

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 14**  
**“KOMPOSISI”**



**DISUSUN OLEH:**  
**SAVILA NUR FADILLA**  
**103112400031**  
**S1 IF-12-01**  
**DOSEN:**  
**Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024/2025**

## **DASAR TEORI**

### **Komposisi**

Pada modul komposisi ini kita akan mempelajari soal-soal dengan tingkat kesulitan lebih, yang mana semua materi yang telah dipelajari saling dikombinasikan. Apabila sebelumnya penggunaan struktur kontrol saling dipisahkan, maka pada soal yang sekarang, memungkinkan melibatkan lebih dari satu jenis struktur kontrol.

Komposisi dalam Java antara dua objek yang saling terkait terjadi ketika ada hubungan yang kuat antara satu kelas dan kelas lainnya. Kelas lain tidak dapat ada tanpa pemilik atau kelas induk. Misalnya, kelas 'Manusia' adalah komposisi jantung dan paru-paru. Ketika objek manusia mati, tidak ada bagian yang tersisa.

Komposisi merupakan bentuk terbatas dari Agregasi. Dalam komposisi, satu kelas mencakup kelas lain dan bergantung padanya sehingga kelas tersebut tidak dapat berfungsi tanpa kelas lain.

## CONTOH SOAL

- 1.) Buatlah program yang digunakan untuk menampilkan sejumlah bilangan ganjil. Masukan terdiri dari sebuah bilangan bulat. Keluaran berupa barisan bilangan ganjil dari 1 hingga bilangan bulat yang diberikan pada masukan.

Source Code:

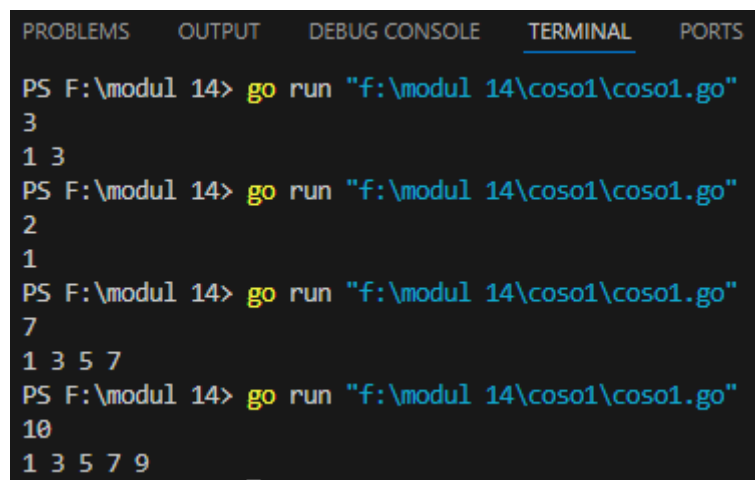
```
package main

import "fmt"

func main() {
    var bilangan, j int
    fmt.Scan(&bilangan)

    for j = 1; j <= bilangan; j += 1 {
        if j % 2 != 0 {
            fmt.Print(j, " ")
        }
    }
}
```

Output:



The screenshot shows a terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\modul 14> go run "f:\modul 14\coso1\coso1.go"
3
1 3
PS F:\modul 14> go run "f:\modul 14\coso1\coso1.go"
2
1
PS F:\modul 14> go run "f:\modul 14\coso1\coso1.go"
7
1 3 5 7
PS F:\modul 14> go run "f:\modul 14\coso1\coso1.go"
10
1 3 5 7 9
```

#### Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menampilkan sejumlah bilangan ganjil. Program meminta kita memasukkan bilangan bulat.

```
| for j = 1; j <= bilangan; j += 1 {  
|  
| }  
|
```

- Loop dimulai dari 1 dan perulangan akan terus berjalan hingga variable j sudah mencapai bilangan. Jika sudah mencapai bilangan, maka perulangan akan berhenti. Variable j += 1, nilai j akan bertambah 1 di setiap iterasi.

