

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II**

**Review Struktur Kontrol**



**Disusun Oleh :**

**Erwin Rivaldo Silaban**

**2311102248**

**S1-IF-11-6**

**Dosen Pengampu :**

**Abednego Dwi Septiadi**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## **I. DASAR TEORI**

Proses pengembangan program Go melibatkan tiga tahap utama: koding, kompilasi, dan eksekusi. Koding dilakukan menggunakan editor teks atau Integrated Development Environment (IDE) yang mendukung Go. Setelah koding selesai, program Go dikompilasi menggunakan perintah 'go build' yang menghasilkan file executable. Proses kompilasi Go relatif cepat dan menghasilkan binary yang dapat langsung dijalankan tanpa dependensi eksternal. Eksekusi program dapat dilakukan langsung dari command line dengan menjalankan file executable yang dihasilkan, atau menggunakan perintah 'go run' yang mengkompilasi dan menjalankan program dalam satu langkah.

Go menyediakan beragam tipe data dan instruksi untuk manipulasi data. Tipe data dasar di Go meliputi integer, floating-point, complex numbers, boolean, dan string. Go juga mendukung tipe data komposit seperti array, slice, map, dan struct. Salah satu keunikan Go adalah adanya tipe data interface yang memungkinkan polimorfisme. Untuk manipulasi data, Go menyediakan berbagai instruksi dan operator, termasuk aritmatika, logika, dan bitwise. Go juga memiliki fitur unik seperti goroutines untuk konkurensi dan channels untuk komunikasi antar goroutine, yang memungkinkan pemrograman konkuren yang efisien dan mudah diimplementasikan.

### **Tipe Data dan Instruksi Data**

#### **Tipe Data Dasar:**

- int, int8, int16, int32, int64: Bilangan bulat
- uint, uint8, uint16, uint32, uint64: Bilangan bulat tak bertanda
- float32, float64: Bilangan desimal
- complex64, complex128: Bilangan kompleks
- bool: Nilai boolean (true/false)
- string: Rangkaian karakter

#### **Instruksi Dasar:**

- Percabangan: if, else, switch
- Perulangan: for
- Fungsi: func
- Defer: defer
- Goroutines: go
- Channels: ch := make(chan int)

## II. GUIDED

1. Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program silahkan masukkan data yang sesuai sebanyak yang di minta program. Perhatikan keluaran yang di peroleh. Coba terangkan apa sebenarnya yang di lakukan program tersebut.
- Source Code

```
package main

import "fmt"

func main() {
    var (
        satu, dua, tiga string
        temp           string
    )
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&tiga)
    fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)
    temp = satu
    satu = dua
    tiga = temp
    fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)
}
```

- Deskripsi Program
- Program akan meminta 3 inputan string dan kemudian menampilkan dua keluaran: keluaran awal sebelum terjadi perubahan dan keluaran akhir setelah pertukaran nilai antar variabel. Berikut adalah penjelasan langkah-langkah yang dilakukan oleh program:
- a) Program meminta pengguna untuk memasukkan tiga string dan menyimpannya dalam variabel `satu`, `dua`, dan `tiga`.
  - b) Program menampilkan nilai ketiga string tersebut dalam urutan awal (sebelum pertukaran) melalui perintah `fmt.Println`.
  - c) Selanjutnya, program melakukan pertukaran nilai sebagai berikut:
    - Nilai `satu` dipindahkan ke variabel sementara `temp`.
    - Nilai `dua` menggantikan nilai `satu`.
    - Nilai di `temp` (yaitu nilai asli dari `satu`) dipindahkan ke `tiga`.
  - d) Setelah pertukaran, program menampilkan keluaran akhir yang mencerminkan hasil dari perubahan tersebut.

Namun, ada yang aneh dalam proses pertukaran: hanya nilai `satu` dan `tiga` yang berubah, sementara nilai `dua` tetap tidak berubah sepanjang proses. Program tampaknya hanya mengganti `satu` dengan `dua`, dan memindahkan nilai awal `satu` ke `tiga`, tanpa memodifikasi variabel `dua`.

