

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2  
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**ANDIKA NEVIANTORO / 2311102167**

**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 Dasar Program Golang

Secara umum, struktur dasar sebuah program Go terdiri dari:

- **package:** Menentukan paket dimana kode berada. Setiap file Go harus berada dalam sebuah paket. Paket main adalah titik masuk eksekusi program.
- **import:** Mengimpor paket-paket lain yang diperlukan.
- **fungsi main:** Titik awal eksekusi program.

Contoh sederhana :

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, World!")
}
```

### Tipe Data

Go mengenal beberapa jenis tipe data, di antaranya adalah tipe data numerik (desimal & non-desimal), string, dan boolean.

Pada pembahasan-pembahasan sebelumnya secara tak sadar kita sudah mengaplikasikan beberapa tipe data, diantaranya ada string dan tipe numerik int.

Pada chapter ini akan dijelaskan beberapa macam tipe data standar yang disediakan oleh Go, disertai juga contohnya.

#### a. Tipe data Numerik Non - Desimal

Tipe data numerik non-desimal atau **non floating point** di Go ada beberapa jenis. Secara umum ada 2 tipe data kategori ini yang perlu diketahui.

- **uint**, tipe data untuk bilangan cacah (bilangan positif).
- **int**, tipe data untuk bilangan bulat (bilangan negatif dan positif).

uint8	0 ↔ 255
uint16	0 ↔ 65535
uint32	0 ↔ 4294967295
uint64	0 ↔ 18446744073709551615
uint	sama dengan uint32 atau uint64 (tergantung nilai)
byte	sama dengan uint8
int8	-128 ↔ 127
int16	-32768 ↔ 32767
int32	-2147483648 ↔ 2147483647
int64	-9223372036854775808 ↔ 9223372036854775807
int	sama dengan int32 atau int64 (tergantung nilai)

### b. Tipe data Numerik Desimal

Tipe data numerik desimal yang perlu diketahui ada 2, float32 dan float64. Perbedaan kedua tipe data tersebut berada di lebar cakupan nilai desimal yang bisa ditampung. Untuk lebih jelasnya bisa merujuk ke spesifikasi IEEE-754 32-bit floating-point numbers.

```
var decimalNumber = 2.62

fmt.Printf("bilangan desimal: %f\n", decimalNumber)
fmt.Printf("bilangan desimal: %.3f\n", decimalNumber)
```

### c. Tipe data Boolean

Tipe data bool berisikan hanya 2 variansi nilai, true dan false. Tipe data ini biasa dimanfaatkan dalam seleksi kondisi dan perulangan.

```
var exist bool = true

fmt.Printf("exist? %t \n", exist)
```

### d. Tipe data String

Ciri khas dari tipe data string adalah nilainya di apit oleh tanda *quote* atau petik dua ("). Contoh penerapannya:

```
var message string = "Halo"  
fmt.Printf("message: %s \n", message)
```

Selain menggunakan tanda quote, deklarasi string juga bisa dengan tanda *grave accent/backticks* (```), tanda ini terletak di sebelah kiri tombol 1. Keistimewaan string yang dideklarasikan menggunakan backticks adalah membuat semua karakter di dalamnya **tidak di escape**, termasuk `\n`, tanda petik dua dan tanda petik satu, baris baru, dan lainnya. Semua akan terdeteksi sebagai string.

```
var message = `Nama saya "John Wick".  
Salam kenal.  
Mari belajar "Golang".`  
  
