

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2  
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**Aby Hakim Al Yasiry Faozi/2311102208**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### Dasar Teori

Struktur kontrol adalah salah satu elemen fundamental dalam pemrograman yang digunakan untuk mengatur alur eksekusi program. Dengan menggunakan struktur kontrol, kita dapat membuat keputusan, menjalankan perintah berulang, atau mengubah arah eksekusi berdasarkan kondisi tertentu. Secara umum, struktur kontrol dibagi menjadi tiga kategori utama:

1. Struktur Kontrol Sekuensial (Sequential Control Structure)
  - Merupakan bentuk paling dasar dari struktur kontrol di mana perintah-perintah dieksekusi secara berurutan, satu per satu dari atas ke bawah.
  - Contoh: Dalam sebuah program yang mencetak kalimat, perintah untuk mencetak kalimat pertama dilakukan terlebih dahulu, kemudian dilanjutkan dengan perintah selanjutnya.
2. Struktur Kontrol Percabangan (Conditional Control Structure)
  - Struktur ini memungkinkan program untuk membuat keputusan berdasarkan suatu kondisi yang diberikan. Jika kondisi terpenuhi (benar), maka satu blok kode akan dijalankan; jika tidak, maka blok kode lain yang akan dijalankan.
  - Struktur percabangan yang umum digunakan adalah:
    - If-Else: Memeriksa satu kondisi, jika benar, maka satu set instruksi dijalankan, jika salah, set instruksi yang berbeda akan dijalankan.
    - Switch-Case: Digunakan ketika ada beberapa kondisi yang harus diperiksa, dan setiap kondisi memiliki aksi yang berbeda.
3. Struktur Kontrol Perulangan (Looping Control Structure)
  - Digunakan untuk menjalankan sekumpulan instruksi secara berulang-ulang berdasarkan kondisi tertentu. Biasanya digunakan ketika kita ingin melakukan operasi yang sama beberapa kali.
  - Jenis perulangan yang umum meliputi:
    - For loop: Digunakan ketika jumlah perulangan sudah diketahui.
    - While loop: Digunakan ketika perulangan dilakukan selama kondisi tertentu masih benar.
    - Do-While loop: Sama seperti while loop, tetapi pengecekan kondisi dilakukan setelah satu kali eksekusi perintah.

## II. GUIDED

### 1. Soal Studi Case

#### Sourcecode

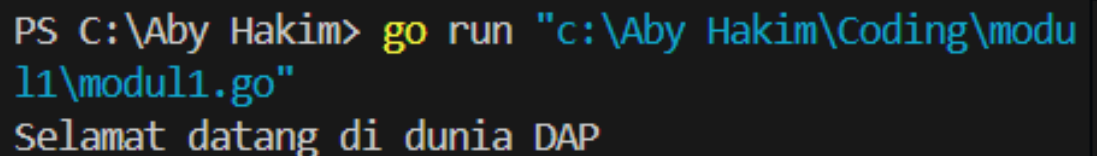
```
package main

import "fmt"

func main() {
    var greetings = "Selamat datang di dunia DAP"
    var a, b int

    fmt.Println(greetings)
    fmt.Scanln(&a, &b)
    fmt.Println("&v + &v = %v\n", a, b, a+b)
}
```

#### Screenshoot Output



```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Coding\modul1\modul1.go"
Selamat datang di dunia DAP
```

#### Deskripsi Program

##### Algoritma

1. Inisialisasi variabel:
  - Buat variabel greetings dengan nilai "Selamat datang di dunia DAP".
  - Buat dua variabel a dan b yang bertipe data integer untuk menyimpan angka yang diinput oleh pengguna.
2. Cetak pesan sambutan: Tampilkan pesan "Selamat datang di dunia DAP" ke layar.
3. Membaca input pengguna: Program meminta pengguna untuk memasukkan dua angka, yang disimpan ke variabel a dan b.
4. Operasi penjumlahan: Program menjumlahkan angka-angka yang telah dimasukkan oleh pengguna (a + b).
5. Cetak hasil penjumlahan: Program mencetak hasil penjumlahan tersebut dalam format tertentu ke layar.

