

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL I
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

Rakha Arbiyandanu / 2311102263

IF-11-6

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Struktur kontrol dalam Golang adalah mekanisme yang mengatur alur eksekusi program. Mereka memungkinkan pengembang untuk membuat keputusan, mengulang blok kode, dan mengontrol jalannya program. dalam kerangka program yang ditulis dalam Bahasa program golang program utama selalu mempunyai dua komponen penting yaitu

- Package main merupakan penanda bahwa file ini berisi program utama
- func main() berisi kode utama dari sebuah program Go

Program pertama golang adalah sebuah program dasar yang memunculkan output atau keluaran berupa text “Hello, World”. Untuk mencoba menjalankan program pertama golang silahkan tekan tombol “COBA” yang berada dibawah contoh pada aplikasi, kemudian tekan tombol “RUN” maka akan muncul jendela output dari program yang dijalankan

```
package main

import "fmt"

// ini adalah sebuah komentar

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

Pada program diatas, pada bagian baris pertama terdapat potongan kode “package main” atau bisa disebut dengan deklarasi dari paket yang digunakan. Paket adalah sebuah cara untuk mengatur dan menggunakan kode program yang dibutuhkan. Program Golang (Go) terbagi menjadi dua jenis yaitu executable dan pustaka (libraries). Program Golang executable adalah program golang yang dapat dijalankan langsung melalui terminal yang tersedia. Sedangkan program Golang pustaka (libraries) adalah program golang yang merupakan kumpulan atau koleksi dari program Golang yang dijadikan paket sehingga dapat dijalankan pada program Golang lain

Tipe Data dan Intruksi Dasar

1. Tipe Data Numerik

Go memiliki beberapa jenis tipe data, termasuk tipe data numerik (baik desimal maupun non-desimal), string, dan boolean. Tanpa disadari, pada pembahasan sebelumnya kita sudah menggunakan beberapa tipe data seperti string dan tipe numerik int. Pada bagian ini, akan dijelaskan berbagai macam tipe data standar yang tersedia di Go, beserta contoh penggunaannya.

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).

Kedua tipe data di atas kemudian dibagi lagi menjadi beberapa jenis, dengan pembagian berdasarkan lebar cakupan nilainya, detailnya bisa dilihat di tabel berikut.

Tipe data	Cakupan bilangan
<code>uint8</code>	0 ↔ 255
<code>uint16</code>	0 ↔ 65535
<code>uint32</code>	0 ↔ 4294967295
<code>uint64</code>	0 ↔ 18446744073709551615
<code>uint</code>	sama dengan <code>uint32</code> atau <code>uint64</code> (tergantung nilai)
<code>byte</code>	sama dengan <code>uint8</code>
<code>int8</code>	-128 ↔ 127
<code>int16</code>	-32768 ↔ 32767
<code>int32</code>	-2147483648 ↔ 2147483647
<code>int64</code>	-9223372036854775808 ↔ 9223372036854775807
<code>int</code>	sama dengan <code>int32</code> atau <code>int64</code> (tergantung nilai)
<code>rune</code>	sama dengan <code>int32</code>

2. 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.

```
var decimalNumber = 2.62

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

Pada kode di atas, variabel `decimalNumber` akan memiliki tipe data `float32`, karena nilainya berada di cakupan tipe data tersebut.

```
novalagung:belajar-golang $ go run bab9.go
bilangan desimal: 2.620000
bilangan desimal: 2.620
novalagung:belajar-golang $
```

3. Tipe Data Boolean

```
var exist bool = true

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

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

4. 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)
```

5. 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{}`)

I. GUIDED

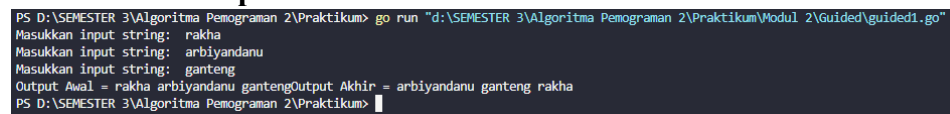
Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

Sourcecode

```
package main
import "fmt"

func main(){
    var(
        satu,dua,tiga string
        temp string
    )
    fmt.Print("Masukkan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukkan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("Masukkan input string: ")
    fmt.Scanln(&tiga)
    fmt.Print("Output Awal = " + satu + " " + dua + " " +
tiga)
    temp = satu
    satu = dua
    dua = tiga
    tiga = temp
    fmt.Print("Output Akhir = " + satu + " " + dua + " " +
+ tiga)
}
```