fmt.Println(message)
```

#### e. Nilai *Nil* & *Zero Value*

nil bukan merupakan tipe data, melainkan sebuah nilai. Variabel yang isi nilainya nil berarti memiliki nilai kosong.

Semua tipe data yang sudah dibahas di atas memiliki zero value (nilai default tipe data). Artinya meskipun variabel dideklarasikan dengan tanpa nilai awal, tetap akan ada nilai default-nya.

- Zero value dari string adalah `""` (string kosong).
- Zero value dari bool adalah `false`.
- Zero value dari tipe numerik non-desimal adalah `0`.
- Zero value dari tipe numerik desimal adalah `0.0`.

Selain tipe data yang disebutkan di atas, ada juga tipe data lain yang zero value-nya adalah nil. Nil merepresentasikan nilai kosong, benar-benar kosong. nil tidak bisa digunakan pada tipe data yang sudah dibahas di atas.

Beberapa tipe data yang bisa di-set nilainya dengan nil, di antaranya:

- pointer
- tipe data fungsi
- slice
- map
- channel
- interface kosong atau any (yang merupakan alias dari interface{})

Nantinya kita akan sering bertemu dengan nilai nil setelah masuk pada pembahasan-pembahasan tersebut.

## II. GUIDED

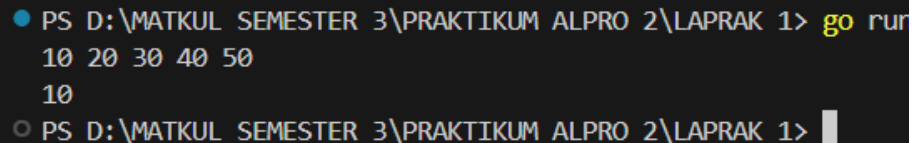
### 1. Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var nama string
    fmt.Scanln(&nama)
    fmt.Println(nama)
}
```

#### Screenshoot Output :



```
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go run 10 20 30 40 50
10
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1>
```

#### Deskripsi Program :

Program ini digunakan untuk menerima input berupa teks dari pengguna, kemudian menampilkannya kembali di layar. Tujuannya adalah untuk mengambil data yang dimasukkan oleh pengguna dan memverifikasinya dengan menampilkannya sebagai output. Ini adalah contoh sederhana dari interaksi input/output dasar di bahasa pemrograman Go.

2. 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 berturutan 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.

#### 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:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go run "d:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\GUIDED\guided2.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:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go run "d:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\GUIDED\guided2.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: ungu kuning hijau merah
BERHASIL : false
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> |
```

## Deskripsi Program :

Program di atas bertujuan untuk memeriksa apakah pengguna dapat memasukkan urutan warna yang benar sebanyak lima kali berturut-turut. Warna yang harus dimasukkan dalam urutan yang benar adalah: merah, kuning, hijau, dan ungu. Program menggunakan paket `bufio` untuk membaca input dari pengguna, di mana setiap input dipisahkan oleh spasi dan dibandingkan dengan urutan warna yang benar. Dalam setiap percobaan, jika pengguna memasukkan urutan warna yang salah, program langsung keluar dari loop dan mencetak "BERHASIL : false". Jika pengguna berhasil mengikuti urutan warna dengan benar dalam semua lima percobaan, program mencetak "BERHASIL : true". Tujuannya adalah untuk menguji ketepatan pengguna dalam mengikuti urutan warna yang diberikan.

### 3. 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 :

```
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go run "d:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\main.go"
5 10 15 20 25
Hasil Penjumlahan 5 10 15 20 25 adalah = 75
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1>
```

### Deskripsi Program :

Program di atas adalah program sederhana untuk menjumlahkan lima bilangan bulat yang diinputkan oleh pengguna. Program ini mendeklarasikan lima variabel 'a', 'b', 'c', 'd', dan 'e' bertipe 'int' untuk menampung input, serta satu variabel 'hasil' untuk menyimpan hasil penjumlahannya. Setelah itu, program menggunakan 'fmt.Scanln' untuk membaca kelima bilangan yang dimasukkan oleh pengguna. Setelah input diterima, kelima bilangan tersebut dijumlahkan dan hasilnya disimpan dalam variabel 'hasil'. Akhirnya, program mencetak hasil penjumlahan beserta kelima bilangan yang dimasukkan. Tujuan dari program ini adalah untuk menghitung dan menampilkan hasil penjumlahan dari lima bilangan bulat.

4. 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("Masukan nilai :")
    fmt.Scan(&nam)

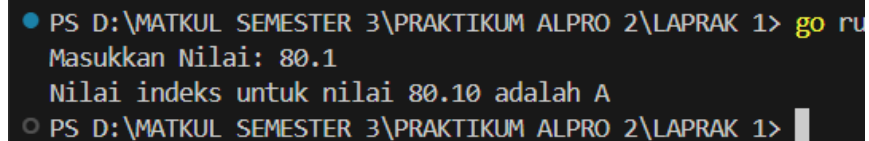
    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "B"
    } else if nam > 65 {
        nmk = "C"
    } else if nam > 50 {
        nmk = "D"
    } else if nam > 40 {
        nmk = "E"
    } else {
        nmk = "F"
    }

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