```
| if j % 2 != 0 {  
|     fmt.Print(j, " ")  
| }  
|
```

- Menghitung sisa hasil bagi j dengan 2, jika tidak ada sisa maka j adalah bilangan ganjil dan akan dicetak. Namun bila sisanya 0 berarti bilangan genap dan tidak akan dicetak. Keluaran berupa barisan bilangan ganjil dari 1 hingga bilangan bulat yang diberikan pada masukan.

- 2.) Buatlah program Go yang digunakan untuk mencari nilai terbesar dan terkecil antara tiga bilangan yang diberikan.  
Masukan terdiri dari 3 bilangan bulat.  
Keluaran terdiri dari dua bilangan yang menyatakan bilangan terbesar dan terkecil dari tiga bilangan yang diberikan.

Source Code:

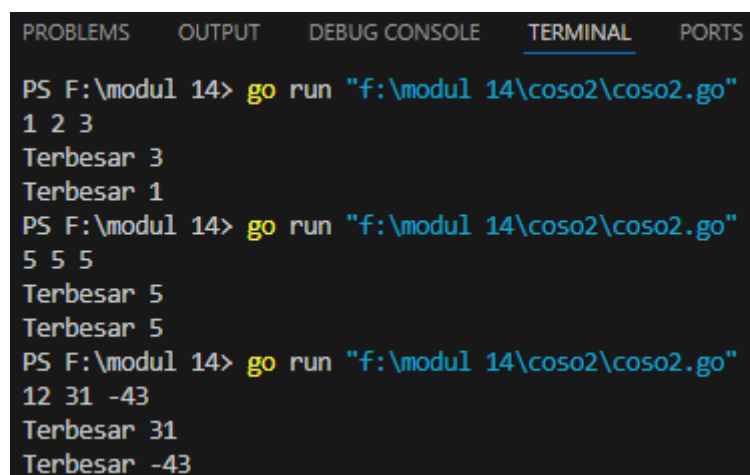
```
package main

import "fmt"

func main() {
    var b1, b2, b3, max, min int
    fmt.Scan(&b1, &b2, &b3)

    if b1 > b2 {
        max = b1
        min = b2
    } else {
        max = b2
        min = b1
    }
    if max < b3 {
        max = b3
    }
    if min > b3 {
        min = b3
    }
    fmt.Println("Terbesar", max)
    fmt.Println("Terkecil", min)
}
```

Output:



The screenshot shows a terminal window with the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS F:\modul 14> go run "f:\modul 14\coso2\coso2.go"
1 2 3
Terbesar 3
Terkecil 1
PS F:\modul 14> go run "f:\modul 14\coso2\coso2.go"
5 5 5
Terbesar 5
Terkecil 5
PS F:\modul 14> go run "f:\modul 14\coso2\coso2.go"
12 31 -43
Terbesar 31
Terkecil -43
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk mencari nilai terbesar dan terkecil antara tiga bilangan yang diberikan. Program meminta kita memasukkan 3 bilangan bulat ( $b_1$ ,  $b_2$ ,  $b_3$ ).

```
if  $b_1 > b_2$  {  
     $max = b_1$   
     $min = b_2$   
} else {  
     $max = b_2$   
     $min = b_1$   
}
```

- Jika  $b_1 > b_2$ , maka nilai  $max$  atau terbesarnya adalah  $b_1$  dan nilai  $min$  atau terkecilnya adalah  $b_2$ . Jika tidak maka nilai  $max$  adalah  $b_2$  dan  $min$  adalah  $b_1$ .

```
if  $max < b_3$  {  
     $max = b_3$   
}  
if  $min > b_3$  {  
     $min = b_3$   
}
```

- Jika nilai  $max < b_3$ , maka nilai terbesar adalah  $b_3$ . Jika nilai  $min > b_3$ , maka nilai terkecil adalah  $b_3$ .

