

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II  
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**Rafi Bintang Maulana / 2311102327**

**Kelas IF-11-06**

**Dosen Pengampu :**

**ABEDNEGO DWI SEPTIADI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## **I. DASAR TEORI**

Algoritma merupakan inti dari pemrograman komputer, yang memungkinkan kita untuk melakukan perhitungan seperti memecahkan masalah secara efektif dan efisien. Secara umum, algoritma adalah serangkaian instruksi yang ditujukan untuk menyelesaikan suatu masalah atau mencapai hasil tertentu dengan langkah-langkah yang sistematis.

Dalam pemrograman, struktur kontrol adalah komponen dasar yang digunakan untuk menentukan alur eksekusi dari instruksi-instruksi di dalam program. Struktur kontrol memungkinkan sebuah program untuk mengambil keputusan, mengulangi suatu perintah, atau bahkan melompati bagian tertentu dari kode berdasarkan kondisi tertentu.

## II. GUIDED

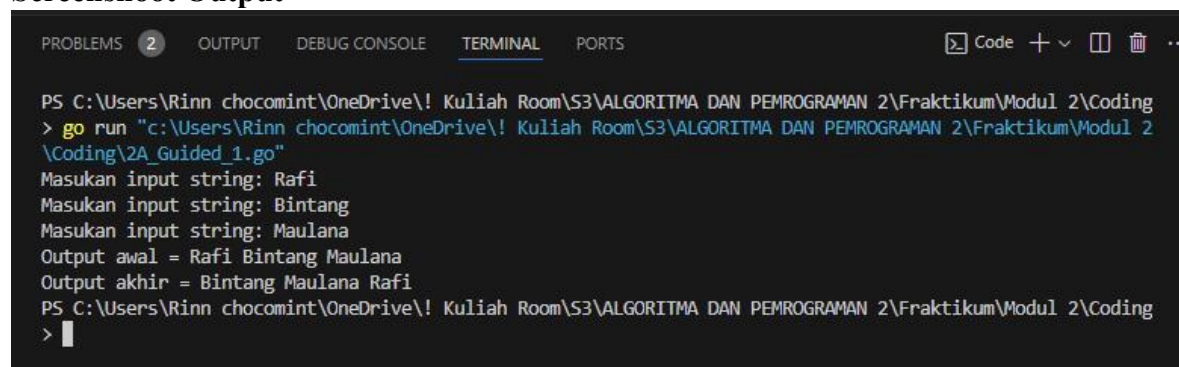
1. Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

### Sourcecode

```
package main
import "fmt"

func main(){
    var (
        satu, dua, tiga string
        temp string
    )
    fmt.Print("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
    dua = tiga
    tiga = temp
    fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga )
}
```

### Screenshoot Output



```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding
> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2
\Coding\2A_Guided_1.go"
Masukan input string: Rafi
Masukan input string: Bintang
Masukan input string: Maulana
Output awal = Rafi Bintang Maulana
Output akhir = Bintang Maulana Rafi
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding
> |
```

### Deskripsi Program

Pada program tersebut, Program awalnya meminta tiga input string dari pengguna, Kemudian menampilkan input tersebut dalam urutan semula dan melakukan pertukaran nilai antara ketiganya. Pada akhir prosesnya, setiap String yang diinputkan akan disimpan dalam variabel, Hasil akhir dari pertukaran string ini kemudian ditampilkan.