```

Jawablah pertanyaan-pertanyaan berikut:

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



```

PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go run
Masukkan Nilai: 80.1
Nilai indeks untuk nilai 80.10 adalah A
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1>

```

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

**Masalah Batasan Nilai:** Ada kesalahan dalam penentuan rentang nilai, seperti kondisi `nam > 80` yang menyebabkan nilai 80.0 tidak masuk kategori "A". Solusinya, gunakan operator `>=` agar batas bawah tercakup dengan tepat, misalnya nilai 80.0 harus termasuk kategori "A".

**Penggunaan Float32:** Float32 tidak diperlukan, lebih baik menggunakan float64 karena lebih umum dan lebih presisi.

**Alur Program Seharusnya:** Nilai nam harus diproses dengan rentang yang tepat, di mana batas bawah dari tiap kategori diikuti dengan `>=`. Contohnya, nilai 80.0 termasuk dalam kategori "A". Alur seharusnya dimulai dari kondisi nilai tertinggi (A), kemudian memeriksa nilai berikutnya secara bertahap ke bawah hingga "F".

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

## Sourcecode

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string

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

    // Mengatur rentang nilai dengan operator >=
    untuk batas bawah
    if nam >= 80 {
        nmk = "A"
    } else if nam >= 70 {
        nmk = "B"
    } else if nam >= 60 {
        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)
}
```

## Sreenshoot Output :

```
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go run main.go
Masukkan Nilai: 93.5
Nilai indeks untuk nilai 93.50 adalah A
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go run main.go
Masukkan Nilai: 70.6
Nilai indeks untuk nilai 70.60 adalah B
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go run main.go
Masukkan Nilai: 49.5
Nilai indeks untuk nilai 49.50 adalah E
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1>
```

### Deskripsi Program :

Program di atas bertujuan untuk menentukan nilai indeks huruf berdasarkan nilai angka yang dimasukkan oleh pengguna. Program dimulai dengan mendeklarasikan variabel `nam` (nilai angka) bertipe `float32` untuk menerima input dari pengguna, dan `nmk` (nilai indeks huruf) bertipe string untuk menyimpan hasil. Pengguna diminta memasukkan nilai melalui `fmt.Scan`, lalu program memeriksa nilai tersebut menggunakan serangkaian kondisi `if-else`. Berdasarkan rentang nilai yang dimasukkan, program akan menentukan nilai huruf: `A` untuk nilai di atas 80, `B` untuk nilai di atas 70, dan seterusnya hingga `F` untuk nilai di bawah 40. Setelah itu, program mencetak nilai indeks huruf yang sesuai dengan format dua desimal untuk nilai angka. Tujuan dari program ini adalah untuk mengonversi nilai angka ke dalam bentuk nilai huruf.

## 2. 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-teratal-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

### Sourcecode

```
package main

import (
    "bufio"    // Paket untuk membaca input dari
    pengguna dengan lebih fleksibel
    "fmt"      // Paket untuk mencetak ke layar dan
    menerima input
    "os"       // Paket untuk menangani sistem operasi,
    seperti membaca dari input standar
    "strings"  // Paket untuk memanipulasi string
    (misalnya trim, split, dll)
)

func main() {
```

```

        // Membuat objek reader untuk membaca input dari
pengguna
        reader := bufio.NewReader(os.Stdin)

        // Variabel N akan menyimpan jumlah nama bunga yang
akan dimasukkan pengguna
        var N int
        fmt.Print("Masukkan jumlah nama bunga: ")
        fmt.Scanln(&N) // Membaca jumlah bunga dari
pengguna

        // Variabel pita akan menyimpan gabungan nama bunga
dengan separator
        var pita string