## 2. 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?

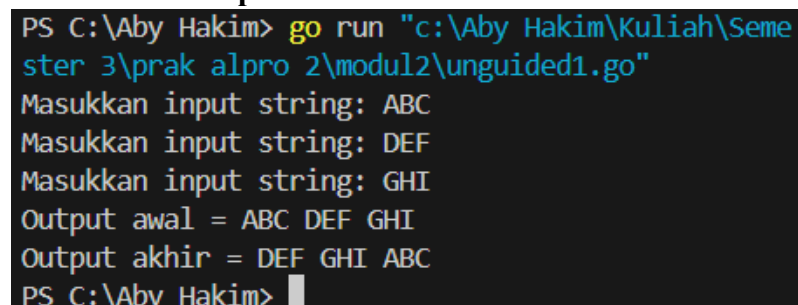
### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var (
        satu, dua, tiga string
        temp            string
    )
    fmt.Print("Masukkan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukkan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("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:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\unguided1.go"
Masukkan input string: ABC
Masukkan input string: DEF
Masukkan input string: GHI
Output awal = ABC DEF GHI
Output akhir = DEF GHI ABC
PS C:\Aby Hakim>
```

### Deskripsi Program

Algoritma:

#### 1. Deklarasi Variabel:

- Buat variabel satu, dua, tiga, dan temp dengan tipe data string.

2. Input Pengguna:

- Minta pengguna untuk memasukkan tiga string, masing-masing disimpan dalam variabel satu, dua, dan tiga.

3. Cetak Output Awal:

- Cetak ketiga string dalam urutan yang dimasukkan oleh pengguna.

4. Penukaran Nilai:

- Simpan nilai satu dalam variabel sementara temp.
- Berikan nilai dua ke satu.
- Berikan nilai tiga ke dua.
- Berikan nilai temp (yang berisi nilai awal satu) ke tiga.

5. Cetak Output Akhir:

- Cetak ketiga string dalam urutan baru setelah penukaran

Cara Kerja Program:

1. Langkah 1: Program dimulai dan menyiapkan variabel-variabel yang diperlukan (satu, dua, tiga, temp).

2. Langkah 2: Program meminta pengguna untuk memasukkan tiga string berturut-turut. Nilai-nilai string ini disimpan dalam variabel satu, dua, dan tiga.

3. Langkah 3: Program mencetak nilai awal dari variabel-variabel tersebut, yaitu urutan string seperti yang dimasukkan oleh pengguna.

4. Langkah 4: Program kemudian melakukan penukaran nilai:

- Variabel sementara temp digunakan untuk menyimpan nilai satu.
- Nilai dua dipindahkan ke satu.
- Nilai tiga dipindahkan ke dua.
- Nilai temp (yang awalnya nilai satu) dipindahkan ke tiga.

5. Langkah 5: Program mencetak nilai akhir dari variabel-variabel tersebut setelah penukaran, menunjukkan urutan string baru.

### 3. 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

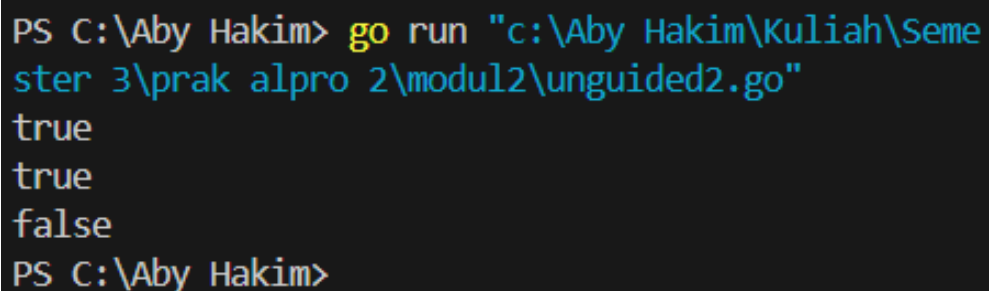
```
package main

import (
    "fmt"
)

func isLeapYear(year int) bool {
    if year%400 == 0 {
        return true
    }
    if year%100 == 0 {
        return false
    }
    if year%4 == 0 {
        return true
    }
    return false
}

func main() {
    fmt.Println(isLeapYear(2016)) // Output: true
    fmt.Println(isLeapYear(2000)) // Output: true
    fmt.Println(isLeapYear(2018)) // Output: false
}
```

#### Screenshoot Output



