dalam salah satu percobaan, program akan mencetak "BERHASIL : true". Dengan menggunakan package bufio, fmt, os, dan strings, program ini memberikan pengalaman interaktif yang sederhana namun efektif untuk menguji ingatan pengguna mengenai urutan warna.

2. **Sourcecode**

```go
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

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 2\guided 2.go"
0 + 0
Hasil Penjumlahan  0 0 0 0 0 adalah =  0
PS D:\Alpro 2>
```

Deskripsi Program

Program di atas ditulis dalam bahasa pemrograman Go dan berfungsi untuk menjumlahkan lima bilangan bulat yang dimasukkan oleh pengguna. Pertama, program mendeklarasikan lima variabel bertipe integer, yaitu a, b, c, d, dan e, serta satu variabel hasil untuk menyimpan hasil penjumlahan. Program kemudian menggunakan fungsi fmt.Scanln untuk membaca input dari pengguna, yang diharapkan berupa lima bilangan bulat yang dipisahkan oleh spasi. Setelah menerima input, program menghitung total penjumlahan kelima bilangan tersebut dan menyimpannya dalam variabel hasil. Terakhir, program mencetak hasil penjumlahan beserta bilangan-bilangan yang dimasukkan ke layar dengan format yang jelas. Dengan demikian, program ini memberikan cara sederhana bagi pengguna untuk melakukan operasi penjumlahan pada beberapa angka sekaligus.

3.

Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian 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 <= 57.5	C
40 < NAM <= 50	D
NAM <=40	E

Program berikut menerima input sebuah bilangan riil yang menyatakan NAM. Program menghitung NMK dan menampilkannya.

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
%s\n", nam, nmk)
}
```

Screenshoot Output

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 2\guided 3.go"
Masukkan nilai : 77.5
Nilai Indeks untuk nilai 77.50 adalah B
PS D:\Alpro 2>
```

Deskripsi Program

Program di atas ditulis dalam bahasa pemrograman Go dan bertujuan untuk mengonversi nilai numerik yang dimasukkan oleh pengguna menjadi indeks nilai huruf. Program dimulai dengan mendeklarasikan dua variabel: nam bertipe float32 untuk menyimpan nilai yang dimasukkan, dan nmk bertipe string untuk menyimpan indeks nilai huruf. Setelah menampilkan prompt "Masukkan nilai :", program menggunakan fmt.Scan untuk membaca input dari pengguna. Berdasarkan nilai yang dimasukkan, program menggunakan serangkaian pernyataan if-else untuk menentukan indeks nilai huruf sesuai dengan rentang yang telah ditentukan: A untuk nilai di atas 80, B untuk nilai di atas 72.5, C untuk di atas 65, D untuk di atas 50, E untuk di atas 40, dan F untuk nilai di 40 atau kurang. Setelah menentukan indeks, program mencetak hasilnya dengan format yang jelas, menunjukkan nilai numerik dan indeks huruf yang sesuai. Dengan demikian, program ini memberikan cara sederhana bagi pengguna untuk memahami kinerja akademis mereka melalui sistem penilaian huruf.

III. UNGUIDED

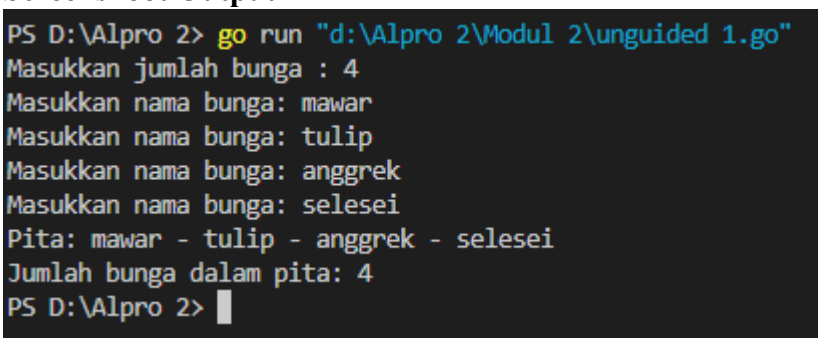
4. Suatu pita (string) berisi kumpulan nama – nama bunga yang dipisahkan oleh spasi dan '-', contoh pita diilustrasikan seperti berikut ini :
   Pita : mawar – melati – tulip – teratai – kamboja – anggrek
   Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak nol) N, kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita. Kemudian, modifikasi program sebelumnya, proses input akan erenti apabila user mengetikkan 'SELESAI'. Kemudian, tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita.