2. 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

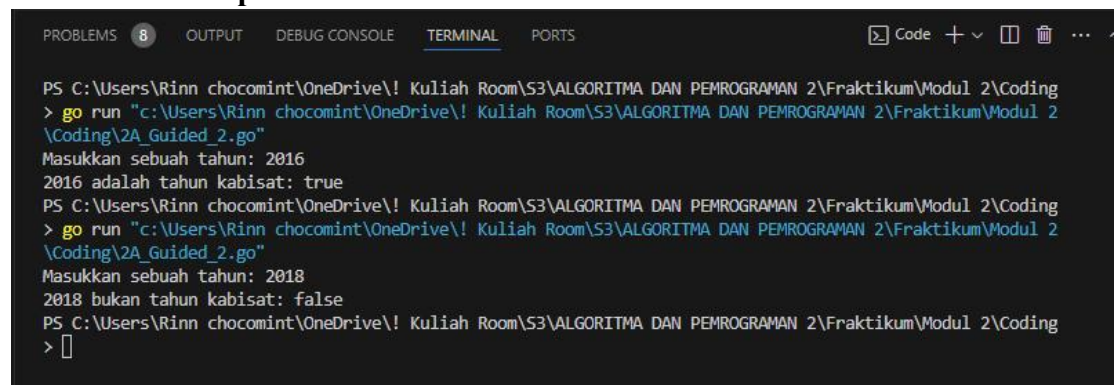
```
package main

import (
    "fmt"
)

func 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")
    }
}
```

### Screenshoot Output



```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding
> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2
\Coding\2A_Guided_2.go"
Masukkan sebuah tahun: 2016
2016 adalah tahun kabisat: true
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding
> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2
\Coding\2A_Guided_2.go"
Masukkan sebuah tahun: 2018
2018 bukan tahun kabisat: false
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding
> 
```

### Deskripsi Program

Program ini memeriksa inputan pengguna, apakah suatu tahun merupakan tahun kabisat atau bukan. Program itu menggunakan logika aritmatika untuk menentukan apakah tahun tersebut habis dibagi 400, atau habis dibagi 4 tetapi tidak habis dibagi 100. Jika salah satu kondisi terpenuhi, program menampilkan bahwa tahun tersebut adalah tahun kabisat. Jika tidak, program akan menampilkan bahwa tahun tersebut bukan tahun kabisat.

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

#### Sourcecode

```
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)
}
```

#### Screenshot Output



```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding
> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2
\Coding\2A_Guided_3.go"
Masukkan jari-jari bola : 5
Volume bola dengan jari-jari 5.00 adalah 523.60
Luas permukaan bola dengan jari-jari 5.00 adalah 314.16
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding
>
```

#### Deskripsi Program

Program ini untuk menghitung volume dan luas permukaan bola berdasarkan jari-jari yang dimasukkan oleh pengguna. Setelah pengguna memasukkan nilai jari-jari, program menggunakan rumus matematika untuk menghitung volume bola. Fungsi “**math.Pi**” digunakan untuk nilai  $\pi$  (pi) dan “**math.Pow**” untuk menghitung pangkat. Hasil perhitungan volume dan luas permukaan ditampilkan dengan format dua angka desimal.

### III. UNGUIDED

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

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

#### Sourcecode

```
package main

import (
    "fmt"
)

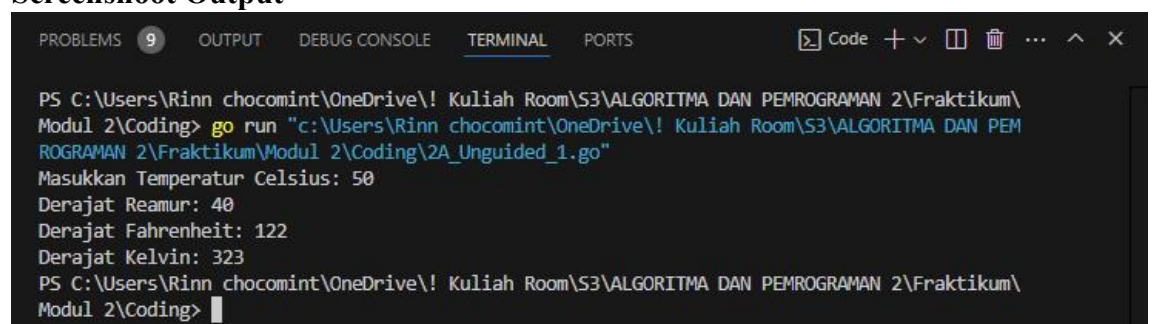
func main() {
    var celsius float64

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

    untukReamur := int(celsius * 4 / 5)
    untukFahrenheit := int((celsius * 9 / 5) + 32)
    untukKelvin := int(celsius + 273.15)

    fmt.Printf("Derajat Reamur: %d\n", untukReamur)
    fmt.Printf("Derajat Fahrenheit: %d\n", untukFahrenheit)
    fmt.Printf("Derajat Kelvin: %d\n", untukKelvin)
}
```