Screenshot Output



```
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\Modul 2\Guided\guided1.go"
Masukkan input string: rakha
Masukkan input string: arbiyandanu
Masukkan input string: ganteng
Output Awal = rakha arbiyandanu gantengOutput Akhir = arbiyandanu ganteng rakha
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> |
```

Deskripsi Program

Program diatas digunakan untuk mengubah suatu variabel string, misalnya dari 1 menjadi 2, 2 menjadi 3. Data string dapat diperoleh dengan memasukkan data string dari 1 hingga 3 dan mengubah data string dari 1 ke 2.

II. GUIDED

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(){

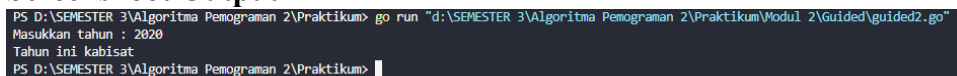
    var tahun int

    fmt.Print("Masukkan tahun : ")
    fmt.Scanln(&tahun)

    if tahun%400 == 0 {
        fmt.Println("Tahun ini kabisat")
    } else if tahun%100 == 0 {
        fmt.Println("Ini bukan tahun kabisat")
    } else if tahun%4 == 0 {
        fmt.Println("Tahun ini kabisat")
    } else {
        fmt.Print("Tidak terdefinisi")
    }

}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\Modul 2\Guided\guided2.go"
Masukkan tahun : 2020
Tahun ini kabisat
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> |
```

Deskripsi Program

Program diatas digunakan untuk menentukan apakah suatu tahun merupakan tahun kabisat atau bukan dengan cara user masukan tahun yang ingin di cek apakah itu tahun kabisat atau bukan jika yaa akan muncul pesan ini tahun kabisat jika tidak akan muncul pesan ini bukan tahun kabisat.

III. GUIDED

Buatkan program hipotenusa

```
package main

import "fmt"

func main() {

    var a, b, c float64

    var hipotenusa bool

    fmt.Print("Masukkan nilai sisi a, b, dan c: ")

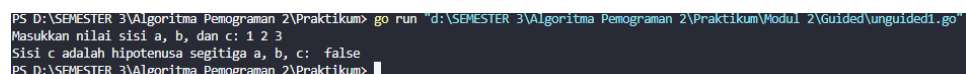
    fmt.Scanln(&a, &b, &c)

    hipotenusa = (c*c) == (a*a + b*b)

    fmt.Println("Sisi c adalah hipotenusa segitiga  
a, b, c: ", hipotenusa)

}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\Modul 2\Guided\unguided1.go"
Masukkan nilai sisi a, b, dan c: 1 2 3
Sisi c adalah hipotenusa segitiga a, b, c: false
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum>
```

Deskripsi Program

Program Go ini menguji apakah sisi c dari sebuah segitiga dengan sisi a, b, dan c merupakan sisi miring. Pengguna memasukkan nilai untuk a, b, dan c. Selanjutnya, program menghitung apakah $c^2 = a^2 + b^2$. Jika benar, variabel sisi miring disetel ke benar, yang menunjukkan bahwa c adalah sisi miring. Hasil ini kemudian ditampilkan kepada pengguna.

IV. GUIDED

Soal Studi Case

Dibaca nilai temperatur dalam derajat Celsius. Nyatakan temperatur tersebut dalam 5 lab 4 Celsius (Fahrenheit-32)x Reamur Celcius x Kelvin (Fahrenheit + 459.67)×5/2

Sourcecode

```
package main

import (
    "fmt"
)

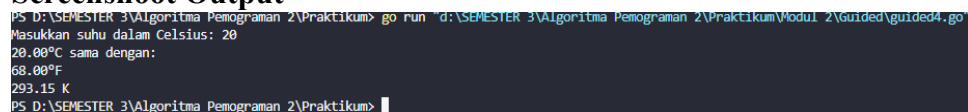
func main() {
    var celsius float64

    fmt.Print("Masukkan suhu dalam Celsius: ")
    fmt.Scan(&celsius)

    fahrenheit := (celsius * 9 / 5) + 32
    kelvin := celsius + 273.15

    fmt.Printf("%.2f°C sama dengan:\n", celsius)
    fmt.Printf("%.2f°F\n", fahrenheit)
    fmt.Printf("%.2f K\n", kelvin)
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum> go run d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\Modul 2\Guided\guided4.go
Masukkan suhu dalam Celsius: 20
20.00°C sama dengan:
68.00°F
293.15 K
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum>
```

Deskripsi Program

Program diatas digunakan untuk menghitung suhu dalam Celsius menjadi fahrenheit dan kelvin dengan rumus $fahrenheit := (celsius * 9 / 5) + 32$ dan $kelvin := celsius + 273.15$ kemudian data suhu tadi di olah menjadi suhu fahrenheit dan kelvin

V. GUIDED

Soal Studi Case

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

Sourcecode