```
 $fmt.Println("Terbesar", max)$   
 $fmt.Println("Terkecil", min)$ 
```

- Keluaran terdiri dari dua bilangan yang menyatakan bilangan terbesar dan terkecil dari tiga bilangan yang diberikan.

- 3.) Buatlah program yang digunakan untuk menampilkan faktor bilangan dari suatu nilai. Faktor bilangan dari x adalah himpunan bilangan yang habis membagi bilangan x. Masukan terdiri dari suatu bilangan bulat positif. Keluaran berupa baris bilangan yang menyatakan semua faktor bilangan dari bilangan yang diberikan pada masukan.

Source Code:

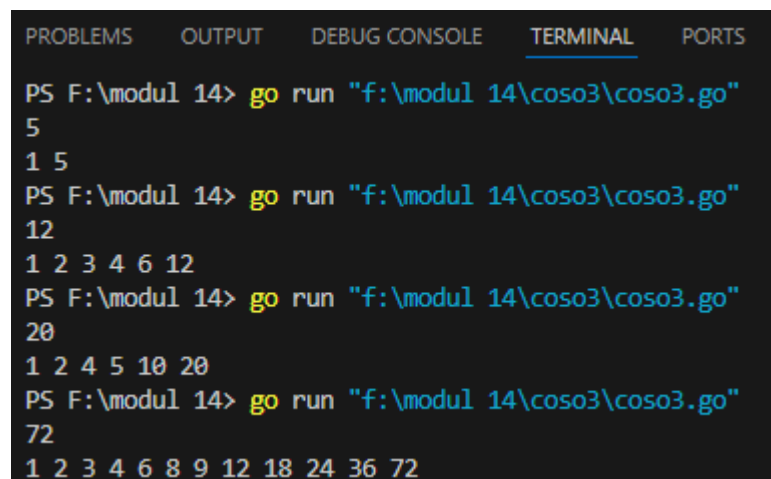
```
package main

import "fmt"

func main() {
    var bilangan, j int
    fmt.Scan(&bilangan)

    for j = 1; j <= bilangan; j += 1 {
        if bilangan % j == 0 {
            fmt.Print(j, " ")
        }
    }
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\modul 14> go run "f:\modul 14\coso3\coso3.go"
5
1 5
PS F:\modul 14> go run "f:\modul 14\coso3\coso3.go"
12
1 2 3 4 6 12
PS F:\modul 14> go run "f:\modul 14\coso3\coso3.go"
20
1 2 4 5 10 20
PS F:\modul 14> go run "f:\modul 14\coso3\coso3.go"
72
1 2 3 4 6 8 9 12 18 24 36 72
```

#### Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menampilkan faktor bilangan dari suatu nilai. Faktor bilangan dari x adalah himpunan bilangan yang habis membagi bilangan x. Program meminta kita untuk memasukkan bilangan bulat positif.

```
| for j = 1; j <= bilangan; j += 1 {  
|  
| }  
|
```

- Loop dimulai dari 1 dan perulangan akan terus berjalan hingga variable j sudah mencapai bilangan. Jika sudah mencapai bilangan, maka perulangan akan berhenti. Variable j += 1, nilai j akan bertambah 1 di setiap iterasi.

```
| if bilangan % j == 0 {  
|     fmt.Print(j, " ")  
| }  
|
```

- Menghitung sisa hasil bagi bilangan dengan j, jika tidak ada sisa atau 0 maka j adalah faktor dari bilangan dan akan dicetak. Namun bila sisanya bukan 0 berarti j bukan faktor dari bilangan dan tidak akan dicetak. Keluaran berupa baris bilangan yang menyatakan semua faktor bilangan dari bilangan yang diberikan pada masukan.



## SOAL LATIHAN

- 1.) Buatlah sebuah program Go yang digunakan untuk menghitung banyaknya bilangan ganjil dari 1 hingga n.  
Masukan terdiri dari suatu bilangan bulat positif n.  
Keluaran terdiri dari suatu teks yang menyatakan banyaknya bilangan ganjil yang terdapat antara 1 hingga n.  
Catatan: Gunakan perulangan untuk pengecekan bilangan, bukan menggunakan operasi aritmatika.

Source Code:

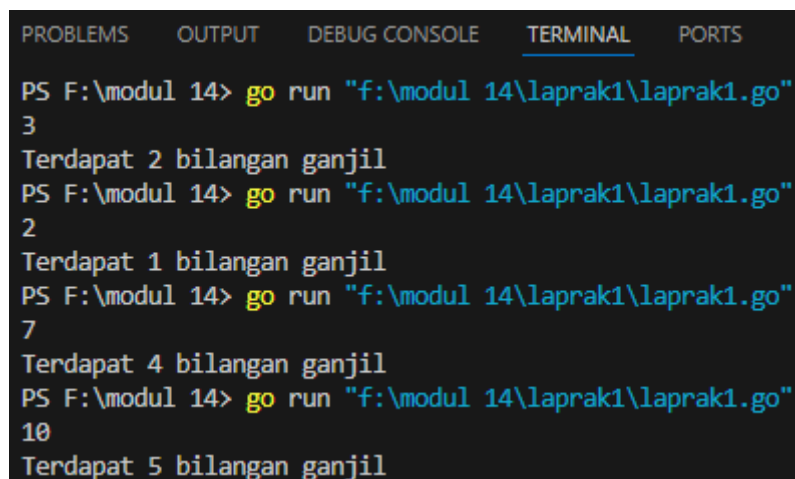
```
package main