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var n int
    var pita string
    var bungaCount int

    // Meminta input dari user untuk jumlah nama bunga
    fmt.Print("Masukkan jumlah bunga : ")
    fmt.Scan(&n)

    // Meminta input nama bunga sebanyak N kali
    for i := 0; i < n; i++ {
        var bunga string
        fmt.Print("Masukkan nama bunga: ")
        fmt.Scan(&bunga)

        // Jika bunga pertama, langsung tambahkan ke
pita, jika tidak tambahkan dengan " - "
        if i == 0 {
            pita = bunga
        } else {
            pita += " - " + bunga
        }

        // Tambah jumlah bunga yang sudah dimasukkan
        bungaCount++
    }

    // Menampilkan pita setelah semua input selesai
    fmt.Println("Pita:", pita)
    fmt.Println("Jumlah bunga dalam pita:", bungaCount)
```

```
}
```

Screenshoot Output



Deskripsi Program

Program di atas ditulis dalam bahasa pemrograman Go dan berfungsi untuk mengumpulkan nama-nama bunga dari pengguna dan menyusunnya dalam format yang terpisah oleh tanda " - ". Program dimulai dengan mendeklarasikan variabel n untuk menyimpan jumlah bunga yang akan dimasukkan, pita sebagai string untuk menyimpan daftar nama bunga, dan bungaCount sebagai penghitung jumlah bunga. Setelah menampilkan prompt "Masukkan jumlah bunga :", program menggunakan fmt.Scan untuk membaca input jumlah bunga dari pengguna. Selanjutnya, program menjalankan loop sebanyak n kali, di mana pada setiap iterasi, pengguna diminta untuk memasukkan nama bunga. Jika bunga yang dimasukkan adalah yang pertama, program langsung menyimpannya ke dalam variabel pita. Untuk bunga-bunga berikutnya, program menambahkan nama bunga tersebut ke pita dengan format yang sesuai. Setelah semua input selesai, program mencetak hasil akhir berupa daftar nama bunga yang telah disusun dalam pita dan juga menampilkan jumlah total bunga yang dimasukkan. Dengan demikian, program ini menyediakan cara interaktif bagi pengguna untuk mengelola dan menampilkan daftar nama bunga dengan mudah.

2. 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.

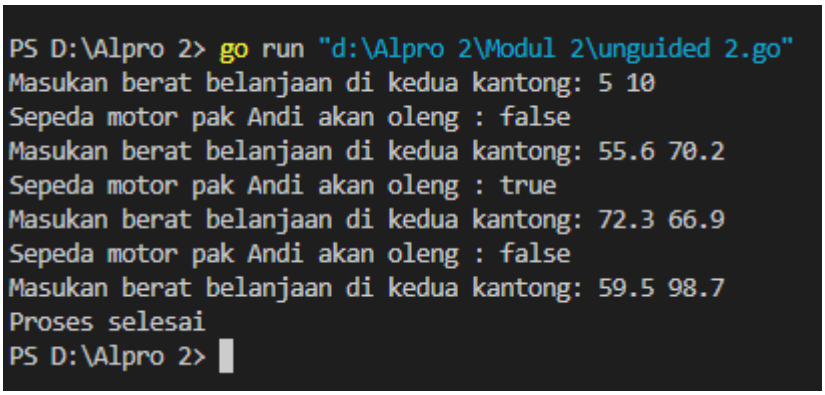
Pada modifikasi program tersebut, program akan menampilkan true jika selisih kedua sisi kantong lebih dari atau sama dengan 9 kg. Program berhenti memproses apabila total berat isi kedua kantong melebihi 250 kg atau salah satu kantong beratnya negatif.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var beratKiri, beratKanan, totalBerat float64

    // Loop yang terus berjalan hingga total berat
melebihi 150 kg atau salah satu kantong negatif
    for {
        // Meminta input dari user untuk berat belanjaan
di kedua kantong
        fmt.Print("Masukan berat belanjaan di kedua
kantong: ")
        fmt.Scan(&beratKiri, &beratKanan)

        // Cek apakah salah satu kantong beratnya
negatif
        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Proses selesai karena salah
satu kantong beratnya negatif.")
            break
        }

        // Menghitung total berat kedua kantong
        totalBerat = beratKiri + beratKanan

        // Cek apakah total berat melebihi 150 kg
        if totalBerat > 150 {
            fmt.Println("Proses selesai")
            break
        }

        // Cek apakah selisih berat kedua kantong lebih
dari atau sama dengan 9 kg
        if math.Abs(beratKiri-beratKanan) >= 9 {
            fmt.Println("Sepeda motor pak Andi akan
oleng : true")
        } else {
            fmt.Println("Sepeda motor pak Andi akan
oleng : false")
        }
```

```
        }
    }
```