Misalnya, jika inputnya:

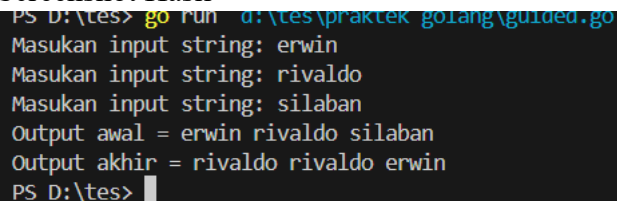
- `satu` = "erwin"
- `dua` = "rivaldo"
- `tiga` = "silaban"

Setelah pertukaran, hasil akhirnya adalah:

- `satu` = "rivaldo"
- `dua` = "rivaldo"
- `tiga` = "erwin"

Sehingga, variabel `dua` tetap, dan `satu` dan `tiga` bertukar nilai.

- Screenshot Hasil



```
PS D:\tes> go run d:\tes\praktek_golang\guided.go
Masukan input string: erwin
Masukan input string: rivaldo
Masukan input string: silaban
Output awal = erwin rivaldo silaban
Output akhir = rivaldo rivaldo erwin
PS D:\tes>
```

2. Tahun kabisat adalah tahun yang habis di bagi 400 atau habis di bagi 4 tetapi tidak habis di bagi 100 . buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat (true) atau bukan (false).

- Source Code

```
package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Masukkan sebuah tahun: ")
    fmt.Scanln(&tahun)

    if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100 != 0) {
        fmt.Println(tahun, "adalah tahun kabisat: true")
    } else {
        fmt.Println(tahun, "bukan tahun kabisat: false")
    }
}
```

- Deskripsi Program

Program akan menghitung Tahun yang habis dibagi 400 atau habis dibagi 4 tapi tidak habis dibagi 100 adalah tahun kabisat. Jika tahun memenuhi syarat tersebut, program akan mencetak bahwa tahun tersebut adalah tahun kabisat dan menampilkan "true". Jika tidak, program mencetak bahwa tahun tersebut bukan tahun kabisat dan menampilkan "false".

- Screenshot Hasil

```
PS D:\tes>
go run "d:\tes\praktek golang\akar.go"
Masukkan sebuah tahun: 440
440 adalah tahun kabisat: true
PS D:\tes>
```

3. Buat program bola yang menerima input jari-jari satu bola (bilangan bulat). Tampilkan volume dan luas kulit bola.

- Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var Jarijari float64
    fmt.Print("Masukkan jari-jari bola: ")
    fmt.Scanln(&Jarijari)
    volume := (4.0 / 3.0) * math.Pi * math.Pow(Jarijari, 3)
    Luaspermukaan := 4 * math.Pi * math.Pow(Jarijari, 2)
    fmt.Printf("Volume bola dengan jari-jari %.2f adalah %.2f\n",
        Jarijari, volume)
    fmt.Printf("Luas permukaan bola dengan jari-jari %.2f adalah
%.2f\n", Jarijari, Luaspermukaan)
}
```

- Deskripsi Program

Program ini dimulai dengan mengimpor paket fmt untuk input/output dan math untuk fungsi matematika, program meminta pengguna memasukkan nilai jari-jari yang disimpan dalam variabel bertipe float64. Selanjutnya, program menghitung volume bola menggunakan rumus  $\frac{4}{3}\pi r^3$  dan luas permukaan

menggunakan rumus  $4\pi r^2$ . Hasil perhitungan kemudian ditampilkan dengan format dua angka desimal menggunakan fungsi `fmt.Printf`

- Screenshot hasil

```
PS D:\tes> go run "d:\tes\praktek golang\akar.go"
Masukkan jari-jari bola: 12
Volume bola dengan jari-jari 12.00 adalah 7238.23
Luas permukaan bola dengan jari-jari 12.00 adalah 1809.56
PS D:\tes> █
```

### III. UNGUIDED

1. Dibaca nilai temperatur dalam nilai derajat celcius. Nyatakan tersebut dalam faranheit

$$Celsius = (Fahrenheit - 32) \times \frac{5}{9} \quad Reamur = Celsius \times \frac{4}{5} \quad Kelvin = (Fahrenheit + 459.67) \times \frac{5}{9}$$

(contoh input/output. Teks bergaris bawah adalah input dari user) :

Temperatur Celsius: 50  
Derajat Fahrenheit: 122

Lanjutkan program diatas sehingga tempertatur dinyatakan juga dalam derajat Reamur dan Kelvin.

(contoh input/ouput, Teks bergaris bawah adalah input dari user ) :

Temperatur Celsius: 50  
Derajat Reamur: 40  
Derajat Fahrenheit: 122  
Derajat Kelvin: 323

- Source code