```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\unguided2.go"
true
true
false
PS C:\Aby Hakim>
```

#### Deskripsi Program

Program ini dibuat untuk menentukan apakah suatu tahun adalah tahun kabisat atau bukan menggunakan bahasa pemrograman Go. Program akan

mengecek tahun yang dimasukkan berdasarkan kriteria yang telah ditentukan dan mencetak hasilnya.

Algoritma:

1. Deklarasi Fungsi:

- Buat fungsi `isLeapYear(year int) bool` yang menerima satu parameter `year` dengan tipe data integer dan mengembalikan nilai boolean (`true` atau `false`).

2. Pengecekan Tahun Kabisat:

- Jika `year` habis dibagi 400, maka kembalikan `true`.
- Jika `year` habis dibagi 100, maka kembalikan `false`.
- Jika `year` habis dibagi 4, maka kembalikan `true`.
- Jika tidak ada kondisi di atas yang terpenuhi, kembalikan `false`.

3. Fungsi Utama:

- Panggil fungsi `isLeapYear` dengan berbagai tahun sebagai parameter, dan cetak hasilnya menggunakan `fmt.Println`.

Cara kerja:

6. Langkah 1: Program dimulai dengan mendeklarasikan paket `main` dan mengimpor paket `fmt` untuk input dan output.
7. Langkah 2: Fungsi `isLeapYear` didefinisikan untuk menentukan apakah tahun yang diberikan adalah tahun kabisat atau bukan.
  - Jika `year` habis dibagi 400: Fungsi mengembalikan nilai `true` karena tahun tersebut adalah tahun kabisat.
  - Jika `year` habis dibagi 100: Fungsi mengembalikan nilai `false` karena tahun tersebut bukan tahun kabisat.
  - Jika `year` habis dibagi 4: Fungsi mengembalikan nilai `true` karena tahun tersebut adalah tahun kabisat.
  - Jika tidak: Fungsi mengembalikan nilai `false` karena tahun tersebut bukan tahun kabisat.
8. Langkah 3: Fungsi `main` memanggil `isLeapYear` dengan beberapa contoh tahun (2016, 2000, 2018) dan mencetak hasilnya.

#### 4. Soal Studi Case

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

(Contoh input/output, Teks bergaris bawah adalah input dari user):

Jejari = 5

Bola dengan jejari 5 memiliki volume 523.5988 dan luas kulit 314.1593

#### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var radius float64
    const pi = 3.1415926535

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

    volume := (4.0 / 3.0) * pi * math.Pow(radius, 3)
    surfaceArea := 4 * pi * math.Pow(radius, 2)

    fmt.Printf("Bola dengan jari-jari %.2f memiliki volume %.4f dan luas permukaan %.4f\n", radius, volume, surfaceArea)
}
```

#### Screenshoot Output

```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\guided3.go"
Masukkan jari-jari bola: 4
Bola dengan jari-jari 4.00 memiliki volume 268.0826 dan luas permukaan 201.0619
PS C:\Aby Hakim> █
```



## Deskripsi Program

Algoritma:

1. Deklarasi Variabel:
  - Buat variabel radius, volume, dan surfaceArea untuk menyimpan nilai jari-jari, volume, dan luas permukaan bola.
  - Gunakan konstanta pi untuk menyimpan nilai Pi ( $\pi$ ).
2. Input Pengguna:
  - Minta pengguna untuk memasukkan nilai jari-jari bola.
3. Hitung Volume dan Luas Permukaan:
  - Hitung volume bola menggunakan rumus: volume
  - Hitung luas permukaan bola menggunakan rumus
4. Cetak Hasil:
  - Cetak hasil perhitungan volume dan luas permukaan bola.

Cara Kerja Program:

1. **Langkah 1:** Program dimulai dengan mendeklarasikan paket main dan mengimpor paket fmt untuk input dan output.
2. **Langkah 2:** Dalam fungsi main, program mendeklarasikan variabel untuk radius, volume, dan luas permukaan bola, serta konstanta pi.
3. **Langkah 3:** Program meminta pengguna untuk memasukkan nilai jari-jari bola dan menyimpannya dalam variabel radius.
4. **Langkah 4:** Program menghitung volume dan luas permukaan bola menggunakan rumus yang telah ditentukan.
5. **Langkah 5:** Program mencetak hasil perhitungan volume dan luas permukaan bola.

### III. UNGUIDED

#### 1. Soal Studi Case

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

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

(Contoh input/output, Teks bergaris bawah adalah input dari user):

Temperatur Celsius: 50  
Derajat Fahrenheit: 122

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

(Contoh input/output, Teks bergaris bawah adalah input dari user):

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

#### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var celsius float64
    fmt.Print("Masukkan suhu dalam Celsius: ")
    fmt.Scanln(&celsius)

    fahrenheit := celsius*9/5 + 32
    reamur := celsius * 4 / 5
    kelvin := celsius + 273.15

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

### Screenshoot Output

```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\unguided3.go"
Masukkan suhu dalam Celsius: 36
Derajat Fahrenheit: 96.80
Derajat Reamur: 28.80
Derajat Kelvin: 309.15
PS C:\Aby Hakim> 
```

### Deskripsi Program

Algoritma:

#### 1. Deklarasi Variabel:

- Buat variabel celsius untuk menyimpan nilai suhu dalam Celsius.
- Buat variabel fahrenheit, reamur, dan kelvin untuk menyimpan nilai suhu hasil konversi.

