

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

Hamzah Ziyad Ibadurrohman / 2311102254

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Koding, Kompilasi, dan Eksekusi Go

Koding

Tidak berbeda dengan penulisan program sumber dalam bahasa lain, program Go harus dibuat menggunakan penyunting teks dan disimpan dalam format teks, bukan dalam format dokumen (doc, docx, atau lainnya).

Setiap program go disimpan dalam file teks dengan ekstensi *.go, dengan nama bebas. Sebaiknya nama file adalah nama untuk program tersebut.

Setiap satu program lengkap Go disimpan dalam satu folder tersendiri. Nama folder merupakan nama program tersebut. Karena itu secara prinsip, satu program Go dapat dipecah dalam beberapa file dengan ekstensi *.go selama disimpan dalam folder yang sama.

Kompilasi

Beberapa bahasa pemrograman dirancang untuk diimplementasikan sebagai interpreter dan lainnya sebagai kompilator. Interpreter akan membaca setiap baris instruksi dan kemudian langsung mengeksekusinya, dengan hanya sedikit pemeriksaan apakah penulisan keseluruhan program sudah benar atau belum. Kompilator akan memeriksa keseluruhan program sumber dan kemudian mengubahnya menjadi program eksekutabel, sehingga konsistensi penulisan informatics 100 (seperti penggunaan tipe data) sudah diperiksa sebelum eksekusi. Selain itu karena program dibuat menjadi eksekutabel lebih dahulu, proses optimasi dapat dilakukan sehingga program menjadi sangat efisien.

Go diimplementasikan sebagai kompilator. Berikut adalah contoh sesi yang biasa dilakukan saat mengkompilasi dan mengeksekusi program dalam bahasa Go:

Panggil shell atau terminal (program/utiliti cmd.exe di Windows)

Masuk ke dalam (cd) folder program (normalnya ada di C:\Users\go\src\ atau yang sejenis)

Kemudian panggil perintah go build atau go build file.go untuk mengkompilasi file.go

Jika gagal, akan muncul pesan eror yang sesuai, pelajari dengan baik pesan tersebut, perbaiki teks program sumber, kemudian ulangi proses build-nya.

Jika berhasil maka pada folder tersebut akan dibuat program dengan nama yang sama dan diakhiri dengan .exe (untuk Windows)

Panggil program eksekutabel tersebut dari terminal yang sama. Jangan memanggil program tersebut dengan mengklik eksekutabel tersebut dari folder A karena program kalian hanya berbasis teks, bukan/belum dirancang dengan tampilan Windows.

Catatan

Semua proses terkait bahasa Go dilakukan melalui utilitas go. Beberapa opsi dengan utilitas go:

- go build: mengkompilasi program sumber yang ada dalam folder menjadi sebuah program.
- go build file.go: mengkompilasi program sumber file.go saja.
- go fmt: membaca semua program sumber dalam folder dan mereformat penulisannya agar sesuai dengan standar penulisan program sumber Go.
- go clean: membersihkan file-file dalam folder sehingga tersisa program sumber nya saja.

Data dan Variabel

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

Nama variabel dimulai dengan huruf dan dapat diikuti dengan sejumlah huruf, angka, atau garisbawah.

Contoh: ketemu, found, rerata, mhs1, data_2, ...

Notasi tipe dasar	Tipe dalam Go	Keterangan
integer	int int8 int32 //rune int64 uint uint8 //byte uint32 uint64	bergantung platform 8 bit: -128..127 32 bit: -10 ⁹ ..10 ⁹ 64 bit: -10 ¹⁹ ..10 ¹⁹ bergantung platform 0..255 0..4294967295 0..(2 ⁶⁴ -1)
real	float32 float64	32bit: -3.4E+38 .. 3.4E+38 64bit: -1.7E+308 .. 1.7E+308
boolean (atau logikal)	bool	false dan true
karakter	byte //uint8 rune //int32	tabel ASCII/UTF-8 tabel UTF-16
string	string	