Screenshoot Output



Deskripsi Program

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk memantau berat belanjaan yang dimasukkan oleh pengguna dalam dua kantong, dengan beberapa kondisi yang harus dipenuhi untuk memastikan keamanan saat berkendara. Program dimulai dengan mendeklarasikan variabel beratKiri, beratKanan, dan totalBerat bertipe float64 untuk menyimpan berat masing-masing kantong dan total beratnya. Setelah itu, program memasuki loop tak terbatas yang meminta pengguna untuk memasukkan berat belanjaan di kedua kantong. Jika salah satu dari berat yang dimasukkan negatif, program akan menghentikan proses dan mencetak pesan bahwa proses telah selesai. Selanjutnya, program menghitung total berat kedua kantong dan memeriksa apakah total tersebut melebihi 150 kg; jika ya, proses juga akan dihentikan dengan pesan yang sesuai. Selain itu, program mengevaluasi selisih antara berat kedua kantong; jika selisihnya lebih dari atau sama dengan 9 kg, program akan mencetak pesan bahwa sepeda motor Pak Andi akan oleng, menandakan potensi ketidakstabilan saat berkendara. Jika selisihnya kurang dari 9 kg, program akan mencetak pesan sebaliknya. Dengan demikian, program ini memberikan cara interaktif bagi pengguna untuk memantau dan mengelola berat belanjaan sambil mempertimbangkan faktor keselamatan saat berkendara.

3.

Diberikan sebuah persamaan sebagai berikut ini.

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

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

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

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

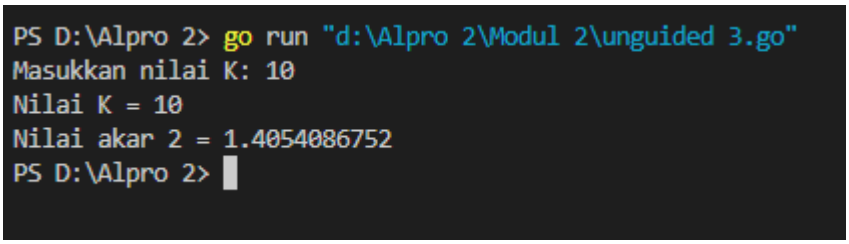
Modifikasi program sebelumnya yang menerima input integer K dan menghitung $\sqrt{2}$ untuk K tersebut. Hampiran $\sqrt{2}$ dituliskan dalam ketelitian 10 angka di belakang koma.