        // Loop untuk meminta input nama bunga sebanyak N
kali
        for i := 0; i < N; i++ {
            // Meminta pengguna memasukkan nama bunga ke-i
            fmt.Printf("Masukkan nama bunga ke-%d: ", i+1)
            bunga, _ := reader.ReadString('\n') // Membaca
input bunga dari pengguna
            bunga = strings.TrimSpace(bunga) //
Menghapus karakter newline/whitespace di awal/akhir
string

            // Jika bukan input pertama, tambahkan " - "
sebagai pemisah
            if i > 0 {
                pita += " - "
            }

            // Menambahkan nama bunga ke dalam pita
            pita += bunga
        }

        // Menampilkan hasil pita yang berisi nama-nama
bunga yang telah digabung
        fmt.Println("Pita:", pita)
    }

```

### Screenshoot Output :

```
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2B
Masukkan jumlah nama bunga: 3
Masukkan nama bunga ke-1: Kertas
Masukkan nama bunga ke-2: Mawar
Masukkan nama bunga ke-3: Tulip
Pita: Kertas - Mawar - Tulip
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2B
Masukkan jumlah nama bunga: 0
Pita:
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> |
```

### Deskripsi Program :

Program di atas bertujuan untuk menggabungkan sejumlah nama bunga yang dimasukkan oleh pengguna dan menampilkannya dalam satu baris dengan pemisah " - ". Program dimulai dengan meminta pengguna untuk memasukkan jumlah bunga yang akan dimasukkan ('N'). Kemudian, program meminta pengguna memasukkan nama bunga satu per satu melalui input, yang dibaca menggunakan objek 'bufio.NewReader'. Setiap nama bunga yang dimasukkan ditambahkan ke variabel 'pita', dan jika sudah ada bunga sebelumnya, program menambahkan pemisah " - ". Setelah semua nama bunga dimasukkan, hasil gabungan nama-nama bunga ditampilkan dalam format: "Pita: [nama-bunga1] - [nama-bunga2] - ...". Tujuan dari program ini adalah untuk menerima sejumlah nama bunga, menggabungkannya dengan format yang rapi, dan menampilkannya dalam satu baris.

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.

## Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    // Variabel untuk menyimpan berat belanjaan di
    // kantong kiri dan kanan
    var beratKiri, beratKanan float64
    var totalBerat float64

    // Loop yang berjalan terus menerus sampai ada
    // kondisi untuk menghentikan
    for {
        // Meminta pengguna memasukkan berat
        // belanjaan di kedua kantong
        fmt.Print("Masukkan berat belanjaan di kedua
        kantong: ")
        fmt.Scanln(&beratKiri, &beratKanan)

        // Jika salah satu berat negatif, proses
        // dihentikan
        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Proses selesai. Salah
            satu kantong memiliki berat negatif.")
            break
        }

        // Menghitung selisih antara berat kiri dan
        // kanan
        selisih := beratKiri - beratKanan
        if selisih < 0 {
            selisih = -selisih // Mengubah selisih
            // menjadi nilai positif (absolut)
        }

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

        // Menghitung total berat dari kedua kantong
        totalBerat = beratKiri + beratKanan
    }
}
```

```

        // Jika total berat melebihi 150 kg, proses
        dihentikan
        if totalBerat > 150 {
            fmt.Println("Proses selesai. Total
berat melebihi 150 kg.")
            break
        }
    }
}

```

### Screenshoot Output :

```

PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2B\
Masukkan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi oleng: false
Masukkan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi oleng: true
Masukkan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi oleng: false
Masukkan berat belanjaan di kedua kantong: 59.5 98.7
Sepeda motor pak Andi oleng: true
Proses selesai. Total berat melebihi 150 kg.
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1>

```

### Deskripsi Program :

Program di atas bertujuan untuk mengevaluasi kestabilan sepeda motor Pak Andi berdasarkan berat belanjaan di dua kantong (kiri dan kanan) serta memeriksa total berat agar tidak melebihi batas 150 kg. Alur program dimulai dengan meminta pengguna untuk memasukkan berat kedua kantong secara berulang. Setelah input diterima, program menghitung selisih berat antara kantong kiri dan kanan. Jika selisihnya 9 kg atau lebih, program menganggap bahwa sepeda motor Pak Andi oleng, dan memberikan pesan "true". Jika tidak oleng, program menampilkan "false". Selain itu, program juga menjumlahkan total berat kedua kantong. Jika total berat melebihi 150 kg atau salah satu berat bernilai negatif, program akan menghentikan proses dengan pesan yang sesuai. Tujuan dari program ini adalah untuk mensimulasikan kondisi stabilitas sepeda motor Pak Andi berdasarkan distribusi beban di kantong belanjannya dan memastikan beban total tidak melebihi kapasitas aman.

