

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
REVIEW STRUKTUR KONTROL**



Disusun Oleh:
Bayu Kuncoro Adi / 2311102031
S1 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

a. Pengertian Go

GO adalah bahasa pemrograman yang digunakan untuk pembuatan aplikasi berat, seperti sistem operasi dan aplikasi web, sehingga programmer yang memahami GO dapat memanfaatkan peluang karir sebagai pembuat aplikasi berat.

1. Tipe Data

- Integer (Bilangan Bulat)

Go menyediakan beberapa tipe integer:

- int: ukuran tergantung platform (32 atau 64 bit)
- int8: -128 hingga 127
- int16: -32768 hingga 32767
- int32: -2147483648 hingga 2147483647
- int64: -9223372036854775808 hingga 9223372036854775807
- uint: unsigned integer, tergantung platform
- uint8: 0 hingga 255
- uint16: 0 hingga 65535
- uint32: 0 hingga 4294967295
- uint64: 0 hingga 18446744073709551615

- Float

Go memiliki dua tipe float:

- float32: $\pm 1.18 \times 10^{-38}$ hingga $\pm 3.4 \times 10^{38}$
- float64: $\pm 2.23 \times 10^{-308}$ hingga $\pm 1.80 \times 10^{308}$

- String

String di Go adalah urutan karakter UTF-8 yang tidak dapat diubah (immutable).

Fitur string di Go:

- Dapat menggunakan tanda kutip ganda (") untuk string biasa
- Dapat menggunakan tanda kutip balik (') untuk raw string

- Boolean

Tipe bool memiliki dua nilai: true atau false

2. Perulangan

- For Tradhisional

```
for i := 0; i < 7; i++ {  
    fmt.Println(i)  
}
```

- For sebagai While

```
counter := 0  
for counter < 3 {  
    fmt.Println(counter)  
    counter++  
}
```

- **For Range**
 slice := []string{"nasi", "bubur", "ayam"}
 for index, nilai := range slice {
 fmt.Printf("Index: %d, Nilai: %s\n", index, nilai)
 }
- **Infinite Loop**
 for {
 fmt.Println("Loop tidak henti")
 // Gunakan 'break' untuk keluar dari loop
 }
- **Break**
 outerLoop:
 for i := 0; i < 4; i++ {
 for j := 0; j < 4; j++ {
 if i*j > 12 {
 break outerLoop
 }
 }
 }
 }

3. Percabangan

- **IF ELSE**
 Struktur if-else di Go memiliki beberapa fitur unik:
 - Tidak memerlukan tanda kurung () untuk kondisi
 - Kurung kurawal {} wajib digunakan
 - Dapat memiliki statement singkat sebelum kondisi
- **SWITCH**
 Switch di Go lebih fleksibel dibanding bahasa lain:
 - Tidak perlu break di setiap case
 - Case bisa memiliki multiple values
 - Bisa menggunakan kondisi di setiap case

II. GUIDED

1. Program untuk membaca dan menampilkan nama.

```
package main

import "fmt"

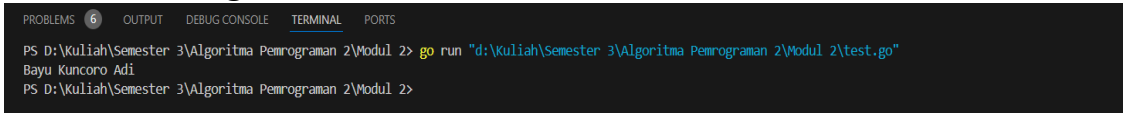
func main() {

    nama := "Bayu Kuncoro Adi"

    fmt.Print(nama)

}
```

Screenshoot Program



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\test.go"
Bayu Kuncoro Adi
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2>
```

Deskripsi dan Cara Kerja Program

Program di atas adalah program sederhana dalam bahasa pemrograman Go yang menampilkan teks berupa nama "Bayu Kuncoro Adi" ke layar. Pertama, variabel `nama` didefinisikan dan diisi dengan string "Bayu Kuncoro Adi". Kemudian, fungsi `fmt.Print()` digunakan untuk mencetak isi dari variabel `nama` ke layar tanpa menambahkan baris baru setelahnya. Ketika program ini dijalankan, nama yang disimpan dalam variabel akan langsung ditampilkan pada terminal atau output standar.

2. Buatlah sebuah program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. 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 urutan warna lainnya. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read)