#### Screenshoot Output



```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding\2A_Unguided_1.go"
Masukkan Temperatur Celsius: 50
Derajat Reamur: 40
Derajat Fahrenheit: 122
Derajat Kelvin: 323
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding>
```

#### Deskripsi Program

Program ini berfungsi untuk mengonversi suhu dari skala Celsius ke tiga skala suhu yaitu Reamur, Fahrenheit, dan Kelvin. Pengguna awalnya diminta untuk memasukkan nilai suhu dalam Celsius. Program lalu akan menghitung konversi ke Reamur, Fahrenheit, dan Kelvin. Setelah itu, hasil perhitungan akan ditampilkan.

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).

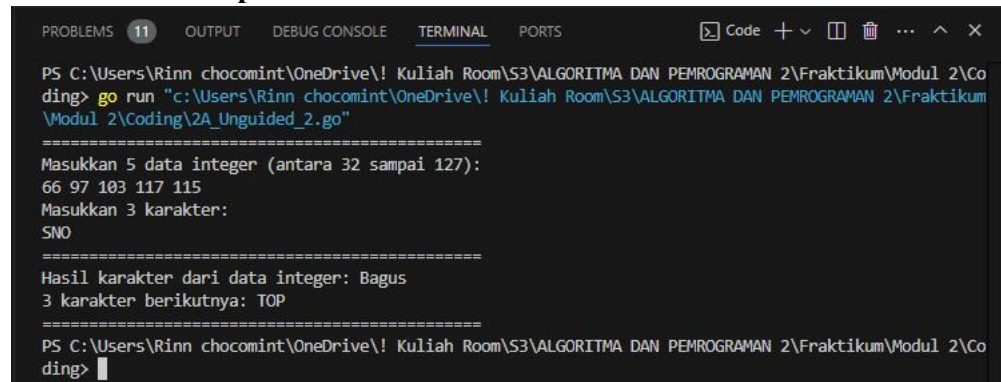
**Keluaran** juga terdiri dari dua baris. Baris pertama berisi 5 buah representasi karakter dari data yang diberikan, yang berdampingan satu dengan lain, tanpa dipisahkan spasi. Baris kedua berisi 3 buah karakter (juga tidak dipisahkan oleh spasi).

### Sourcecode

```
package main
import (
    "fmt"
)
func main() {
    var BilanganInteger [5]int
    var Karakter string
    fmt.Println("=====")
    fmt.Println("Masukkan 5 data integer (antara 32 sampai 127):")
    for i := 0; i < 5; i++ {
        fmt.Scan(&BilanganInteger[i])
    }
    fmt.Println("Masukkan 3 karakter:")
    fmt.Scan(&Karakter)
    fmt.Println("=====")
    fmt.Print("Hasil karakter dari data integer: ")
    for _, value := range BilanganInteger {
        if value >= 32 && value <= 127 {
            fmt.Printf("%c", value)
        }
    }
    fmt.Println()

    if len(Karakter) == 3 {
        fmt.Print("3 karakter berikutnya: ")
        for _, ch := range Karakter {
            fmt.Printf("%c", ch+1)
        }
    }
    fmt.Println("\n=====")
    } else {
        fmt.Println("Input harus terdiri dari tepat 3 karakter.")
    }
}
```