```
package main

import (
    "fmt"
)

func main() {

    var celsius float64
    fmt.Println(" Masukkan Temperatur Celcius : ")
    fmt.Scan(&celsius)

    faranheit := (celsius * 9 / 5) + 22
    reamur := celsius * 4 / 5
    kelvin := celsius + 273.15

    fmt.Printf("Derajat Reamur : %.2f\n", reamur)
    fmt.Printf("Derajat Franheit : %.2f\n", faranheit)
    fmt.Printf("Derajat Kelvin : %.2f\n", kelvin)
}
```

- Screenshot output

```
PS D:\tes> go run "d:\tes\praktek golang\paraktek.go"
Masukkan Temperatur Celcius :
78
Derajat Reamur : 62.40
Derajat Franheit : 162.40
Derajat Kelvin : 351.15
PS D:\tes> go run "d:\tes\praktek golang\paraktek.go"
Masukkan Temperatur Celcius :
34
Derajat Reamur : 27.20
Derajat Franheit : 83.20
Derajat Kelvin : 307.15
PS D:\tes> █
```

- Deskripsi Program

Program bertujuan untuk mengkonversi suhu dari skala Celsius ke skala Fahrenheit, Reamur, dan Kelvin. Program dimulai dengan mengimpor package "fmt" yang digunakan untuk input/output standar. Dalam fungsi main, program meminta pengguna untuk memasukkan nilai suhu dalam Celsius melalui konsol. Setelah menerima input, program melakukan perhitungan konversi suhu menggunakan rumus-rumus yang telah ditentukan. Untuk Fahrenheit, digunakan rumus  $(\text{celsius} * 9/5) + 22$ , meskipun perlu dicatat bahwa rumus ini tidak standar dan mungkin mengandung kesalahan (biasanya digunakan  $+32$  bukan  $+22$ ). Untuk Reamur, digunakan rumus  $\text{celsius} * 4/5$ , dan untuk Kelvin,  $\text{celsius} + 273.15$ . Hasil konversi kemudian ditampilkan ke layar dengan format dua angka desimal menggunakan fungsi `fmt.Printf`.

2. Tipe karakter sebenarnya hanya apa yang tampak dalam tampilan. Di dalamnya tersimpan dalam bentuk biner 8 bit (byte) atau 32 bit (rune) saja.

Buat program ASCII yang akan membaca 5 buah data integer dan mencetaknya dalam format karakter. Kemudian membaca 3 buah data karakter dan mencetak 3 buah karakter setelah karakter tersebut (menurut tabel ASCII)

Masukan terdiri dari dua baris. Baris pertama berisi 5 buah data integer. Data integer mempunyai nilai antara 32 s.d. 127. Baris kedua berisi 3 buah karakter yang berdampingan satu dengan yang lain (tanpa dipisahkan spasi).

Keluaran juga terdiri dari dua baris. Baris pertama berisi 5 buah representasi karakter dari data yang diberikan, yang berdampingan satu dengan lain, tanpa dipisahkan spasi. Baris kedua berisi 3 buah karakter (juga tidak dipisahkan oleh spasi).

No.	Masukan	Keluaran
1	66 97 103 117 115 SNO	Bagus TOP

- Source code



```

package main

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

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

    fmt.Println("Masukkan 5 data integer (32-127) dipisahkan oleh spasi:")
    intLine, _ := reader.ReadString('\n')
    intStrs := strings.Fields(strings.TrimSpace(intLine))

    if len(intStrs) != 5 {
        fmt.Println("Error: Harap masukkan tepat 5 angka.")
        return
    }

    var intChars string
    for _, s := range intStrs {
        num, err := strconv.Atoi(s)
        if err != nil || num < 32 || num > 127 {
            fmt.Println("Error: Masukkan harus berupa angka antara 32 dan 127.")
            return
        }
        intChars += string(rune(num))
    }

    fmt.Println("Masukkan 3 karakter (tanpa spasi):")
    charLine, _ := reader.ReadString('\n')
    chars := strings.TrimSpace(charLine)

    if len(chars) != 3 {
        fmt.Println("Error: Harap masukkan tepat 3 karakter.")
        return
    }

    fmt.Println("Keluaran:")
    fmt.Println(intChars)

    var nextChars string
    for _, ch := range chars {

```

```

        nextChars += string(ch + 1)
    }
    fmt.Println(nextChars)
}

```

- Screenshot Hasil