#### 2. Input Pengguna:

- Minta pengguna untuk memasukkan suhu dalam Celsius dan simpan nilai tersebut di variabel celsius.

#### 3. Konversi Suhu:

- Konversi suhu dari Celsius ke Fahrenheit dengan rumus:  $fahrenheit = celsius * 9/5 + 32$ .
- Konversi suhu dari Celsius ke Reamur dengan rumus:  $reamur = celsius * 4/5$ .
- Konversi suhu dari Celsius ke Kelvin dengan rumus:  $kelvin = celsius + 273.15$ .

#### 4. Cetak Hasil:

- Cetak hasil konversi suhu ke Fahrenheit, Reamur, dan Kelvin.

## 2. 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).

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

**Catatan:** Gunakan `fmt.Scanf("%c", &var)` untuk pembacaan satu karakter dan `fmt.Printf("%c", var)` untuk penulisan satu karakter.

### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var a, b, c, d, e int
    var x, y, z rune

    fmt.Scanf("%d %d %d %d %d", &a, &b, &c, &d, &e)
    fmt.Scanf(" %c%c%c", &x, &y, &z)
    fmt.Printf("%c%c%c%c%c\n", a, b, c, d, e)
    fmt.Printf("%c%c%c\n", x, y, z)
}
```

## Screenshoot Output

```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\unguided2\unguided2.go"
Masukkan 5 bilangan integer: 12345
++
Masukkan 3 karakter: acb
PS C:\Aby Hakim> 
```

## Deskripsi Program

Algoritma:

1. Deklarasi Variabel:
  - Buat variabel a, b, c, d, e untuk menyimpan lima bilangan integer.
  - Buat variabel x, y, z untuk menyimpan tiga karakter.
2. Input Pengguna:
  - Baca 5 bilangan integer dari pengguna.
  - Baca 3 karakter dari pengguna.
3. Cetak Karakter ASCII:
  - Cetak representasi karakter dari lima bilangan integer.
  - Cetak tiga karakter yang telah dibaca.

Cara Kerja Program:

1. Langkah 1: Program dimulai dengan mendeklarasikan paket main dan mengimpor paket fmt untuk input dan output.
2. Langkah 2: Dalam fungsi main, program membaca lima bilangan integer dan tiga karakter dari pengguna.
3. Langkah 3: Program mencetak karakter ASCII yang sesuai untuk lima bilangan integer.
4. Langkah 4: Program mencetak tiga karakter yang telah dibaca.

### 3. Soal Studi Case

#### Sourcecode

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

func main() {
    correctSequence := []string{"merah", "kuning",
    "hijau", "ungu"}

    scanner := bufio.NewScanner(os.Stdin)

    experiments := make([][]string, 5)
    for i := 0; i < 5; i++ {
        fmt.Printf("Percobaan %d: ", i+1)
        if scanner.Scan() {
            experiments[i] =
strings.Fields(scanner.Text())
        }
    }
    success := true
    for _, experiment := range experiments {
        if !equal(experiment, correctSequence) {
            success = false
            break
        }
    }
    if success {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false")
    }
}

func equal(a, b []string) bool {
    if len(a) != len(b) {
        return false
    }
    for i := range a {
        if a[i] != b[i] {
            return false
        }
    }
    return true
}
```

## Screenshoot Output

```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\unguided3\unguided3.go"
Percobaan 1: kuning
Percobaan 2: hijau
Percobaan 3: ungu
Percobaan 4: merah
Percobaan 5: hitam
BERHASIL: false
PS C:\Aby Hakim>
```

## Deskripsi Program

Algoritma:

### 1. Deklarasi Variabel:

- Buat variabel untuk menyimpan urutan warna yang benar.
- Buat array untuk menyimpan hasil percobaan dari pengguna.

### 2. Input Pengguna:

- Minta pengguna untuk memasukkan urutan warna untuk 5 percobaan.

### 3. Validasi Urutan Warna:

- Periksa apakah setiap percobaan memiliki urutan warna yang sesuai dengan urutan yang benar.