## Screenshoot Output

A screenshot of a Windows terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The command prompt shows the user running a Go program. The program prompts the user to enter 5 integers between 32 and 127, and then 3 characters. The user enters '66 97 103 117 115' and 'SNO'. The program outputs 'Hasil karakter dari data integer: Bagus' and '3 karakter berikutnya: TOP'.

```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding\2A_Unguided_2.go"
=====
Masukkan 5 data integer (antara 32 sampai 127):
66 97 103 117 115
Masukkan 3 karakter:
SNO
=====
Hasil karakter dari data integer: Bagus
3 karakter berikutnya: TOP
=====
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding>
```

## Deskripsi Program

Program gunanya untuk mengonversi bilangan integer menjadi karakter ASCII serta memanipulasi karakter string. Pengguna diminta memasukkan lima bilangan integer dalam rentang 32 hingga 127, yang kemudian dikonversi ke karakter sesuai dengan kode ASCII-nya. Cara kerjanya sendiri Pertama, program menerima lima bilangan integer dari pengguna dan kemudian mencetak karakter yang sesuai dengan nilai ASCII dari bilangan tersebut. Selanjutnya, pengguna diminta memasukkan tiga karakter, dan program akan menampilkan karakter berikutnya dalam urutan ASCII untuk setiap karakter yang dimasukkan.

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 warna sesuai sesuai dengan dengan informasi informasi yang yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna lainnya.

## Sourcecode

```
package main
import (
    "bufio"
    "fmt"
    "os"
    "strings"
)
```



```

func main() {
    for ulang := 1; ulang <= 2; ulang++ {
        yangHarusDiinputkan := []string{"merah", "kuning", "hijau",
"ungu"}

        berhasil := true
        bacaInput := bufio.NewScanner(os.Stdin)

        fmt.Printf("=====\\n")
        for i := 1; i <= 5; i++ {
            fmt.Printf("Percobaan %d: ", i)

            if !bacaInput.Scan() {
                fmt.Println("\\nInput tidak cukup.")
                return
            }
            line := bacaInput.Text()
            colors := strings.Fields(line)

            if len(colors) != 4 {
                berhasil = false
                continue
            }
            for j, color := range colors {
                if strings.ToLower(color) !=
yangHarusDiinputkan[j] {
                    berhasil = false
                    break
                }
            }
        }
        fmt.Printf("BERHASIL: %v\\n", berhasil)
    }
}

```

## Screenshoot Output

```

PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding\2B_Unguided_3.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
=====
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau merah
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding>

```

### Deskripsi Program

Program tersebut berfungsi untuk memeriksa apakah pengguna dapat memasukkan urutan warna yang benar sesuai dengan target yang telah ditentukan. Inputnya sendiri adalah nama warna, diprogram tersebut sudah diprogram kalau data nama warna disimpan menggunakan array ber-Variabel “**yangHarusDiinputkan**”. Jika urutan input benar pada setiap percobaan, program akan menandai keberhasilan, dan jika salah, akan menandai kegagalan.

4. 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.

### Sourcecode

```
package main
import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    scanner := bufio.NewScanner(os.Stdin)
    pita := ""
    BunganyaAda := 0

    i := 1
    for {
        fmt.Printf("Bunga %d: ", i)
        scanner.Scan()
        bunga := scanner.Text()

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

        pita += bunga + " - "
        BunganyaAda++
        i++
    }

    if pita == "" {
        fmt.Println("Pita: ")
    } else {
```

```

        fmt.Printf("Pita: %s\n", strings.TrimSpace(pita))
    }

    fmt.Printf("Bunga: %d\n", BunganyaAda)
}

```

### Screenshot Output

```

PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Co
ding> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum
\Modul 2\Coding\tempCodeRunnerFile.go"
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Bunga 4: SELESAI
Pita: Kertas - Mawar - Tulip -
Bunga: 3
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Co
ding>

```

### Deskripsi Program

Program untuk mengumpulkan nama-nama bunga yang dimasukkan oleh pengguna hingga pengguna mengetik “SELESAI” Pengguna dapat terus menambahkan nama bunga sampai mereka mengetik “SELESAI”, yang akan menghentikan proses input. Jika tidak ada bunga yang dimasukkan, program hanya akan menampilkan string kosong untuk “pita”.