Sourcecode

```
package main

import (
    "fmt"
)

func f(k int) float64 {
    // Menghitung f(k) berdasarkan rumus
    numerator := float64((4*k + 2) * (4*k + 2))
    denominator := float64((4*k + 1) * (4*k + 3))
    return numerator / denominator
}

func main() {
    var k int

    // Input nilai K dari pengguna
    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&k)

    // Hitung hampiran akar 2 dengan menjumlahkan nilai
f(i) hingga i = k
    approxSqrt2 := 1.0
    for i := 0; i < k; i++ {
        approxSqrt2 *= f(i)
```

```
    }

    // Tampilkan hasilnya
    fmt.Printf("Nilai K = %d\n", k)
    fmt.Printf("Nilai akar 2 = %.10f\n", approxSqrt2)
}
```

Screenshoot Output



Deskripsi Program

Program ini ditulis dalam bahasa pemrograman Go dan bertujuan untuk menghampiri nilai akar dari 2 menggunakan rumus matematis tertentu. Program dimulai dengan mendefinisikan fungsi f(k int) yang menghitung nilai berdasarkan rumus yang melibatkan pembilang dan penyebut yang dinyatakan dalam bentuk ekspresi aritmatika. Dalam fungsi ini, pembilang dihitung sebagai kuadrat dari (4*k + 2), sedangkan penyebut dihitung sebagai hasil kali dari (4*k + 1) dan (4*k + 3). Setelah mendefinisikan fungsi, program meminta pengguna untuk memasukkan nilai k, yang akan menentukan berapa banyak iterasi yang dilakukan dalam perhitungan. Selanjutnya, program menghitung hampiran akar 2 dengan mengalikan hasil dari fungsi f(i) untuk setiap nilai i dari 0 hingga k-1. Hasil akhir, yaitu hampiran akar 2, kemudian ditampilkan dengan presisi sepuluh desimal. Dengan demikian, program ini memberikan cara interaktif bagi pengguna untuk menghitung hampiran akar 2 secara numerik menggunakan metode perhitungan berbasis rumus.

4. PT Pos membutuhkan aplikasi perhitungan biaya kirimberdasarkan berat parsel. Maka, buatlah program biaya pos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut!
   Dari berat parsel (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

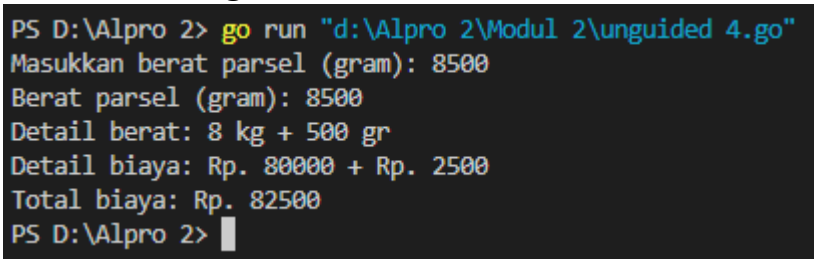
(yang kurang dari 1 kg) digratiskan biayanya apabila total berat ternyata lebih dari 10 kg.

Sourcecode

```
package main

import (
    "fmt"
)

func hitungBiayaPengiriman(berat int) (int, int, int) {
    // Konversi berat ke kg dan gram
    kg := berat / 1000
    gram := berat % 1000

    // Biaya dasar pengiriman
    biayaKg := kg * 10000
    biayaGram := 0

    // Hitung biaya tambahan untuk gram hanya jika total
kg di bawah 10 kg
    if kg < 10 {
        if gram <= 500 {
            biayaGram = gram * 5
        } else {
            biayaGram = gram * 15
        }
    }

    // Total biaya
    totalBiaya := biayaKg + biayaGram

    return biayaKg, biayaGram, totalBiaya
}

func main() {
    var berat int

    // Input berat parsel dari pengguna (dalam gram)
    fmt.Print("Masukkan berat parsel (gram): ")
    fmt.Scan(&berat)

    // Hitung biaya pengiriman
    biayaKg, biayaGram, totalBiaya :=
hitungBiayaPengiriman(berat)

    // Konversi berat ke kg dan gram untuk ditampilkan
    kg := berat / 1000
    gram := berat % 1000

    // Tampilkan hasil perhitungan
```

```
    fmt.Printf("Berat parsel (gram): %d\n", berat)
    fmt.Printf("Detail berat: %d kg + %d gr\n", kg,
gram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n",
biayaKg, biayaGram)
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}
```