import "fmt"

func main() {
    var n, total int
    fmt.Scan(&n)

    for i := 1; i <= n; i++ {
        if i % 2 != 0 {
            total++
        }
    }
    fmt.Printf("Terdapat %v bilangan ganjil", total)
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\modul 14> go run "f:\modul 14\laprak1\laprak1.go"
3
Terdapat 2 bilangan ganjil
PS F:\modul 14> go run "f:\modul 14\laprak1\laprak1.go"
2
Terdapat 1 bilangan ganjil
PS F:\modul 14> go run "f:\modul 14\laprak1\laprak1.go"
7
Terdapat 4 bilangan ganjil
PS F:\modul 14> go run "f:\modul 14\laprak1\laprak1.go"
10
Terdapat 5 bilangan ganjil
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menghitung banyaknya bilangan ganjil dari 1 hingga n. Program meminta kita memasukkan bilangan bulat positif n.

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

- Loop dimulai dari 1 dan perulangan akan terus berjalan hingga variable i sudah mencapai n. Jika sudah mencapai n, maka perulangan akan berhenti. Variable i++, nilai i akan bertambah 1 di setiap iterasi.

```
| if i % 2 != 0 {  
|     total++  
| }  
|
```

- Menghitung sisa hasil bagi i dengan 2, jika sisa pembagian tidak sama dengan 0, maka i adalah bilangan ganjil. Jika kondisi ini terpenuhi, maka variable total++, nilai total akan bertambah 1. Namun jika kondisi tidak terpenuhi atau sisa pembagian i dengan 2 itu sama dengan 0 (tak bersisa) maka i adalah bilangan genap sehingga variable total tidak bertambah.

```
| fmt.Printf("Terdapat %v bilangan ganjil", total)
```

- Keluaran terdiri dari suatu teks yang menyatakan banyaknya bilangan ganjil yang terdapat antara 1 sampai n.

- 2.) Sebuah program digunakan untuk menentukan sebuah bilangan adalah prima atau bukan. Bilangan dikatakan prima apabila hanya memiliki faktor yaitu satu dan bilangan itu sendiri. Sebagai catatan bilangan satu bukanlah bilangan prima. Masukan terdiri dari suatu bilangan bulat positif. Keluaran berupa teks yang menyatakan bilangan adalah "prima" atau "bukan prima".

Source Code:

```
package main

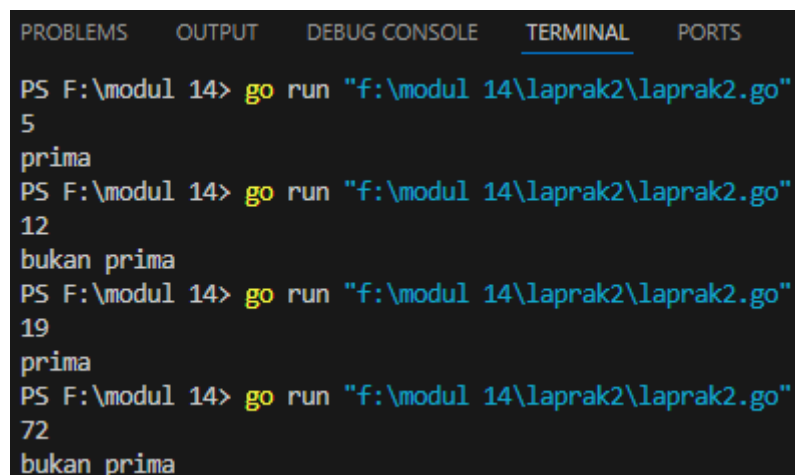
import "fmt"

func main() {
    var n, faktor int
    var teks string
    fmt.Scan(&n)

    for i := 1; i <= n; i++ {
        if n % i == 0 {
            faktor = faktor + 1
        }
    }

    if faktor == 2 {
        teks = "prima"
    } else {
        teks = "bukan prima"
    }
    fmt.Print(teks)
}
```