Operator dalam Go	Tipe data terkait	Keterangan
+	string integer dan real	konkatenasi 2 string operasi penjumlahan
- * /	integer dan real	operasi pengurangan, perkalian, dan pembagian
%	integer	operasi sisa pembagian integer (modulo)
& ^ &^	integer	operasi per-bit AND, OR, XOR, AND-NOT
<< >>	integer dan unsigned integer	operasi geser bit kiri/kanan sebanyak unsigned integer yang diberikan
< <= > >= == !=	selain boolean	komparasi menghasilkan nilai boolean komparasi karakter sesuai dengan posisi karakter tersebut dalam tabel ASCII/UTF-16 komparasi string sesuai dengan operasi karakter per karakter, dimulai dari karakter paling kiri (awal)
&& !	boolean	operasi boolean AND, OR, dan NOT
* &	variabel apasaja	mendapatkan data dari lokasi memori dan mendapatkan lokasi dari variabel

II. GUIDED

1. Soal Studi Case

Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

Sourcecode

```
package main

import "fmt"

func main() {
    var (
        satu, dua, tiga string
        temp          string
    )
    fmt.Println("Masukkan Input string: ")
    fmt.Scanln(&satu)
    fmt.Println("Masukkan Input string: ")
    fmt.Scanln(&dua)
    fmt.Println("Masukkan Input string: ")
    fmt.Scanln(&tiga)
    fmt.Println("output awal =" + satu + " " + dua + " "
+ tiga)
    temp = satu
    satu = dua
    dua = tiga
    tiga = temp
    fmt.Println("output akhir =" + satu + " " + dua + " "
+ tiga)
}
```

Screenshoot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code> go run hello.go
Masukkan Input string:
Hamzah
Masukkan Input string:
Ziyad
Masukkan Input string:
Ibadurrohman
output awal =Hamzah Ziyad Ibadurrohman
output akhir =Ziyad Ibadurrohman Hamzah
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code>
```

Deskripsi Program

Program Golang ini berfungsi untuk menerima tiga input string dari pengguna, menukar nilai-nilai tersebut, dan menampilkan hasilnya sebelum dan sesudah pertukaran. Program ini dimulai dengan mendeklarasikan tiga variabel (satu, dua, dan tiga) untuk menyimpan input string, serta satu variabel sementara (temp) yang digunakan untuk membantu proses pertukaran nilai. Setelah meminta pengguna untuk memasukkan tiga string dengan menggunakan `fmt.Scanln()`, program menampilkan input awal dalam format output awal = satu dua tiga. Selanjutnya, program menggunakan temp untuk menukar nilai ketiga variabel tersebut. Setelah proses pertukaran selesai, hasil akhir ditampilkan dalam format output akhir = satu dua tiga.

2. Soal Studi Case

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

Sourcecode

```
package main

import (
    "fmt"
)

func cekTahunKabisat() {
    var tahun int


    fmt.Print("tahun: ")
```

```
        fmt.Scanln(&tahun)