```
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 Program

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\test.go"
Bayu Kuncoro Adi
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\tempCodeRunnerFile.go"
Percobaan 1: go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\tempCodeRunnerFile.go"
BERHASIL : false
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\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:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> |
```

Deskripsi dan Cara Kerja Program

Program di atas adalah program dalam bahasa Go yang mengecek apakah pengguna dapat memasukkan urutan warna yang benar dalam 5 percobaan. Urutan warna yang benar adalah "merah", "kuning", "hijau", dan "ungu". Program menggunakan paket `bufio` untuk membaca input dari pengguna, yang kemudian dipisahkan berdasarkan spasi untuk mendapatkan urutan warna. Untuk setiap percobaan, program membandingkan input pengguna dengan urutan warna yang benar. Jika input salah pada salah satu percobaan, program segera menghentikan pemeriksaan dan menampilkan "BERHASIL : false". Jika seluruh percobaan berhasil (semua urutan warna benar), program akan menampilkan "BERHASIL : true".

3. Penjumlahan dari angka yang diinputkan pengguna

```
package main

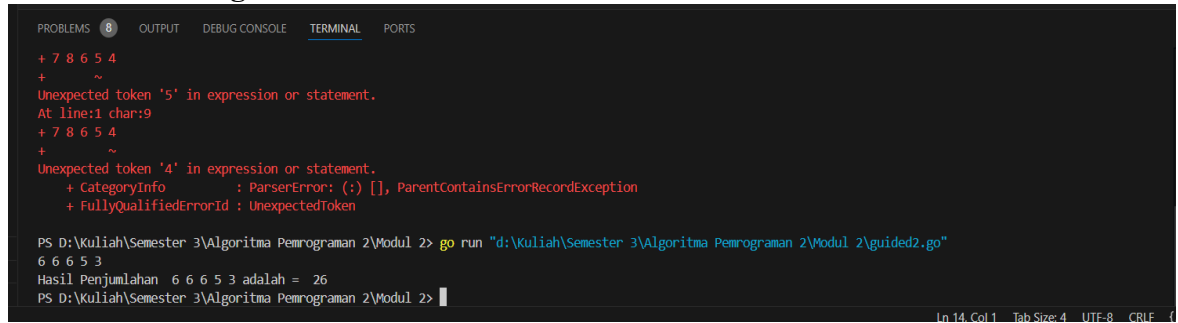
import "fmt"

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)
}
```

Screenshot Program



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
+ 7 8 6 5 4
+ ~
Unexpected token '5' in expression or statement.
At line:1 char:9
+ 7 8 6 5 4
+ ~
Unexpected token '4' in expression or statement.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnexpectedToken

PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\guided2.go"
6 6 6 5 3
Hasil Penjumlahan 6 6 6 5 3 adalah = 26
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2>
```

Deskripsi Program

Program di atas merupakan program dalam bahasa Go yang meminta input dari pengguna berupa lima angka integer ('a', 'b', 'c', 'd', dan 'e'), kemudian menghitung dan menampilkan hasil penjumlahannya. Pertama, variabel-variabel untuk lima angka dan satu variabel hasil dijabarkan dengan tipe data integer. Fungsi `fmt.Scanln()` digunakan untuk membaca input dari pengguna, yang kemudian disimpan dalam variabel-variabel 'a', 'b', 'c', 'd', dan 'e'. Setelah itu, kelima angka tersebut dijumlahkan dan disimpan dalam variabel 'hasil'. Terakhir, program menampilkan hasil penjumlahan ke layar dengan menggunakan `fmt.Println()`, di mana angka yang dimasukkan dan hasil penjumlahannya ditampilkan bersama dalam satu baris output.