```

PS D:\tes> go run "d:\praktek golang\tugas2.go"
Masukkan 5 data integer (32-127) dipisahkan oleh spasi:
45 67 56 78 100
Masukkan 3 karakter (tanpa spasi):
WIN
Keluaran:
-C8Nd
XJO
PS D:\tes> go run "d:\praktek golang\tugas2.go"
Masukkan 5 data integer (32-127) dipisahkan oleh spasi:
23 45 21 12 23
Error: Masukkan harus berupa angka antara 32 dan 127.
PS D:\tes> 

```

- Deskripsi Program

Alur program dimulai dengan meminta pengguna memasukkan 5 angka integer antara 32 dan 127 (yang merepresentasikan karakter ASCII yang dapat dicetak). Program kemudian memvalidasi input, memastikan ada tepat 5 angka dan semuanya dalam rentang yang benar. Jika valid, program mengonversi angka-angka tersebut menjadi karakter ASCII yang sesuai.

Selanjutnya, program meminta pengguna memasukkan 3 karakter tanpa spasi. Setelah memvalidasi bahwa tepat 3 karakter telah dimasukkan, program menampilkan dua baris output. Baris pertama menampilkan karakter ASCII yang dihasilkan dari 5 angka input pertama. Baris kedua menampilkan karakter yang mengikuti setiap karakter dari 3 karakter input kedua dalam urutan ASCII.

3. 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 warna sesuai dengan informasi yang urutan warna lainnya. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Percobaan 1:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 2:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 3:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 4:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 5:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
BERHASIL: true				

Percobaan 1:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 2:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 3:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 4:	<u>ungu</u>	<u>kuning</u>	<u>hijau</u>	<u>merah</u>
Percobaan 5:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
BERHASIL: false				

- Source Code

```
package main

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

func main() {
    reader := bufio.NewReader(os.Stdin)
    win := []string{"merah", "kuning", "hijau", "ungu"}
    berhasil := true

    for i := 1; i <= 5; i++ {
        fmt.Println("Warna : Merah, Kuning, Hijau, Ungu")
        fmt.Printf("Percobaan %d: ", i)
        input, _ := reader.ReadString('\n')
        warna := strings.Fields(strings.TrimSpace(input))

        if len(warna) != 4 {
            fmt.Println("Error: Masukkan 4 warna untuk setiap percobaan.")
            return
        }

        fmt.Printf("Percobaan %d: %s %s %s %s\n", i, warna[0], warna[1], warna[2], warna[3])

        for j := 0; j < 4; j++ {
            if warna[j] != win[j] {
                berhasil = false
            }
        }
    }
}
```

```

    }
}

fmt.Printf("BERHASIL: %v\n", berhasil)
}

```

- Screenshot Hasil

```

PS D:\tes> go run "d:\tes\praktek golang\paraktek.go"
Warna : Merah, Kuning, Hijau, Ungu
Percobaan 1: kuning hijau ungu merah
Percobaan 1: kuning hijau ungu merah
Warna : Merah, Kuning, Hijau, Ungu
Percobaan 2: hijau ungu merah kuning
Percobaan 2: hijau ungu merah kuning
Warna : Merah, Kuning, Hijau, Ungu
Percobaan 3: ungu hijau kuning merah
Percobaan 3: ungu hijau kuning merah
Warna : Merah, Kuning, Hijau, Ungu
Percobaan 4: merah ungu hijau kuning
Percobaan 4: merah ungu hijau kuning
Warna : Merah, Kuning, Hijau, Ungu
Percobaan 5: hijau ungu kuning merah
Percobaan 5: hijau ungu kuning merah
BERHASIL: false
PS D:\tes>
PS D:\tes>

```

- Deskripsi Hasil

Program di atas meminta pengguna untuk memasukkan 4 warna dalam 5 percobaan. Warna-warna yang harus ditebak adalah "merah", "kuning", "hijau", dan "ungu", yang disimpan dalam array win. Pada setiap percobaan, pengguna diminta untuk memasukkan 4 warna, dan program akan memeriksa apakah warna yang dimasukkan sesuai dengan urutan yang benar. Jika jumlah warna yang dimasukkan tidak tepat, program akan memberikan pesan error dan berhenti. Setelah 5 percobaan, program akan mencetak apakah pengguna berhasil atau tidak dalam menebak warna dengan benar.

4. Satu pita (string) berisi kumpulan nama nama bunga yang di pisahkan oleh spasi dan '-', contoh : pita di ilustrasikan seperti berikut Pita : melati - mawar - tulip - teratai - kamboja - matahari 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. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

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

Modifikasi Program sebelumnya, Proses Input akan berhenti bila user mengetikan 'SELESAI' kemudian tampilkan isi pita beserta banyaknya bunga yang ada dalam pita Perhatikan contoh sesi interaksi program seperti dibawah 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	

- Source code