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 diatas.

$\sqrt{2}$  merupakan bilangan irasional. Meskipun demikian, nilai tersebut dapat dihipir 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 dibelakang koma.

**Sourcecode :**

```
package main

import (
    "fmt" // Mengimpor package fmt untuk menggunakan
    fungsi input/output
)

func main() {
    var K int // Mendeklarasikan variabel K untuk
    menyimpan input dari pengguna

    fmt.Print("Nilai K = ") // Mencetak pesan untuk
    meminta input nilai K
    fmt.Scan(&K) // Menerima input dari pengguna dan
    menyimpannya ke dalam variabel K

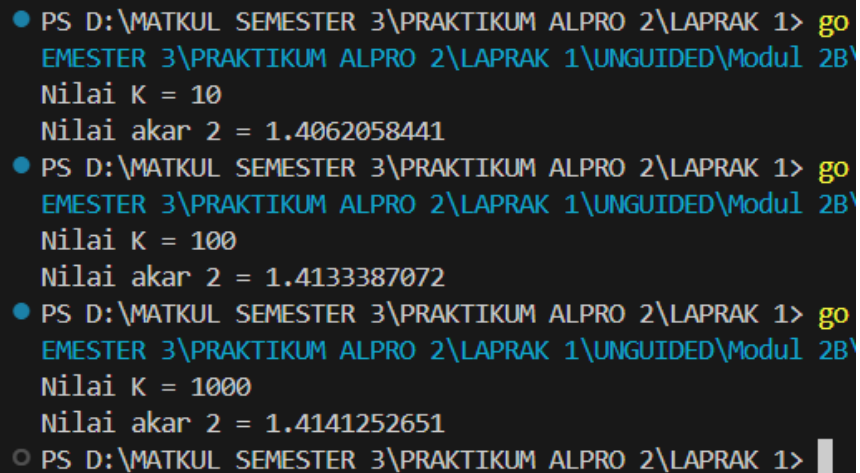
    sqrt2 := 1.0 // Inisialisasi variabel sqrt2 dengan
    nilai awal 1.0

    // Loop dari 0 hingga K (inklusif) untuk menghitung
    nilai akar 2
    for k := 0; k <= K; k++ {
        k := float64(k) // Mengonversi nilai k ke tipe
        float64 untuk perhitungan yang lebih presisi
        // Menghitung hasil dengan rumus tertentu
        hasil := (4*k + 2) * (4*k + 2) / ((4*k + 1) *
        (4*k + 3))
        sqrt2 *= hasil // Mengalikan nilai sqrt2 dengan
        hasil yang dihitung
    }

    // Mencetak hasil akhir dari nilai akar 2 dengan
    format 10 desimal
```

```
fmt.Printf("Nilai akar 2 = %.10f\n", sqrt2)
}
```

#### Screenshoot Output :



```
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2B
Nilai K = 10
Nilai akar 2 = 1.4062058441
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2B
Nilai K = 100
Nilai akar 2 = 1.4133387072
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2B
Nilai K = 1000
Nilai akar 2 = 1.4141252651
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1>
```

#### Deskripsi Program :

Program di atas menghitung nilai pendekatan dari akar 2 dengan menggunakan suatu formula rekursif. Alurnya dimulai dengan mendeklarasikan variabel `K` yang digunakan untuk menyimpan jumlah iterasi atau tingkat presisi yang diinginkan oleh pengguna. Setelah pengguna memasukkan nilai `K`, program menjalankan perhitungan dalam sebuah loop dari 0 hingga `K`. Pada setiap iterasi, program menghitung nilai tertentu berdasarkan rumus matematika yang mengandung variabel `k` yang dikonversi ke tipe `float64` untuk presisi yang lebih tinggi. Hasil dari perhitungan ini dikalikan secara berulang dengan variabel `sqrt2`, yang pada awalnya bernilai 1. Setelah semua iterasi selesai, program mencetak nilai pendekatan akar 2 hingga 10 angka desimal.

- PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program BlayaPos 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 Nibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

**Sourcecode :**

```
package main

import (
    "fmt" // Mengimpor package fmt untuk input/output
)

func main() {
    // Deklarasi variabel untuk menyimpan berat (gram),
    berat dalam kg, gram sisa, biaya per kg, total biaya,
    dan biaya tambahan
    var berat, kg, gram int
    var biayaPerKg, totalBiaya, biayaTambahan float64

    biayaPerKg = 10000.0 // Inisialisasi biaya per
    kilogram sebesar Rp. 10.000

    // Meminta input berat parcel dari pengguna dalam
    satuan gram
    fmt.Print("Berat parcel (gram): ")
    fmt.Scanln(&berat)

    // Menghitung berat dalam satuan kg dan gram sisa
    kg = berat / 1000 // Mengonversi berat dari
    gram ke kilogram
    gram = berat % 1000 // Mengambil sisa berat
    yang kurang dari 1 kg

    // Menghitung total biaya berdasarkan jumlah kg
    totalBiaya = float64(kg) * biayaPerKg // Total biaya
    untuk kilogram

    // Kondisi jika berat lebih dari 10 kg, biaya
    tambahan = 0
    if kg > 10 {
        biayaTambahan = 0 // Tidak ada biaya tambahan
        jika berat melebihi 10 kg
    } else if gram >= 500 {
        // Jika sisa berat >= 500 gram, biaya tambahan
        per gram adalah Rp. 5
        biayaTambahan = float64(gram) * 5
    } else {
        // Jika sisa berat < 500 gram, biaya tambahan
        per gram adalah Rp. 15
        biayaTambahan = float64(gram) * 15
    }

    // Menampilkan detail berat parcel dalam kg dan gram
```

```

        fmt.Printf("Detail berat: %d kg + %d gr\n", kg,
gram)

        // Menampilkan rincian biaya: total biaya untuk
kilogram + biaya tambahan berdasarkan gram
        fmt.Printf("Detail biaya: Rp. %.0f + Rp. %.0f\n",
totalBiaya, biayaTambahan)

        // Menambahkan biaya tambahan ke total biaya
totalBiaya += biayaTambahan

        // Menampilkan total biaya keseluruhan
fmt.Printf("Total biaya: Rp. %.0f\n", totalBiaya)
}

```

### Screenshoot Output :

```

● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go r
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2C\
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go r
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2C\
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go r
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2C\
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1>

```

### Deskripsi Program :

Program di atas bertujuan menghitung biaya pengiriman parcel berdasarkan beratnya. Pengguna memasukkan berat parcel dalam satuan gram, kemudian program mengonversinya menjadi kilogram dan gram sisa. Biaya pengiriman dasar dihitung berdasarkan kilogram dengan tarif Rp 10.000 per kg. Jika berat parcel kurang dari atau sama dengan 10 kg, tambahan biaya dihitung berdasarkan gram sisa. Jika sisa gram lebih dari atau sama dengan 500 gram, tarifnya Rp 5 per gram, sementara untuk gram di bawah 500 dikenakan Rp 15 per gram. Program kemudian menampilkan rincian berat dan total biaya pengiriman yang dihitung.

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 :

```
package main

import (
    "fmt" // Mengimpor package fmt untuk input/output
)

func main() {
    var b int // Mendeklarasikan variabel untuk
    menyimpan bilangan yang akan diperiksa

    // Meminta input bilangan dari pengguna
    fmt.Print("Bilangan: ")
    fmt.Scanln(&b)

    // Memeriksa apakah bilangan lebih kecil atau sama
    dengan 1
    if b <= 1 {
        // Jika bilangan <= 1, cetak pesan kesalahan dan
        hentikan program
        fmt.Println("Bilangan harus lebih besar dari 1")
        return // Keluar dari fungsi jika kondisi ini
        terpenuhi
    }

    // Menyiapkan untuk mencetak faktor dari bilangan
    fmt.Print("Faktor: ")

    // Inisialisasi variabel isPrima untuk menandai
    apakah bilangan adalah bilangan prima (default: true)
    isPrima := true
```