4. Diberikan sebuah nilai akhir mata kuliah (NAM) [0...100] dan standar penilaian data 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

```
package main

import "fmt"

func main() {
```

```

var nam float32
var nmk string

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

// Logika penentuan nilai huruf berdasarkan nilai numerik
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"
}

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

```

Screenshoot Program

```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
At line:1 char:9
+ 7 8 6 5 4
+ ~
Unexpected token '4' in expression or statement.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnexpectedToken

PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\guided2.go"
6 6 6 5 3
Hasil Penjumlahan 6 6 6 5 3 adalah = 26
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\tempCodeRunnerFile.go"
Masukkan nilai : 78
Nilai Indeks untuk nilai 78.00 adalah B
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2>

```

Deskripsi dan Cara Kerja Program

Program di atas adalah program dalam bahasa Go yang meminta input berupa nilai numerik dari pengguna dan kemudian menentukan nilai indeks huruf berdasarkan nilai tersebut. Pertama, program mendeklarasikan dua variabel: `nam` bertipe float32 untuk menyimpan nilai numerik dan `nmk` bertipe string untuk menyimpan nilai indeks huruf. Setelah itu, program meminta input nilai dari pengguna menggunakan `fmt.Scan(&nam)`. Program kemudian menggunakan serangkaian pernyataan kondisi (`if-else if`) untuk menentukan nilai indeks huruf berdasarkan rentang nilai numerik. Nilai huruf yang dihasilkan akan berkisar antara "A" hingga "F" sesuai dengan nilai yang diberikan. Terakhir, program menampilkan nilai numerik beserta indeks huruf yang sesuai menggunakan `fmt.Printf()` dengan format angka dua desimal.

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.

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.

N: <u>3</u>	N : <u>0</u>
Bunga 1: <u>Kertas</u>	Pita :
Bunga 2: <u>Mawar</u>	
Bunga 3: <u>Tulip</u>	
Pita: Kertas - Mawar - Tulip -	

Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetikkan ‘SELESAI’. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bunga 1: <u>Kertas</u>	Bunga 1: <u>SELESAI</u>
Bunga 2: <u>Mawar</u>	Pita :
Bunga 3: <u>Tulip</u>	Bunga: 0
Bunga 4: <u>SELESAI</u>	
Pita: Kertas - Mawar - Tulip -	
Bunga: 3	

```
package main

import "fmt"

func main() {

    var pita string
    var banyakBunga int

    for {
        var bunga string

        fmt.Print("Bunga (ketik 'SELESAI' untuk berhenti): ")
        fmt.Scanln(&bunga)
```

```

        if bunga == "SELESAI" {
            break
        }

        if pita != "" {
            pita += " - "
        }

        pita += bunga

        banyakBunga++
    }

    fmt.Println("Pita:", pita)
    fmt.Println("Banyaknya bunga:", banyakBunga)
}

```

Screenshoot Program

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\tempCodeRunnerFile.go"
Masukkan nilai : 78
Nilai Indeks untuk nilai 78.00 adalah B
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\unguided1.go"
Bunga (ketik 'SELESAI' untuk berhenti): Mawar
Bunga (ketik 'SELESAI' untuk berhenti): Melati
Bunga (ketik 'SELESAI' untuk berhenti): Tulip
Bunga (ketik 'SELESAI' untuk berhenti): Teratai
Bunga (ketik 'SELESAI' untuk berhenti): Kamboja
Bunga (ketik 'SELESAI' untuk berhenti): Anggrek
Bunga (ketik 'SELESAI' untuk berhenti): SELESAI
Pita: Mawar - Melati - Tulip - Teratai - Kamboja - Anggrek
Banyaknya bunga: 6
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2>
Ln 34, Col 2 (449 selected) Tab Size: 4 UTF-8 C

```

Deskripsi dan Cara Kerja Program

Program di atas adalah program dalam bahasa Go yang meminta input berupa nama-nama bunga dari pengguna dan menggabungkannya menjadi satu string yang diakhiri dengan jumlah total bunga yang dimasukkan. Pertama, variabel `pita` digunakan untuk menyimpan string yang berisi daftar bunga, sedangkan variabel `banyakBunga` digunakan untuk menghitung jumlah bunga yang dimasukkan. Dalam loop tak terbatas (infinite loop), pengguna diminta untuk memasukkan nama bunga satu per satu. Jika pengguna mengetik "SELESAI", loop akan berhenti. Setiap bunga yang dimasukkan akan ditambahkan ke string `pita`, dengan tanda pemisah "-" jika ada lebih dari satu bunga. Setelah loop berhenti, program menampilkan daftar bunga yang disusun dalam string `pita` dan jumlah bunga yang dihitung dengan variabel `banyakBunga`.

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