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. Lalu 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"
)

func main() {
    var beratBebanKiri, beratBebanKanan float64

    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&beratBebanKiri, &beratBebanKanan)
    }
}

```

```

        totalBerat := beratBebanKiri + beratBebanKanan
        if beratBebanKiri < 0 || beratBebanKanan < 0 || totalBerat > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := beratBebanKiri - beratBebanKanan
        if selisih < 0 {
            selisih = -selisih
        }

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

```

## Screenshoot Output

```

PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding\2B_Unguided_5.go"
Masukan berat belanja di kedua kantong: 5 10
Sepeda motor Pak Andi akan oleng: false
Masukan berat belanja di kedua kantong: 55.6 70.2
Sepeda motor Pak Andi akan oleng: true
Masukan berat belanja di kedua kantong: 72.3 66.9
Sepeda motor Pak Andi akan oleng: false
Masukan berat belanja di kedua kantong: 59.5 98.7
Proses selesai.
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding>

```

## Deskripsi Program

Program ini bertujuan untuk menghitung keseimbangan berat beban di sepeda motor Pak Andi berdasarkan berat belanja di dua kantong. Pengguna diminta memasukkan berat beban di kantong kiri dan kanan. Program akan menghitung total berat, dan jika salah satu berat negatif atau total berat melebihi 150 kg, program akan menghentikan proses dengan menampilkan pesan “Proses selesai”.

6. Diberikan 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.

Kemudian 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"
)

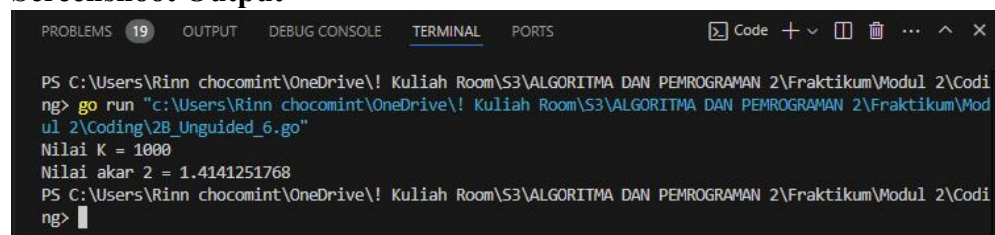
func hitungAkar2(k int) float64 {

    //Rumusnya ketika di code kan
    hasil := 1.0
    for i := 0; i < k; i++ {
        NilaiAtas := (4*float64(i) + 2) * (4*float64(i) + 2)
        NilaiBawah := (4*float64(i) + 1) * (4*float64(i) + 3)
        hasil *= NilaiAtas / NilaiBawah
    }
    return hasil
}

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

    akar2 := hitungAkar2(k)
    fmt.Printf("Nilai akar 2 = %.10f\n", akar2)
}
```

### Screenshoot Output



```
PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding\2B_Unguided_6.go"
Nilai K = 1000
Nilai akar 2 = 1.4141251768
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 2\Coding>
```

### Deskripsi Program

Program berguna untuk menghitung pendekatan nilai akar kuadrat dari 2 menggunakan metode iteratif berdasarkan parameter k yang dimasukkan oleh pengguna. Setelah pengguna memasukkan nilai k, program akan menampilkan hasil perhitungan nilai akar 2 dengan presisi 10 angka desimal. Fungsi hitungAkar2 melakukan perhitungan dengan menggunakan sebuah loop sebanyak k kali, setiap iterasi menghitung dan mengalikan hasil sebelumnya dengan rasio tertentu yang terdiri dari ekspresi matematika.