### 4. Cetak Hasil:

- Jika semua percobaan berhasil, cetak "BERHASIL: true".
- Jika ada satu percobaan yang gagal, cetak "BERHASIL: false".

#### 4. Soal Studi Case

##### Sourcecode

```
package main

import (
    "fmt"
    "strings"
)

func gantiNamaBunga(bunga string) string {
    pengganti := map[string]string{
        "Kertas": "Tulip",
        "Mawar":  "Melati",
        "Tulip":  "Anggrek",
    }

    if namaBaru, ok := pengganti[bunga]; ok {
        return namaBaru
    }
    return bunga
}

func main() {
    for {
        var input string
        fmt.Print("Masukkan nama-nama bunga (pisahkan dengan '-'), atau ketik 'SELESAI' untuk berhenti: ")
        fmt.Scanln(&input)

        if input == "SELESAI" {
            break
        }

        bungaList := strings.Split(input, "-")

        for _, bunga := range bungaList {
            namaBaru := gantiNamaBunga(bunga)
            fmt.Printf("Nama asli: %s, Nama baru: %s\n",
                bunga, namaBaru)
        }
    }
}
```



## Screenshoot Output

```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\unguided4\main.go"
Masukkan nama-nama bunga (pisahkan dengan '-'), atau ketik 'SELESAI' untuk berhenti:
Kertas _ Mawar _ Tulip
Nama asli: Kertas, Nama baru: Tulip
Masukkan nama-nama bunga (pisahkan dengan '-'), atau ketik 'SELESAI' untuk berhenti:
Nama asli: Mawar, Nama baru: Melati
Masukkan nama-nama bunga (pisahkan dengan '-'), atau ketik 'SELESAI' untuk berhenti:
Nama asli: Tulip, Nama baru: Anggrek
Masukkan nama-nama bunga (pisahkan dengan '-'), atau ketik 'SELESAI' untuk berhenti:
```

## Deskripsi Program

### Algoritma

1. Input: Menerima input string dari pengguna.
2. Pemrosesan String: Memisahkan string input menjadi nama-nama bunga menggunakan tanda "-" sebagai pemisah.
3. Penggantian Nama Bunga: Mengganti setiap nama bunga dengan nama baru sesuai dengan aturan yang telah ditentukan.
4. Output: Menampilkan pasangan nama bunga asli dan nama baru pada baris yang terpisah.
5. Pengulangan: Mengulangi proses hingga pengguna mengetikkan "SELESAI".

### Cara Kerja Program

1. Deklarasi Variabel: Mendeklarasikan variabel untuk menyimpan input string dan hasil pemrosesan.
2. Input Pengguna: Menerima input dari pengguna berupa string nama-nama bunga.
3. Pemrosesan String: Memisahkan string input menjadi nama-nama bunga menggunakan fungsi `strings.Split`.
4. Penggantian Nama Bunga: Menggunakan map untuk mengganti nama bunga sesuai dengan aturan yang telah ditentukan.
5. Menampilkan Hasil: Menampilkan pasangan nama bunga asli dan nama baru pada baris yang terpisah.
6. Pengulangan: Mengulangi proses hingga pengguna mengetikkan "SELESAI".

## 5. Soal Studi Case

### Sourcecode

```
package main

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

func cekKeseimbangan(berat1, berat2 float64) bool {
    selisih := math.Abs(berat1 - berat2)
    return selisih <= 1
}

func main() {
    scanner := bufio.NewScanner(os.Stdin)

    for {
        fmt.Print("Masukkan berat belanjaan di kedua\nkantong (pisahkan dengan spasi), atau ketik 'SELESAI'\nuntuk berhenti: ")
        scanner.Scan()
        input := scanner.Text()

        if strings.ToUpper(input) == "SELESAI" {
            fmt.Println("Proses selesai.")
            break
        }

        var berat1, berat2 float64
        _, err := fmt.Sscanf(input, "%f %f", &berat1,
&berat2)
        if err != nil {
            fmt.Println("Input tidak valid. Silakan coba\nlagi.")
            continue
        }

        seimbang := cekKeseimbangan(berat1, berat2)
        fmt.Printf("Sepeda motor Pak Andi akan oleng:\n%t\n", !seimbang)
    }
}
```

### Screenshoot Output

```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\unguided5\main.go"
Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi), atau ketik 'SELESAI' untuk berhenti: 5.5 1.0
Sepeda motor Pak Andi akan oleng: true
Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi), atau ketik 'SELESAI' untuk berhenti: █
```

### Deskripsi Program

#### Algoritma

1. Input: Menerima input berat barang belanjaan di kedua kantong.
2. Perhitungan Selisih Berat: Menghitung selisih berat antara kedua kantong.
3. Pemeriksaan Keseimbangan: Memeriksa apakah selisih berat kurang dari atau sama dengan 1 kg.
4. Output: Menampilkan hasil pemeriksaan keseimbangan.
5. Pengulangan: Mengulangi proses hingga pengguna mengetikkan "SELESAI".