```
Masukan berat belanjaan di kedua kantong: 5.5 1.0
Masukan berat belanjaan di kedua kantong: 7.1 8.5
Masukan berat belanjaan di kedua kantong: 2 6
Masukan berat belanjaan di kedua kantong: 9 5.8
Proses selesai.
```

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

```
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.
```

```
package main

import (
    "bufio"      // Mengimpor paket bufio untuk membaca input dari
pengguna
    "fmt"        // Mengimpor paket fmt untuk menampilkan output
    "os"         // Mengimpor paket os untuk menggunakan input dari sistem
operasi
```

```

    "strconv" // Mengimpor paket strconv untuk konversi string ke tipe
data numerik
    "strings" // Mengimpor paket strings untuk manipulasi string
)

func main() {
    reader := bufio.NewReader(os.Stdin) // Membuat reader untuk membaca
input pengguna

    // Loop untuk meminta input pengguna secara berulang
    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        input, _ := reader.ReadString('\n') // Membaca input pengguna
        input = strings.TrimSpace(input)    // Menghapus spasi dan
karakter newline di awal dan akhir

        // Memisahkan input menjadi dua bagian berdasarkan spasi
        berat := strings.Split(input, " ")

        // Mengecek apakah input terdiri dari dua angka
        if len(berat) != 2 {
            fmt.Println("Input tidak valid. Masukkan dua angka.") //
Menampilkan pesan kesalahan jika input tidak valid
            continue //
Melanjutkan ke iterasi berikutnya untuk meminta input ulang
        }

        // Mengonversi input dari string ke float64
        berat1, _ := strconv.ParseFloat(berat[0], 64)
        berat2, _ := strconv.ParseFloat(berat[1], 64)

        // Jika salah satu berat negatif, sepeda tidak oleng dan keluar
dari loop
        if berat1 < 0 || berat2 < 0 {
            fmt.Println("Sepeda motor pak Andi akan oleng: false") //
Menampilkan bahwa sepeda tidak oleng
            break //
Keluar dari loop
        }

        // Jika total berat lebih dari 150 kg, sepeda tidak oleng dan
keluar dari loop
        if berat1+berat2 > 150 {
            fmt.Println("Sepeda motor pak Andi akan oleng: false") //
Sepeda tidak oleng karena berat berlebihan
            break //
Keluar dari loop
        }

        // Menghitung selisih berat antara kedua kantong
        selisih := berat1 - berat2
        if selisih < 0 {

```

```

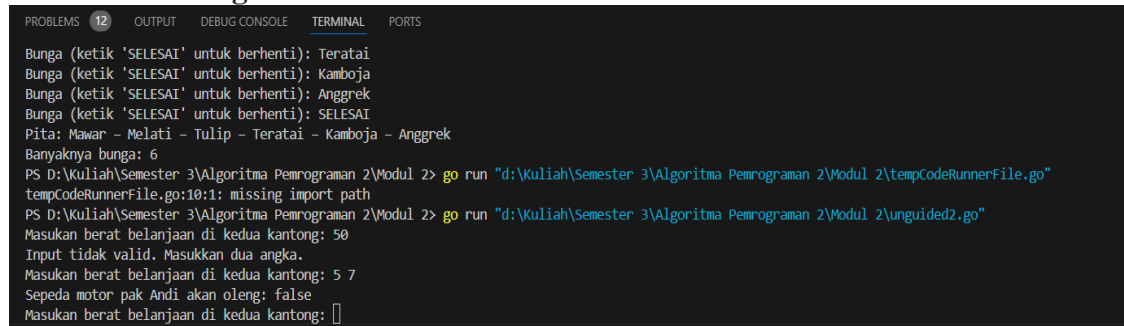
        selisih = -selisih // Mengubah selisih menjadi nilai
absolut (selisih selalu positif)
    }

    // Jika selisih berat lebih dari atau sama dengan 9 kg, sepeda
akan oleng
    if selisih >= 9 {
        fmt.Println("Sepeda motor pak Andi akan oleng: true")
    } else {
        fmt.Println("Sepeda motor pak Andi akan oleng: false")
    }
}

// Pesan yang ditampilkan setelah proses selesai
fmt.Println("Proses selesai.")
}

```

Screenshoot Program