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

func main() {
    cekTahunKabisat()
}
```

Screenshot Output



```
tahun: 2016
kabisat: true
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code>
```

Deskripsi Program

Kode Golang ini bertujuan untuk memeriksa apakah tahun yang dimasukkan oleh pengguna merupakan tahun kabisat atau tidak. Di fungsi `cekTahunKabisat()` tidak menerima parameter dan tidak mengembalikan nilai, berfungsi untuk mengecek tahun kabisat. Di dalam fungsi ini, variabel `tahun` bertipe `int` digunakan untuk menyimpan input tahun dari pengguna. Program meminta pengguna untuk memasukkan tahun, dan menyimpan input tersebut di variabel `tahun` menggunakan `fmt.Scanln()`. Selanjutnya, program menggunakan struktur `if` untuk memeriksa apakah tahun yang diberikan adalah kabisat. Aturan untuk menentukan tahun kabisat adalah tahun dianggap kabisat jika dapat dibagi 400, atau dapat dibagi 4 tetapi tidak dapat dibagi 100. Jika salah satu dari kondisi tersebut terpenuhi, program akan mencetak `kabisat: true`, dan jika tidak, akan mencetak `kabisat: false`. Fungsi `main()` merupakan titik awal dari eksekusi program, di mana program

memanggil fungsi cekTahunKabisat() untuk menjalankan logika pemeriksaan tahun kabisat.

3. Soal Studi Case

Buat program Bola yang menerima input jari-jari suatu bola (bilangan bulat). Tampilkan Volume dan Luas kulit bola.
 $\text{volumebola} = \frac{4}{3}\pi r^3$ dan $\text{luasbola} = 4\pi r^2$ ($\pi = 3.1415926535$).

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func hitungluasvolume() {
    var jarijari float64

    fmt.Print("Masukkan jari-jari bola :")
    fmt.Scanln(&jarijari)

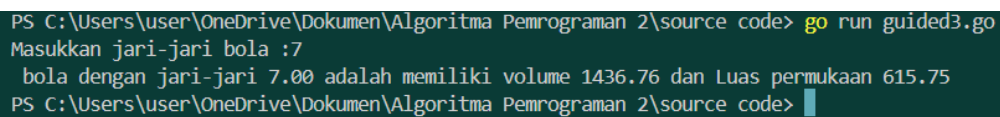
    volume := (4.0 / 3.0) * math.Pi * math.Pow(jarijari,
3)

    luasPermukaan := 4 * math.Pi * math.Pow(jarijari, 2)
```



```
        fmt.Printf(" bola dengan jari-jari %.2f adalah  
memiliki volume %.2f dan Luas permukaan %.2f\n", jarijari,  
volume, luasPermukaan)  
  
    }  
  
    func main() {  
        hitungluasvolume()  
    }
```

Screenshot Output



```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code> go run guided3.go  
Masukkan jari-jari bola :7  
bola dengan jari-jari 7.00 adalah memiliki volume 1436.76 dan Luas permukaan 615.75  
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code>
```

Deskripsi Program

Kode yang diberikan berfungsi untuk menghitung volume dan luas permukaan bola berdasarkan jari-jari yang dimasukkan oleh pengguna. Fungsi `hitungluasvolume` mendeklarasikan variabel `jarijari` sebagai tipe data `float64` untuk menyimpan nilai jari-jari bola. Program kemudian meminta pengguna memasukkan jari-jari dan membaca input tersebut. Setelah itu, volume dihitung menggunakan rumus $\frac{4}{3} \cdot \pi \cdot r^3$ dan luas permukaan menggunakan rumus $4 \cdot \pi \cdot r^2$. Hasil perhitungan ditampilkan dengan format tertentu, di mana nilai ditampilkan dengan dua angka di belakang koma. Fungsi utama adalah titik masuk program yang memanggil fungsi `hitungluasvolume()`, menjalankan seluruh logika perhitungan. Ketika dijalankan, jika pengguna memasukkan 5 sebagai jari-jari, output yang dihasilkan akan menunjukkan informasi mengenai jari-jari, volume, dan luas permukaan bola dengan format yang jelas.

III. UNGUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

Soal Studi Case

Dibaca nilai temperatur dalam derajat Celsius. Nyatakan temperatur tersebut dalam Fahrenheit.

Lanjutkan program di atas, sehingga temperatur dinyatakan juga dalam derajat Reamur dan Kelvin

Sourcecode

```
package main

import "fmt"

func celsiuskefahrenheit(suhu float64) float64 {
    return (9.0 / 5.0 * suhu) + 32
}

func celsiuskekelvin(suhu float64) float64 {
    return suhu + 273.15
}

func celsiuskereamur(suhu float64) float64 {
    return 4.0 / 5.0 * suhu
}

func main() {

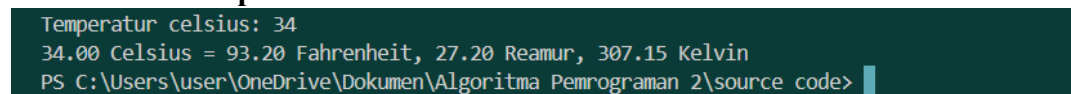
    var suhu float64
    fmt.Print("Temperatur celsius: ")
    fmt.Scanln(&suhu)

    fahrenheit := celsiuskefahrenheit(suhu)
    kelvin := celsiuskekelvin(suhu)
    reamur := celsiuskereamur(suhu)

    fmt.Println("Temperatur Celsius: %.0f\n", suhu)
    fmt.Println("Derajat Reamur: %.0f\n", reamur)
    fmt.Println("Derajat Fahrenheit: %.0f\n",
fahrenheit)
    fmt.Println("Derajat Kelvin: %.0f\n", kelvin)

}
```