```
package main

import "fmt"

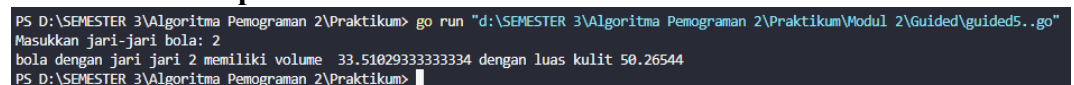
func main() {
    var jariJari float64
    const pi = 3.14159

    fmt.Print("Masukkan jari-jari bola: ")
    fmt.Scan(&jariJari)

    luasKulit := 4 * pi * (jariJari * jariJari)
    volume := (4.0 / 3.0) * pi * (jariJari * jariJari *
    jariJari)

    fmt.Println("bola dengan jari jari", jariJari,
    "memiliki volume ", volume, "dengan luas kulit",
    luasKulit)
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\Modul 2\Guided\guided5..go"
Masukkan jari-jari bola: 2
bola dengan jari jari 2 memiliki volume 33.51029333333334 dengan luas kulit 50.26544
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum>
```

Deskripsi Program

Program diatas digunakan untuk menghitung jari jari suatu bola dengan inputan yang di input oleh user kemudian di hitung dengan rumus $\text{luasKulit} := 4 * \pi * (\text{jariJari} * \text{jariJari})$ dan $\text{volume} := (4.0 / 3.0) * \pi * (\text{jariJari} * \text{jariJari} * \text{jariJari})$ kemudian keluar output yang sudah di hitung dengan jari jari yang di inputkan oleh user

I. UNGUIDED

Soal Studi Case

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).

Sourcecode

```
package main

import (
    "fmt"
)

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

    fmt.Println("Masukkan 5 angka integer (32-127)
dipisahkan spasi:")
    for i := 0; i < 5; i++ {
        fmt.Scan(&integers[i])
    }

    fmt.Println("Masukkan 3 karakter tanpa spasi:")
    var input string
    fmt.Scan(&input)
    for i := 0; i < 3 && i < len(input); i++ {
        characters[i] = input[i]
    }

    fmt.Print("Output baris pertama: ")
    for _, i := range integers {
        fmt.Printf("%c", rune(i))
    }
    fmt.Println()

    fmt.Print("Output baris kedua: ")
    for _, c := range characters {
        fmt.Printf("%c", c+1)
    }
}
```