```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Bunga (ketik 'SELESAI' untuk berhenti): Teratai
Bunga (ketik 'SELESAI' untuk berhenti): Kamboja
Bunga (ketik 'SELESAI' untuk berhenti): Anggrek
Bunga (ketik 'SELESAI' untuk berhenti): SELESAI
Pita: Mawar - Melati - Tulip - Teratai - Kamboja - Anggrek
Banyaknya bunga: 6
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\tempCodeRunnerFile.go"
tempCodeRunnerFile.go:10:1: missing import path
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\unguided2.go"
Masukan berat belanja di kedua kantong: 50
Input tidak valid. Masukkan dua angka.
Masukan berat belanja di kedua kantong: 5 7
Sepeda motor pak Andi akan oleng: false
Masukan berat belanja di kedua kantong: 

```

Deskripsi dan Cara Kerja Program

Program di atas adalah program dalam bahasa Go yang digunakan untuk menentukan apakah sepeda motor Pak Andi akan oleng berdasarkan berat belanjaan di dua kantong yang dimasukkan pengguna. Program membaca input dari pengguna berupa dua angka (berat belanjaan dalam dua kantong) menggunakan `bufio.NewReader` dan memprosesnya menjadi dua variabel `berat1` dan `berat2`. Jika input kurang dari dua angka atau terdapat berat negatif, program menghentikan loop dan menampilkan pesan bahwa sepeda tidak oleng. Program juga memeriksa apakah total berat melebihi 150 kg, di mana sepeda dipastikan tidak oleng dan loop berhenti. Selain itu, jika selisih berat antara dua kantong lebih dari atau sama dengan 9 kg, program menampilkan bahwa sepeda motor Pak Andi akan oleng, dan jika selisihnya kurang dari 9 kg, sepeda tidak oleng. Program akan terus berjalan hingga salah satu kondisi penghentian terpenuhi.

3. Diberikan sebuah persamaan sebagai berikut:

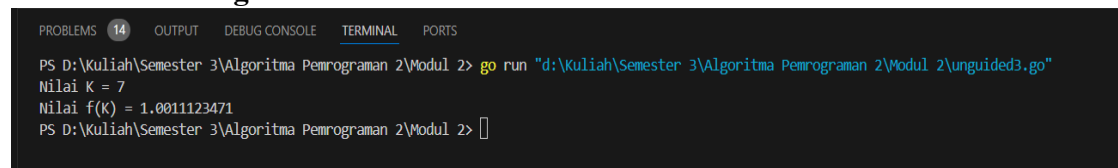
$$f(k) = \frac{(4k + 2)^2}{(4k + 1)(4k + 3)}$$

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 = 100  
Nilai f(K) = 1.0000061880
```

```
package main  
  
import (  
    "fmt"  
)  
  
func main() {  
    // Deklarasi variabel k dengan tipe float64  
    var k float64  
    // Menampilkan pesan untuk memasukkan nilai K  
    fmt.Print("Nilai K = ")  
    // Membaca input dari pengguna dan menyimpannya ke variabel k  
    fmt.Scanln(&k)  
  
    // Hitung f(K) dengan rumus: f(k) = ((4*k + 2)^2) / ((4*k + 1) *  
    (4*k + 3))  
    f_k := (4*k + 2) * (4*k + 2) / ((4*k + 1) * (4*k + 3))  
  
    // Menampilkan hasil perhitungan f(K) dengan presisi 10 angka di  
    belakang koma  
    fmt.Printf("Nilai f(K) = %.10f\n", f_k)  
}
```

Screenshot Program



```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\unguided3.go"  
Nilai K = 7  
Nilai f(K) = 1.001123471  
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> █
```

Deskripsi dan Cara Kerja Program

Program di atas adalah program dalam bahasa Go yang meminta input berupa nilai `k` dari pengguna, lalu menghitung nilai fungsi matematika `f(k)` berdasarkan rumus:

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

Program dimulai dengan mendeklarasikan variabel `k` dengan tipe `float64`. Pengguna diminta memasukkan nilai `k` melalui input yang dibaca menggunakan `fmt.Scanln()`. Setelah itu, program menghitung nilai `f(k)` menggunakan rumus yang telah ditentukan. Hasil perhitungan ditampilkan ke layar dengan presisi 10 angka di belakang koma menggunakan `fmt.Printf()`.

4. PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program BiayaPos 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. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

1	<p>Contoh #1</p> <p>Berat parcel (gram): <u>8500</u></p> <p>Detail berat: 8 kg + 500 gr</p> <p>Detail biaya: Rp. 80000 + Rp. 2500</p> <p>Total biaya: Rp. 82500</p>
2	<p>Contoh #2</p> <p>Berat parcel (gram): <u>9250</u></p> <p>Detail berat: 9 kg + 250 gr</p> <p>Detail biaya: Rp. 90000 + Rp. 3750</p> <p>Total biaya: Rp. 93750</p>
3	<p>Contoh #3</p> <p>Berat parcel (gram): <u>11750</u></p> <p>Detail berat: 11 kg + 750 gr</p> <p>Detail biaya: Rp. 110000 + Rp. 3750</p> <p>Total biaya: Rp. 110000</p>

```

package main

import "fmt"

// Fungsi untuk menghitung biaya pengiriman
func hitungBiaya(beratGram int) (int, int, int) {
    // Konversi berat dari gram ke kg
    beratKg := beratGram / 1000
    sisaGram := beratGram % 1000

    // Hitung biaya pengiriman per kg
    biayaKg := beratKg * 10000

    // Hitung biaya pengiriman untuk sisa gram
    var biayaSisa int
    if sisaGram >= 500 {
        biayaSisa = sisaGram * 5
    } else if sisaGram > 0 && beratKg <= 10 {
        biayaSisa = sisaGram * 15
    } else {
        biayaSisa = 0
    }

    return biayaKg, biayaSisa, beratKg
}

func main() {
    var beratGram int
    fmt.Print("Masukkan berat parcel (dalam gram): ")
    fmt.Scanln(&beratGram)

    // Panggil fungsi hitungBiaya untuk mendapatkan biaya pengiriman
    biayaKg, biayaSisa, beratKg := hitungBiaya(beratGram)

    // Tampilkan hasil perhitungan
    fmt.Printf("Total berat: %d kg %d gram\n", beratKg, beratGram%1000)
    fmt.Printf("Biaya pengiriman: Rp. %d\n", biayaKg+biayaSisa)
    fmt.Printf(" - Biaya per kg: Rp. %d\n", biayaKg)
    fmt.Printf(" - Biaya sisa gram: Rp. %d\n", biayaSisa)
}

```

Screenshoot Program

```

PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\unguided3.go"
Nilai K = 7
Nilai f(K) = 1.0011123471
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\unguided4.go"
Masukkan berat parcel (dalam gram): 45
Total berat: 0 kg 45 gram
Biaya pengiriman: Rp. 675
 - Biaya per kg: Rp. 0
 - Biaya sisa gram: Rp. 675
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> 

```

Deskripsi dan Cara Kerja Program

Program di atas adalah program Go yang menghitung biaya pengiriman berdasarkan berat paket dalam gram. Pertama, berat dalam gram dimasukkan oleh pengguna, lalu dikonversi menjadi kilogram (kg) dengan menghitung pembagian dan sisa gram. Biaya pengiriman dihitung berdasarkan berat kilogram dengan tarif Rp. 10.000 per kilogram, dan untuk sisa gram, jika lebih dari atau sama dengan 500 gram dikenakan biaya

tambahan sebesar 5 Rupiah per gram, sedangkan untuk sisa kurang dari 500 gram yang berat totalnya tidak lebih dari 10 kg, biaya dikenakan 15 Rupiah per gram. Fungsi `hitungBiaya` mengembalikan biaya pengiriman untuk kilogram dan sisa gram. Program kemudian menampilkan berat total paket dalam kg dan gram, serta biaya pengiriman lengkap dengan rincian biaya per kilogram dan biaya untuk sisa gram.

5. Buatlah program yang menerima input sebuah bilangan bulat b dan $b > 1$. Program harus dapat mencari dan menampilkan semua faktor dari bilangan tersebut! 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! Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bilangan: <u>12</u>	Bilangan: <u>7</u>
Faktor: 1 2 3 4 6 12	Faktor: 1 7