#### Cara Kerja Program

1. Deklarasi Variabel: Mendeklarasikan variabel untuk menyimpan berat barang di kedua kantong.
2. Input Pengguna: Menerima input dari pengguna berupa berat barang di kedua kantong.
3. Perhitungan Selisih Berat: Menghitung selisih berat antara kedua kantong.
4. Pemeriksaan Keseimbangan: Memeriksa apakah selisih berat kurang dari atau sama dengan 1 kg.
5. Menampilkan Hasil: Menampilkan hasil pemeriksaan keseimbangan.
6. Pengulangan: Mengulangi proses hingga pengguna mengetikkan "SELESAI".

## 6. Soal Studi Case

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

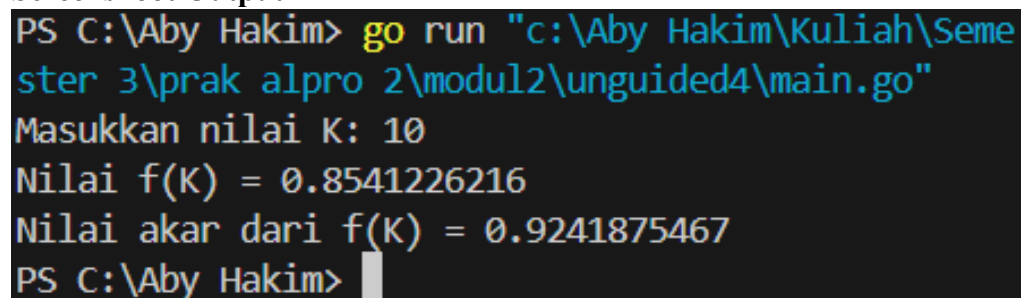
// Fungsi untuk menghitung nilai f(K)
func f(k float64) float64 {
    return (4*math.Pow(k, 2) + math.Pow(2, 2)) / ((k + 1) * (4*k + 3))
}

func main() {
    var K float64
    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&K)

    // Menghitung nilai f(K)
    hasilF := f(K)
    // Menghitung akar dari hasil f(K)
    hasilAkar := math.Sqrt(hasilF)

    // Menampilkan hasil
    fmt.Printf("Nilai f(K) = %.10f\n", hasilF)
    fmt.Printf("Nilai akar dari f(K) = %.10f\n",
        hasilAkar)
}
```

### Screenshoot Output



```
PS C:\Aby Hakim> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul2\unguided4\main.go"
Masukkan nilai K: 10
Nilai f(K) = 0.8541226216
Nilai akar dari f(K) = 0.9241875467
PS C:\Aby Hakim>
```

### Deskripsi Program

#### Algoritma

1. Input: Menerima input bilangan bulat ( K ).

2. Perhitungan Fungsi: Menghitung nilai (  $f(K)$  ) menggunakan rumus yang diberikan.
3. Perhitungan Akar: Menghitung akar dari nilai (  $f(K)$  ) menggunakan metode numerik.
4. Output: Menampilkan hasil perhitungan (  $f(K)$  ) dan akar dari (  $f(K)$  ).

#### Cara Kerja Program

1. Deklarasi Variabel: Mendeklarasikan variabel yang diperlukan untuk menyimpan nilai (  $K$  ), hasil perhitungan (  $f(K)$  ), dan akar dari (  $f(K)$  ).
2. Input Pengguna: Menerima input dari pengguna untuk nilai (  $K$  ).
3. Perhitungan (  $f(K)$  ): Menggunakan rumus (  $f(k) = \frac{4k^2 + 2^2}{(k + 1)(4k + 3)}$  ) untuk menghitung nilai (  $f(K)$  ).
4. Perhitungan Akar: Menggunakan fungsi matematika untuk menghitung akar dari (  $f(K)$  ).
5. Menampilkan Hasil: Menampilkan hasil perhitungan (  $f(K)$  ) dan akar dari (  $f(K)$  ) dengan presisi hingga sepuluh angka di belakang koma.