```
    fmt.Println()  
}
```

Screenshoot Output

```
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\Modul 2\Guided\unguided1.go"  
Masukkan 5 angka integer (32-127) dipisahkan spasi:  
33 34 34 36 39 40  
Masukkan 3 karakter tanpa spasi:  
Output baris pertama: !""$'  
Output baris kedua: 510  
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> |
```

Deskripsi Program

Kode GoLang di atas meminta pengguna untuk memasukkan 5 angka integer (dalam rentang 32-127, sesuai dengan kode ASCII) dan 3 karakter secara langsung tanpa spasi. Program kemudian menampilkan dua baris output: baris pertama mengonversi angka integer yang dimasukkan ke dalam karakter ASCII yang sesuai, sedangkan baris kedua menampilkan karakter yang dimasukkan oleh pengguna, tetapi dengan setiap karakter digeser satu posisi maju dalam tabel ASCII (yaitu karakter berikutnya).

II. UNGUIDED

Siswa kelas IPA disalah satu sekolah menengah atas di indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan 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 dare 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"
)

func main() {

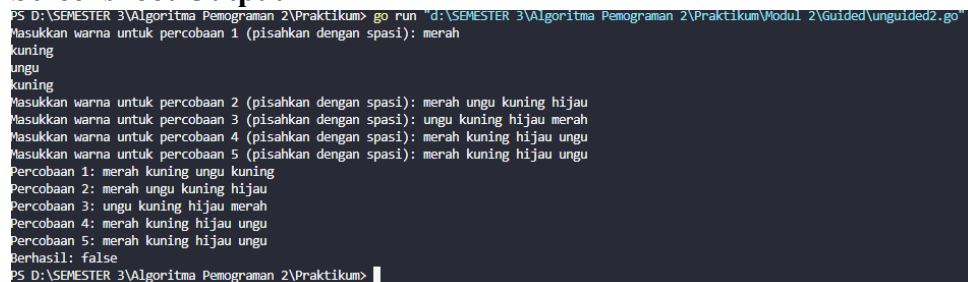
    var warna [5][4]string
    urutanBenar := [4]string{"merah", "kuning", "hijau",
"ungu"}
    berhasil := true

    for i := 0; i < 5; i++ {
        fmt.Printf("Masukkan warna untuk percobaan %d
(pisahkan dengan spasi): ", i+1)
        fmt.Scan(&warna[i][0], &warna[i][1],
&warna[i][2], &warna[i][3])
    }

    for i := 0; i < 5; i++ {
        fmt.Printf("Percobaan %d: %s %s %s %s\n", i+1,
warna[i][0], warna[i][1], warna[i][2], warna[i][3])
        if warna[i] != urutanBenar {
            berhasil = false
        }
    }

    fmt.Printf("Berhasil: %t\n", berhasil)
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\Modul 2\Guided\unguided2.go"
Masukkan warna untuk percobaan 1 (pisahkan dengan spasi): merah
kuning
ungu
kuning
Masukkan warna untuk percobaan 2 (pisahkan dengan spasi): merah ungu kuning hijau
Masukkan warna untuk percobaan 3 (pisahkan dengan spasi): ungu kuning hijau merah
Masukkan warna untuk percobaan 4 (pisahkan dengan spasi): merah kuning hijau ungu
Masukkan warna untuk percobaan 5 (pisahkan dengan spasi): merah kuning hijau ungu
Percobaan 1: merah kuning ungu kuning
Percobaan 2: merah ungu kuning hijau
Percobaan 3: ungu kuning hijau merah
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
Berhasil: false
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> █
```

Deskripsi Program

Program Go ini meminta pengguna untuk memasukkan urutan warna dalam lima percobaan, di mana setiap percobaan terdiri dari empat warna. Program kemudian memverifikasi apakah setiap percobaan sesuai dengan urutan

warna yang benar (merah, kuning, hijau, ungu). Jika semua percobaan sesuai dengan urutan tersebut, variabel berhasil tetap true. Namun, jika ada percobaan yang tidak sesuai, variabel berhasil akan berubah menjadi false. Pada akhir program, hasilnya (berhasil: true atau false) akan ditampilkan kepada pengguna

III. UNGUIDED

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. Tampilkan isi pita setelah proses input selesai. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read) 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 dibawah ini (teks bergaris bawah adalah input/read):

Sourcecode

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var N int
    fmt.Print("Masukkan jumlah bunga (N): ")
    fmt.Scan(&N)

    var pita strings.Builder
    var bunga string
    count := 0

    for i := 1; i <= N; i++ {
        fmt.Printf("Bunga %d: ", i)
        fmt.Scan(&bunga)

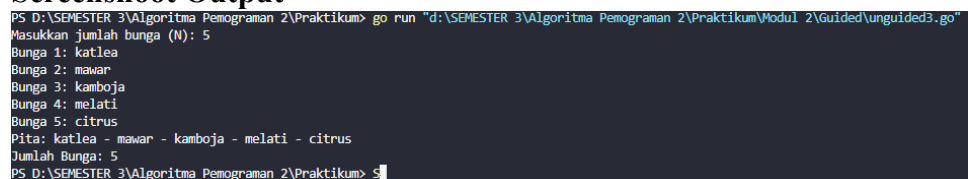
        if strings.ToUpper(bunga) == "SELESAI" {
            break
        }

        if pita.Len() > 0 {
            pita.WriteString(" - ")
        }

        pita.WriteString(bunga)
        count++
    }

    fmt.Printf("Pita: %s\n", pita.String())
    fmt.Printf("Jumlah Bunga: %d\n", count)
}
```

Screenshot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\Modul 2\Guided\unguided3.go"
Masukkan jumlah bunga (N): 5
Bunga 1: katlea
Bunga 2: mawar
Bunga 3: kamboja
Bunga 4: melati
Bunga 5: citrus
Pita: katlea - mawar - kamboja - melati - citrus
Jumlah Bunga: 5
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum> $
```

Deskripsi Program

Program Go ini meminta pengguna untuk memasukkan sejumlah nama bunga dalam sebuah pita, hingga mencapai jumlah tertentu (N) atau sampai pengguna memasukkan kata "SELESAI". Nama-nama bunga digabungkan dalam sebuah string dengan tanda "-" sebagai pemisah.