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

Bilangan: <u>12</u>	Bilangan: <u>7</u>
Faktor: 1 2 3 4 6 12	Faktor: 1 7
Prima: false	Prima: true

```
package main

import (
    "fmt"
)

func main() {
    // Deklarasi variabel untuk menyimpan input bilangan
    var b int

    // Meminta pengguna untuk memasukkan bilangan
    fmt.Print("Bilangan: ")
    fmt.Scanln(&b)

    // Mencetak faktor-faktor dari bilangan
    fmt.Print("Faktor: ")
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Print(i, " ")
        }
    }
}
```

```

    }
}

// Mengecek apakah bilangan adalah bilangan prima
prima := true
if b <= 1 {
    prima = false
} else {
    // Loop untuk mencari pembagi selain 1 dan b
    for i := 2; i*i <= b; i++ {
        if b%i == 0 {
            prima = false
            break
        }
    }
}

// Menampilkan apakah bilangan prima atau bukan
fmt.Println("\nPrima:", prima)
}

```

Screenshoot Program

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\unguided3.go"
Nilai K = 7
Nilai f(K) = 1.0011123471
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\unguided4.go"
Masukkan berat parcel (dalam gram): 45
Total berat: 0 kg 45 gram
Biaya pengiriman: Rp. 675
- Biaya per kg: Rp. 0
- Biaya sisa gram: Rp. 675
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2\unguided5.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 2>

```

Deskripsi dan Cara Kerja Program

Program di atas merupakan program sederhana dalam bahasa Go yang meminta pengguna untuk memasukkan sebuah bilangan bulat, kemudian menampilkan faktor-faktor dari bilangan tersebut serta mengecek apakah bilangan tersebut adalah bilangan prima. Pertama, pengguna diminta memasukkan bilangan, lalu program menggunakan loop `for` untuk mencari dan mencetak faktor-faktornya. Untuk mengecek apakah bilangan prima, program mendefinisikan variabel `prima` sebagai `true`, lalu melakukan pengecekan dengan membagi bilangan tersebut dengan angka-angka dari 2 hingga akar dari bilangan tersebut. Jika ditemukan pembagi selain 1 dan bilangan itu sendiri, maka variabel `prima` diubah menjadi `false`. Terakhir, program mencetak apakah bilangan tersebut adalah bilangan prima atau bukan.

KESIMPULAN

Laporan praktikum ini telah berhasil mengimplementasikan berbagai konsep struktur kontrol dalam bahasa pemrograman Go. Melalui serangkaian program yang disusun, mulai dari program sederhana seperti menampilkan teks hingga program yang lebih kompleks seperti menghitung biaya pengiriman dan menentukan bilangan prima, pemahaman mengenai struktur perulangan, percabangan, dan pengambilan keputusan dalam Go semakin terasah. Laporan ini juga menunjukkan bagaimana Go dapat digunakan untuk menyelesaikan berbagai macam masalah pemrograman, mulai dari masalah sederhana hingga masalah yang lebih kompleks. Dengan demikian, laporan ini dapat menjadi referensi yang baik bagi pembelajar Go untuk memperdalam pemahaman mereka tentang struktur kontrol dalam bahasa pemrograman ini.

Secara keseluruhan, laporan ini telah mencapai tujuannya yaitu:

1. Menguasai struktur kontrol dasar di Go: Laporan ini secara komprehensif mencakup berbagai jenis struktur kontrol seperti perulangan (for), percabangan (if-else, switch), dan pengambilan keputusan.
2. Menerapkan konsep pada kasus nyata: Setiap program yang dibuat dalam laporan ini memiliki relevansi dengan masalah dunia nyata, sehingga memperkuat pemahaman tentang penerapan konsep pemrograman.
3. Meningkatkan kemampuan pemecahan masalah: Melalui penyelesaian berbagai soal pemrograman, kemampuan dalam menganalisis masalah dan merancang algoritma semakin terasah.

Saran untuk perbaikan:

1. Dokumentasi yang lebih detail: Penjelasan mengenai algoritma dan logika program dapat diperdalam untuk memberikan pemahaman yang lebih baik bagi pembaca.
2. Penggunaan fungsi: Penggunaan fungsi secara lebih konsisten dapat meningkatkan modularitas dan readability kode.
3. Pengujian yang lebih menyeluruh: Melakukan pengujian terhadap berbagai kasus input untuk memastikan program berjalan dengan benar.
4. Dengan perbaikan-perbaikan tersebut, laporan ini dapat menjadi sumber belajar yang lebih baik bagi pembelajar Go.

DAFTAR PUSTAKA

Modul 2 Praktikum Algoritma dan Pemrograman 2