Screenshoot Output



```
Temperatur celsius: 34
34.00 Celsius = 93.20 Fahrenheit, 27.20 Reamur, 307.15 Kelvin
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code>
```

Deskripsi Program

Kode golang ini berfungsi untuk mengonversi suhu dari derajat Celsius ke derajat Fahrenheit, Kelvin, dan Reamur. Terdapat tiga fungsi konversi: `celsiuskefahrenheit` yang mengonversi suhu Celsius ke Fahrenheit menggunakan rumus $9/5 * \text{suhu} + 32$; `celsiuskekelvin` yang mengonversi suhu Celsius ke Kelvin dengan rumus $\text{suhu} + 273.15$; dan `celsiuskereamur` yang mengonversi suhu Celsius ke Reamur dengan rumus $4/5 * \text{suhu}$. Kemudian fungsi `main` mendeklarasikan variabel suhu untuk menyimpan nilai suhu Celsius yang dimasukkan oleh pengguna. Program meminta pengguna untuk memasukkan suhu, kemudian membaca input tersebut dan memanggil ketiga fungsi konversi untuk menghitung nilai suhu dalam Fahrenheit, Kelvin, dan Reamur. Hasil konversi ditampilkan ke layar dengan menggunakan `fmt.Println`.

2. Soal Studi Case

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)

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var integers [5]int
    var chars [3]byte

    for i := 0; i < 5; i++ {
        fmt.Scanf("%d", &integers[i])
    }
}
```

```
        if integers[i] < 32 || integers[i] > 127 {  
            fmt.Println("Input integer harus di antara 32 dan  
127.")  
            return  
        }  
    }  
}  
  
fmt.Scanf("\n%c%c%c", &chars[0], &chars[1], &chars[2])  
  
fmt.Print("Karakter dari Integer: ")  
for i := 0; i < 5; i++ {  
    fmt.Printf("%c", integers[i])  
}  
fmt.Println()  
  
fmt.Print("Karakter: ")  
for i := 0; i < 3; i++ {  
    fmt.Printf("%c", chars[i])  
}  
fmt.Println()  
}
```

Screenshot output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code> go run unguided2.go
72 86 91 85 89
KCJ
Karakter dari Integer: HV[UY
Karakter: KCJ
```

Deskripsi Output

Program ini adalah program yang membaca input integer dan karakter. lalu mengonversi nilai integer menjadi karakter sesuai kode ASCII, Serta mencetak karakter yang sesuai dengan nilai integer yang dimasukkan. dalam fungsi main, dua array dideklarasikan: integers untuk menyimpan 5 bilangan bulat dan chars untuk menyimpan 3 karakter. Program kemudian melakukan iterasi menggunakan loop untuk membaca 5 data integer, di mana setiap input integer divalidasi untuk memastikan nilainya berada di antara 32 dan 127. Jika nilai tidak valid, program akan menampilkan pesan kesalahan dan menghentikan eksekusi. Setelah semua integer dibaca, program meminta pengguna untuk memasukkan 3 karakter. Program kemudian mencetak karakter yang sesuai dengan nilai integer yang dimasukkan dengan mengonversi nilai integer ke karakter ASCII. Selain itu, program juga mencetak ketiga karakter yang dibaca dari input.

3. Soal Studi Case

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.