```
package main

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

func main() {
    reader := bufio.NewReader(os.Stdin)
    var pita string
    var jumlahBunga int

    for {
        fmt.Print("Jumlah Bunga : ")
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

        if input == "SELESAI" {
            fmt.Println("Pita:", pita)
            fmt.Printf("Bunga: %d\n", jumlahBunga)
            return
        }

        n, err := strconv.Atoi(input)
        if err != nil || n <= 0 {
            fmt.Println("Masukkan Jumlah Bunga : ")
            continue
        }

        for i := 1; i <= n; i++ {
```

```

        fmt.Printf("Bunga %d: ", i)
        bunga, _ := reader.ReadString('\n')
        bunga = strings.TrimSpace(bunga)

        if bunga == "SELESAI" {
            fmt.Println("Pita:", pita)
            fmt.Printf("Bunga: %d\n", jumlahBunga)
            return
        }

        if pita == "" {
            pita = bunga
        } else {
            pita += " - " + bunga
        }
        jumlahBunga++
    }

    fmt.Println("Pita:", pita)
    fmt.Printf("Bunga: %d\n", jumlahBunga)
}

```

- Deskripsi Program

Program ini meminta pengguna untuk memasukkan jumlah bunga dan nama bunga secara berulang, hingga pengguna mengetikkan "SELESAI" untuk mengakhiri program. Setiap kali pengguna memasukkan jumlah bunga, program akan meminta nama-nama bunga sebanyak jumlah tersebut. Nama-nama bunga yang dimasukkan akan disimpan dalam variabel `pita` yang terus diperbarui dengan menambahkan nama bunga yang baru dengan pemisah " - ".

Program juga menghitung total bunga yang dimasukkan dalam variabel `jumlahBunga`. Jika input yang diberikan untuk jumlah bunga bukan angka positif, program akan meminta pengguna untuk memasukkan ulang jumlah yang benar. Ketika pengguna mengetikkan "SELESAI", program akan mencetak pita yang berisi daftar nama bunga yang telah dimasukkan dan jumlah total bunga, lalu keluar dari program.

- Screenshot Program

```
PS D:\tes> go run "d:\praktek golang\tugas2.go"
Jumlah Bunga : 7
Bunga 1: mawar merah
Bunga 2: mawar putih
Bunga 3: anggrek
Bunga 4: tulip
Bunga 5: edelwis
Bunga 6: SELESAI
Pita: mawar merah - mawar putih - anggrek - tulip - edelwis
Bunga: 5
PS D:\tes> █
```

5. 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.
```

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.
```

- Source Code

```
package main

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

func main() {
    reader := bufio.NewReader(os.Stdin)
    totalBerat := 0.0

    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

        weights := strings.Split(input, " ")
        if len(weights) != 2 {
            fmt.Println("Masukkan dua angka dipisahkan oleh spasi.")
            continue
        }

        weight1, err1 := strconv.ParseFloat(weights[0], 64)
        weight2, err2 := strconv.ParseFloat(weights[1], 64)

        if err1 != nil || err2 != nil || weight1 < 0 || weight2 < 0 {
            fmt.Println("Masukkan dua bilangan real positif.")
            continue
        }

        totalBerat += weight1 + weight2
        selisihBerat := math.Abs(weight1 - weight2)

        if weight1 >= 9 || weight2 >= 9 || totalBerat > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        if selisihBerat >= 9 {
            fmt.Println("Sepeda motor pak Andi akan oleng: true")
        } else {

```



```

        fmt.Println("Sepeda motor pak Andi akan oleng: false")
    }
}
}

```

- Deskripsi Program

Program ini meminta pengguna untuk memasukkan berat belanjaan di dua kantong dalam bentuk angka yang dipisahkan oleh spasi. Program memeriksa apakah input valid dan menghitung total berat kedua kantong serta selisih antara keduanya. Jika salah satu kantong beratnya lebih dari 9 kg atau total berat belanjaan melebihi 150 kg, program akan berhenti. Selain itu, program memeriksa apakah selisih berat antara kedua kantong lebih dari 9 kg, yang akan menyebabkan sepeda motor Pak Andi "oleng". Jika ya, program mencetak "true", jika tidak mencetak "false".

- Screenshot Hasil