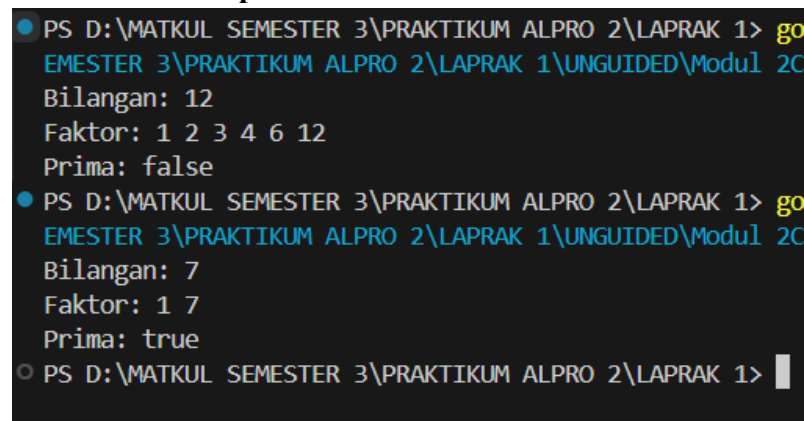
```

// Loop dari 1 hingga bilangan b untuk mencari faktor
for i := 1; i <= b; i++ {
    // Jika bilangan b habis dibagi i (faktor
ditemukan)
    if b%i == 0 {
        fmt.Print(i, " ") // Cetak faktor
        // Jika faktor bukan 1 atau bilangan itu
sendiri, maka b bukan bilangan prima
        if i != 1 && i != b {
            isPrima = false // Set isPrima ke false
jika b memiliki faktor lain selain 1 dan b
        }
    }
}
fmt.Println() // Baris baru setelah mencetak semua
faktor

// Mengecek apakah bilangan tersebut prima
berdasarkan nilai isPrima
if isPrima {
    fmt.Println("Prima: true") // Jika isPrima tetap
true, bilangan adalah prima
} else {
    fmt.Println("Prima: false") // Jika isPrima
berubah menjadi false, bilangan bukan prima
}
}

```

### Screenshoot Output :



```

PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2C
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1> go
EMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1\UNGUIDED\Modul 2C
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\LAPRAK 1>

```

### Deskripsi Program :

Program di atas bertujuan untuk menemukan faktor-faktor suatu bilangan dan memeriksa apakah bilangan tersebut merupakan bilangan prima. Setelah pengguna memasukkan bilangan bulat yang lebih besar dari 1, program mencari dan mencetak semua faktor dari bilangan tersebut. Jika bilangan hanya memiliki dua faktor, yaitu 1 dan bilangan

itu sendiri, maka bilangan tersebut dinyatakan sebagai bilangan prima. Jika ada faktor lain, bilangan tersebut bukan bilangan prima. Program juga menampilkan apakah bilangan tersebut adalah prima setelah mencetak faktor-faktornya.

## **KESIMPULAN**

struktur kontrol merupakan konsep yang sangat penting dalam pemrograman Go. Dengan menguasai struktur kontrol, kita dapat membangun program yang lebih baik, lebih efisien, dan lebih fleksibel.

Inisialisasi dan Penanganan Kesalahan: Dalam Go, baik if maupun switch dapat digunakan untuk menginisialisasi variabel lokal. Selain itu, keharusan menggunakan kurung kurawal dalam if membantu dalam menangani kesalahan dengan menghilangkan kebutuhan untuk kondisi else yang tidak diperlukan.

Pengulangan dan Tipe Dinamis: Pengulangan for memiliki tiga bentuk dan memanfaatkan klausa range untuk memudahkan iterasi pada struktur data, sementara switch dapat digunakan untuk menangani tipe dinamis dari variabel interface.

## **REFERENSI**

[1] Agung, N. dasar pemrograman golang :

<https://dasarpemrogramangolang.novalagung.com/A-tipe-data.html>