Sourcecode

```
package main
```

```
import (  
    "fmt"  
    "strings"  
)  
  
func main() {  
    var warna1, warna2, warna3, warna4 string  
    var berhasil bool = true  
  
    for i := 1; i <= 5; i++ {  
        fmt.Printf("Percobaan %d: ", i)  
        fmt.Scanln(&warna1, &warna2, &warna3, &warna4)  
  
        if !strings.EqualFold(warna1, "merah") ||  
!strings.EqualFold(warna2, "kuning") ||  
!strings.EqualFold(warna3, "hijau") || !strings.EqualFold(warna4,  
"ungu") {  
            berhasil = false  
        }  
    }  
  
    fmt.Println("BERHASIL:", berhasil)  
}
```

Screenshot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code> go run unguided3.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 C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code>
```

Deskripsi Program

Ini adalah Program golang yang dibuat untuk menerima input warna selama lima percobaan dan memeriksa kesesuaian warna yang dimasukkan dengan urutan yang benar. Program dimulai dengan mendeklarasikan paket utama dan mengimpor paket fmt untuk keperluan input dan output, serta paket strings untuk manipulasi string. Dalam fungsi main, empat variabel bertipe string (warna1, warna2, warna3, warna4) didefinisikan untuk menyimpan input warna, sementara variabel berhasil diinisialisasi dengan nilai true, yang menunjukkan bahwa semua percobaan dianggap berhasil. Program kemudian menjalankan loop untuk meminta pengguna memasukkan empat warna dalam lima percobaan. Setiap input warna akan divalidasi untuk memastikan urutannya sesuai dengan yang diharapkan: "merah", "kuning", "hijau", dan "ungu". Apabila salah satu warna tidak sesuai, variabel berhasil akan diubah menjadi false. Setelah seluruh percobaan selesai, program mencetak hasil akhir.

4. Soal Studi Case

Suatu pita (string) berisi kumpulan nama-nama bunga yang dipisahkan oleh spasi dan '-', contoh pita diilustrasikan seperti berikut ini.

Plta: 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.

Sourcecode

```
package main

import "fmt"
```

```
func main() {  
  
    var N int  
  
    var pita string  
  
    var count int  
  
  
    fmt.Print("Masukkan jumlah bunga (input berhenti saat  
mengetik 'SELESAI'): ")  
  
    fmt.Scanln(&N)  
  
  
    for i := 0; i < N; i++ {  
  
        var namaBunga string  
  
        fmt.Printf("Masukkan nama bunga ke-%d: ", i+1)  
  
        fmt.Scanln(&namaBunga)  
  
  
  
        if namaBunga == "SELESAI" {  
  
            break  
  
        }  
  
  
  
        if count > 0 {  
  
            pita += " - "  
  
        }  
  
        pita += namaBunga  
  
        count++  
  
    }  
}
```



```
        fmt.Println("Pita:", pita)

        fmt.Println("Bunga:", count)
    }
}
```

Screenshot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code> go run unguided4.go
Masukkan jumlah bunga (input berhenti saat mengetik 'SELESAI'): 3
Masukkan nama bunga ke-1: SELESAI
Pita:
Bunga: 0
```

Deskripsi Program

Program golang ini diatas dirancang untuk menerima input nama bunga dari pengguna, menggabungkannya ke dalam string yang disebut "pita", dan menghitung jumlah bunga yang dimasukkan. Program dimulai dengan mendeklarasikan paket utama dan mengimpor paket fmt untuk input dan output. Dalam fungsi main, tiga variabel dideklarasikan: N untuk menyimpan jumlah bunga yang ingin dimasukkan, pita sebagai string untuk menggabungkan nama bunga, dan count untuk menghitung jumlah bunga yang dimasukkan. lalu Program meminta pengguna untuk memasukkan jumlah bunga yang ingin mereka masukkan, yang disimpan dalam variabel N. Kemudian, program menggunakan loop untuk meminta pengguna memasukkan nama bunga sebanyak N kali. setiap iterasi, program menampilkan prompt dan menyimpan input pengguna dalam variabel namaBunga. Jika pengguna mengetik "SELESAI", program akan keluar dari loop. Nama bunga yang valid kemudian ditambahkan ke string pita, dan variabel count diperbarui untuk mencatat jumlah bunga yang telah dimasukkan. Setelah semua input selesai, program mencetak string pita yang berisi semua nama bunga dan jumlah bunga yang telah dimasukkan.