Output:



The screenshot shows a terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\modul 14> go run "f:\modul 14\laprak2\laprak2.go"
5
prima
PS F:\modul 14> go run "f:\modul 14\laprak2\laprak2.go"
12
bukan prima
PS F:\modul 14> go run "f:\modul 14\laprak2\laprak2.go"
19
prima
PS F:\modul 14> go run "f:\modul 14\laprak2\laprak2.go"
72
bukan prima
```

#### Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menentukan sebuah bilangan adalah prima atau bukan. Bilangan dikatakan prima apabila hanya memiliki faktor yaitu satu dan bilangan itu sendiri. Program meminta kita memasukkan bilangan bulat positif  $n$ .

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

- Loop dimulai dari 1 dan perulangan akan terus berjalan hingga variable  $i$  sudah mencapai  $n$ . Jika sudah mencapai  $n$ , maka perulangan akan berhenti. Variable  $i++$ , nilai  $i$  akan bertambah 1 di setiap iterasi.

```
| if n % i == 0 {  
|     faktor = faktor + 1  
| }  
|
```

- Menghitung sisa hasil bagi  $n$  dengan  $i$ , jika tidak ada sisa atau 0, maka kondisi terpenuhi. Jika kondisi terpenuhi, maka variable faktor akan bertambah 1.

```
| if faktor == 2 {  
|     teks = "prima"  
| else {  
|     teks = "bukan prima"  
| }  
|
```

- Menghitung jumlah faktor, jika jumlah faktor adalah 2, maka program akan mencetak teks berupa "prima". Namun jika tidak program akan mencetak teks berupa "bukan prima".

```
| fmt.Print(teks)
```

- Keluaran berupa teks yang menyatakan bilangan adalah "prima" atau "bukan prima".

- 3.) Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berturutan adalah 'merah', 'kuning', 'hijau', dan 'ungu' selama 5 kali percobaan berulang. Buatlah sebuah program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan true apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna lainnya.

Source Code:

```
package main

import "fmt"

func main() {
    const warna1 = "merah"
    const warna2 = "kuning"
    const warna3 = "hijau"
    const warna4 = "ungu"

    var total int
    var w1, w2, w3, w4 string
    var hasil bool

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %v: ", i)
        fmt.Scan(&w1, &w2, &w3, &w4)

        if w1 == warna1 && w2 == warna2 && w3 == warna3 && w4 ==
warna4 {
            total++
        }
    }
    hasil = total == 5
    fmt.Println("BERHASIL:", hasil)
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\modul 14> go run "f:\modul 14\laprak3\laprak3.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
```

Deskripsi Program:

Untuk const sendiri adalah untuk variable yang harus diinput oleh kita.

```
| for i := 1; i <= 5; i++ {  
|  
| }  
|
```

- Loop dimulai dari 1 dan perulangan akan terus berjalan hingga variable i sudah mencapai 5. Jika sudah mencapai 5, maka perulangan akan berhenti. Variable i++, nilai i akan bertambah 1 di setiap iterasi.

```
| fmt.Printf("Percobaan %v: ", i)  
| fmt.Scan(&w1, &w2, &w3, &w4)
```

- Program akan mencetak teks berupa percobaan kesekian. Lalu program meminta kita memasukkan 4 warna.

```
| if w1 == warna1 && w2 == warna2 && w3 == warna3 && w4 == warna4 {  
|     total++  
| }  
|
```

- Mengecek apakah warna yang diinput sesuai dengan const yang sudah ditetapkan. Jika sesuai maka total++, variable total akan bertambah 1.