Program juga menghitung dan menampilkan jumlah bunga yang input.

IV. UNGUIDED

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. Tampilkan isi pita setelah proses input selesai. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read) 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 dibawah ini (teks bergaris bawah adalah input/read):

Sourcecode

```
package main

import (
    "fmt"
)

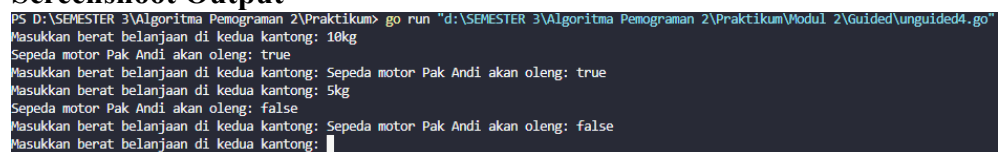
func main() {

    var berat1, berat2 float64

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

        if berat1 >= 9 || berat2 >= 9 {
            fmt.Println("Sepeda motor Pak Andi akan
oleng: true")
        } else {
            fmt.Println("Sepeda motor Pak Andi akan
oleng: false")
        }
    }
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\Modul 2\Guided\unguided4.go"
Masukkan berat belanjaan di kedua kantong: 10kg
Sepeda motor Pak Andi akan oleng: true
Masukkan berat belanjaan di kedua kantong: Sepeda motor Pak Andi akan oleng: true
Masukkan berat belanjaan di kedua kantong: 5kg
Sepeda motor Pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong: Sepeda motor Pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong: 
```

Deskripsi Program

Program Go ini meminta pengguna untuk memasukkan berat belanjaan di dua kantong secara berulang. Jika salah satu kantong memiliki berat 9 kg atau lebih, program akan menampilkan pesan bahwa sepeda motor Pak Andi akan oleng (berat sebelah). Jika tidak, program menyatakan sepeda motor tidak akan oleng. Proses ini berlangsung dalam loop tanpa henti sampaiimbang.

V. UNGUIDED

buatlah 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 dibawah ini (teks bergaris bawah adalah input/read):

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func hitungF(k int) float64 {
    pembilang := math.Pow(float64(4*k+2), 2)
    penyebut := float64((4*k + 1) * (4*k + 3))
    return pembilang / penyebut
}

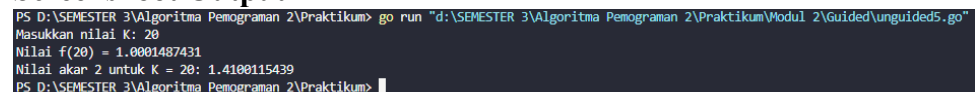
func hitungAkarDua(k int) float64 {
    hasil := 1.0
    for i := 0; i <= k; i++ {
        hasil *= hitungF(i)
    }
    return hasil
}

func main() {
    var k int
    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&k)

    nilaiF := hitungF(k)
    fmt.Printf("Nilai f(%d) = %.10f\n", k, nilaiF)

    akarDua := hitungAkarDua(k)
    fmt.Printf("Nilai akar 2 untuk K = %d: %.10f\n", k,
    akarDua)
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum> go run "d:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\Modul 2\Guided\unguided5.go"
Masukkan nilai K: 20
Nilai f(20) = 1.0001487431
Nilai akar 2 untuk K = 20: 1.4100115439
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum>
```

Deskripsi Program

Program Go ini menghitung nilai dari sebuah fungsi matematika dan mendekati nilai akar dua dengan nilai K tertentu. Fungsi `hitungF(k)` menghitung nilai dengan pembilang dan penyebut . Fungsi `hitungAkarDua(k)` mengalikan hasil dari `hitungF(i)` dari 0 sampai K untuk mendekati nilai akar dua. Program meminta input nilai K dari pengguna, kemudian menampilkan nilaidan hasil pendekatan nilai akar dua untuk K yang dimasukkan.