```

PS D:\tes> go run "d:\tes\praktek golang\paraktek.go"
Masukan berat belanjaan di kedua kantong: 2 4
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.4
Proses selesai.
PS D:\tes> go run "d:\tes\praktek golang\paraktek.go"
Masukan berat belanjaan di kedua kantong: 

```

6. Diberikan persamaan sebagai berikut ini:

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

Buatlah sebuah program yang meminta input sebuah bilangan sebagai K, kemudian menghitung dan menampilkan nilai f(k) sesuai dengan persamaan diatas. Perhatikan contoh sesi interaksi program seperti dibawah ini (teks bergaris bawah adalah input /read):

Nilai K= 100 Nilai (fk) = 1.0000061880
---

akar 2 merupakan bilangan irasional. Meskipun demikian, nilai tersebut dihipotesiskan 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 akar2 untuk K tersebut. Hampiran akar 2 dituliskan dalam 10 angka di belakang koma. Perhatikan contoh sesi interaksi program seperti dibawah ini (teks bergaris bawah adalah input/ read) :

1	Nilai K= 10 Nilai akar 2 = 1.460205441
2	Nilai K = 100

	Nilai akar 2 = 1.4133387072
3	Nilai K = 1000 Nilai akar 2 = 1.1414252651

- Source Code

```

package main

import (
    "fmt"
    "math"
)

func f(k float64) float64 {
    numerator := math.Pow(4*k+2, 2)
    denominator := (4*k + 1) * (4*k + 3)
    return numerator / denominator
}

func approximateSqrt2(K int) float64 {
    result := 1.0
    for k := 0; k < K; k++ {
        result *= f(float64(k))
    }
    return result
}

func main() {
    var K int
    fmt.Print("Nilai K = ")
    fmt.Scan(&K)
    go func(go) {

        fK := f(float64(K))
        fmt.Printf("Nilai f(K) = %.10f\n", fK)

        sqrt2 := approximateSqrt2(K)
        fmt.Printf("Nilai akar 2 = %.10f\n", sqrt2)

    }()
}

```

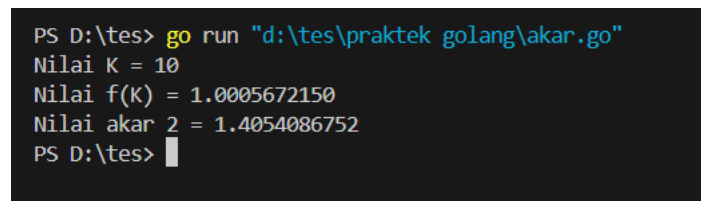
- Deskripsi Program

Program ini bertujuan untuk mendekati nilai akar kuadrat dari 2 menggunakan sebuah fungsi matematika. Fungsi `f(k)` menghitung suatu ekspresi matematika berdasarkan nilai `k`, di mana hasilnya adalah pembagian antara hasil kuadrat dari ekspresi di

pembilang dan hasil perkalian ekspresi di penyebut. Fungsi `approximateSqrt2(K)` menghitung perkiraan akar 2 dengan mengalikan hasil dari fungsi `f(k)` secara berulang, dari `k = 0` hingga `k = K`. Pada bagian `main`, pengguna diminta untuk memasukkan nilai `K` sebagai batas iterasi. Program kemudian menghitung nilai dari fungsi `f(K)` dan menampilkan hasilnya dengan format desimal hingga 10 tempat. Setelah itu, program menghitung dan menampilkan nilai perkiraan akar 2 berdasarkan iterasi fungsi `approximateSqrt2` sebanyak `K` kali.

Intinya, program ini menggunakan metode perkalian fungsi matematika untuk mendekati nilai akar kuadrat dari 2 sesuai dengan jumlah iterasi yang ditentukan oleh pengguna.

- Screenshot Hasil



```
PS D:\tes> go run "d:\tes\praktek\golang\akar.go"
Nilai K = 10
Nilai f(K) = 1.0005672150
Nilai akar 2 = 1.4054086752
PS D:\tes> 
```