5. Soal Studi Case

Setiap hari Pak Andi membawa banyak barang belanjaan dari pasar dengan mengendarai sepeda motor. Barang belanjaan tersebut dibawa dalam kantong terpal di kiri-kanan motor. Sepeda motor tidak akan oleng jika selisih berat barang di kedua kantong sisi tidak lebih dari 9 kg.

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

Pada modifikasi program tersebut, program akan menampilkan true jika selisih kedua 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.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var berat1, berat2, total float64

    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&berat1, &berat2)

        if berat1 < 0 || berat2 < 0 {
            fmt.Println("Proses selesai.")
            break
        }
    }
}
```

```
        selisih := math.Abs(berat1 - berat2)

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

        total = berat1 + berat2

        if total > 150 {
            fmt.Println("Proses selesai.")
            break
        }
    }
}
```

Screenshot Output

```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code> go run test.go
Masukan berat belanjaan di kedua kantong: 54 60
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 90 20
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 80 80
Sepeda motor pak Andi akan oleng: false
Proses selesai.
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code>

```

Deskripsi Program

Program golang ini dibuat untuk mengukur keseimbangan beban pada sepeda motor Pak Andi berdasarkan berat belanjaan di dua kantong, Dan menghentikan proses jika kondisi tertentu terpenuhi. Programnya dimulai dengan meminta input berat dua kantong dari pengguna dalam loop yang terus berjalan. Jika salah satu berat bernilai negatif, program akan berhenti dengan mencetak "Proses selesai." Program kemudian menghitung selisih berat antara kedua kantong, dan jika selisihnya lebih dari atau sama dengan 9 kg, program akan mencetak bahwa "Sepeda motor pak Andi akan oleng: true". Jika selisihnya kurang dari 9 kg, program mencetak "Sepeda motor pak Andi akan oleng: false". Selanjutnya, program menghitung total berat dari kedua kantong, dan jika total berat melebihi 150 kg, proses dihentikan dengan mencetak "Proses selesai."

6. Soal Studi Case

Diberikan sebuah persamaan sebagai berikut ini.

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

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

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

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

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

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

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

func sqrtApproximation(kMax int) float64 {
    product := 1.0
    for k := 0; k <= kMax; k++ {
        product *= term(k)
    }
    return product
}

func main() {
    // Input nilai K
    var k int
    fmt.Print("Nilai K: ")
    fmt.Scan(&k)
```

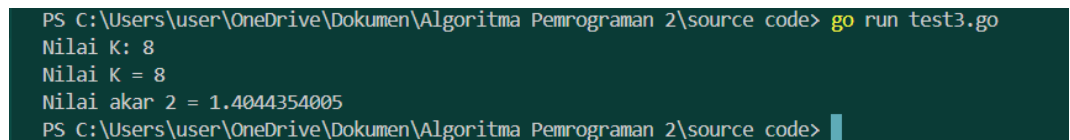
```
    approx := sqrtApproximation(k)

    fmt.Printf("Nilai K = %d\n", k)

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

}
```

Screenshot Output



```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code> go run test3.go
Nilai K: 8
Nilai K = 8
Nilai akar 2 = 1.4044354005
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code>
```

Deskripsi Program

Program yang ada diatas itu adalah program yang dirancang untuk menghitung hampiran (aproksimasi) akar kuadrat dari 2 metode produk deret yang bergantung pada nilai k. Saat program dijalankan program mendeklarasikan paket main dan mengimpor paket fmt untuk input/output serta math untuk operasi matematika. Lalu fungsi `term(k int) float64` menghitung nilai suku berdasarkan rumus. Kemudian fungsi `sqrtApproximation(kMax int) float64` menggunakan loop untuk mengalikan setiap suku dari `term(k)` ke dalam variabel `product`. lalu Fungsi `main` meminta input nilai K dari pengguna, menghitung hasil menggunakan fungsi `sqrtApproximation(k)`, dan kemudian menampilkan hasil hampiran akar kuadrat 2 dengan format desimal hingga 10 angka di belakang koma. Semakin besar nilai K yang dimasukkan, semakin mendekati hasilnya dengan nilai asli akar kuadrat 2. Itu saja yang saya ketahui