Screenshoot Program



Deskripsi Program

Program ini ditulis dalam bahasa pemrograman Go dan dirancang untuk menghitung biaya pengiriman berdasarkan berat parsel yang dimasukkan oleh pengguna dalam satuan gram. Program dimulai dengan mendefinisikan fungsi hitungBiayaPengiriman, yang menerima parameter berat dalam gram dan mengembalikan biaya pengiriman berdasarkan berat tersebut. Dalam fungsi ini, berat diubah menjadi kilogram dan gram, di mana biaya dasar pengiriman dihitung berdasarkan kilogram dengan tarif Rp. 10.000 per kilogram. Jika total berat kurang dari 10 kg, biaya tambahan untuk gram dihitung; jika gram kurang dari atau sama dengan 500, tarifnya adalah Rp. 5 per gram, sedangkan untuk gram lebih dari 500, tarifnya menjadi Rp. 15 per gram. Setelah menghitung biaya per kilogram dan per gram, total biaya pengiriman dihitung dengan menjumlahkan kedua biaya tersebut. Di dalam fungsi main, program meminta pengguna untuk memasukkan berat parsel, kemudian memanggil fungsi hitungBiayaPengiriman untuk mendapatkan detail biaya dan total biaya pengiriman. Akhirnya, program menampilkan informasi lengkap mengenai berat parsel, rincian biaya, dan total biaya dalam format yang jelas. Dengan demikian, program ini memberikan solusi interaktif bagi pengguna untuk menghitung biaya pengiriman secara akurat berdasarkan berat parsel yang mereka masukkan.

5. 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 bilangn 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"
)

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

    // Menemukan dan menampilkan semua faktor
    fmt.Print("Faktor: ")
    isPrime := true
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Printf("%d ", i)
            if i != 1 && i != b {
                isPrime = false
            }
        }
    }
    fmt.Println()

    // Menentukan apakah bilangan merupakan bilangan
prima
    if isPrime && b > 1 {
        fmt.Println("Prima: true")
    } else {
        fmt.Println("Prima: false")
    }
}
```

Screenshoot Output

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 2\unguided 6.go"
Masukkan bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Alpro 2>
```

Deskripsi Program

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menemukan faktor dari sebuah bilangan yang dimasukkan oleh pengguna serta menentukan apakah bilangan tersebut merupakan bilangan prima. Program dimulai dengan mendeklarasikan variabel b untuk menyimpan bilangan yang akan diperiksa. Setelah meminta pengguna untuk memasukkan bilangan, program kemudian mencetak semua faktor dari bilangan tersebut dengan menggunakan loop yang berjalan dari 1 hingga b. Di dalam loop, jika b dapat dibagi habis oleh i, maka i dicetak sebagai faktor. Selain itu, program juga memeriksa apakah i bukan 1 atau b itu sendiri; jika demikian, variabel isPrime di-set menjadi false, menandakan bahwa bilangan tersebut bukan prima. Setelah semua faktor dicetak, program mengevaluasi nilai dari isPrime. Jika bilangan tersebut adalah lebih dari 1 dan tetap terdeteksi sebagai prima, program akan mencetak "Prima: true"; jika tidak, akan mencetak "Prima: false". Dengan demikian, program ini memberikan cara interaktif bagi pengguna untuk memahami faktor-faktor dari bilangan dan status primalitasnya dengan mudah.

DAFTAR PUSTAKA

Maulana, F. R., Aziz, F. A., Kholiq, N. A., Ramadhan, R. I., & Ramadhan, M. I. (2024). Panduan Golang Dan JavaScript Dalam Pengembangan Web Service Presensi. Penerbit Buku Pedia.

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