```
| hasil = total == 5  
| fmt.Println("BERHASIL:", hasil)
```

- Mengecek apakah sudah menginput dengan total 5 dan inputan benar sesuai dengan warna yang ditentukan, Jika iya maka hasilnya true, jika tidak maka hasilnya false.

- 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. (Petunjuk: gunakan operasi penggabungan string dengan operator “+” ). Tampilkan isi pita setelah proses input selesai.”.

Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetikkan ‘SELESAI’. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita.

Source Code:

```
package main

import "fmt"

func main() {
    var jumlah int
    var bunga, all string

    for {
        fmt.Printf("Bunga %v: ", jumlah+1)
        fmt.Scan(&bunga)

        if bunga == "SELESAI" {
            break
        }
        all += bunga + " - "
        jumlah++
    }
    fmt.Println("Pita: ", all)
    fmt.Print("Bunga: ", jumlah)
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\modul 14> go run "f:\modul 14\laprak4\laprak4.go"
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Bunga 4: SELESAI
Pita: kertas - mawar - tulip -
Bunga: 3
PS F:\modul 14> go run "f:\modul 14\laprak4\laprak4.go"
Bunga 1: SELESAI
Pita:
Bunga: 0
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk mencetak pita dan jumlah bunga.

```
for {
    fmt.Printf("Bunga %v: ", jumlah+1)
    fmt.Scan(&bunga)
}

if bunga == "SELESAI" {
    break
}
```

- Loop ini akan terus berjalan sampai kondisi break. Pertama program akan mencetak bunga ke sekian dan meminta kita memasukkan nama bunga. Jika kita mengetikkan SELESAI, maka perulangan akan berhenti.

```
all += bunga + " - "
jumlah++
```

- Menambahkan “-“ setelah nama bunga saat dicetak nanti. Variable jumlah++, jumlah akan bertambah 1 setiap kita memasukkan nama bunga.

```
fmt.Println("Pita: ", all)
fmt.Print("Bunga: ", jumlah)
```

- Mencetak pita dan jumlah bunga.



- 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 riil 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.

Source Code:

```
package main

import "fmt"

func main() {
    var kantong1, kantong2, totalBerat, selisih float64
    var oleng bool

    for totalBerat <= 150 && (kantong1 >= 0 && kantong2 >= 0) {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&kantong1, &kantong2)
        if kantong2 >= kantong1 {
            selisih = kantong2 - kantong1
        } else if kantong1 >= kantong2 {
            selisih = kantong1 - kantong2
        }

        totalBerat = kantong1 + kantong2
        if totalBerat >= 150 {
            break
        }

        oleng = selisih >= 9
        fmt.Println("Sepeda motor pak Andi akan oleng:", oleng)
    }
    fmt.Println("Proses selesai.")
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\modul 14> go run "f:\modul 14\laprak5\laprak5.go"
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menentukan apakah sepeda motor oleng atau tidak.

```
| for totalBerat <= 150 && (kantong1 >= 0 && kantong2 >= 0
|
| }
|
```

- Loop akan berjalan jika total berat tidak lebih dari 150 dan berat masing masing kantong lebih dari atau sama dengan 0 (non negative). Jika salah satu kondisi bernilai false maka perulangan akan berhenti.

```
| fmt.Print("Masukan berat belanjaan di kedua kantong: ")
| fmt.Scan(&kantong1, &kantong2)
```

- Program akan mencetak teks dan meminta kita memasukkan berat belanjaan di masing masing kantong.

```
| if kantong2 >= kantong1 {
|     selisih = kantong2 - kantong1
| } else if kantong1 >= kantong2 {
|     selisih = kantong1 - kantong2
| }
|
```

- Jika berat kantong2 lebih dari atau sama dengan kantong1, maka selisihnya adalah kantong2 dikurangi kantong1. Jika tidak, maka sebaliknya (kantong1 dikurang kantong2).

```

| totalBerat = kantong1 + kantong2
| if totalBerat >= 150 {
|     break
| }

```

- Program menghitung total berat, yaitu *kantong1* dan *kantong2* dijumlahkan. Jika total berat kedua kantong lebih dari atau sama dengan 150, maka program akan berhenti.

```

| oleng = selisih >= 9
| fmt.Println("Sepeda motor pak Andi akan oleng:", oleng)

```

- Motor akan oleng jika selisih berat kedua kantong lebih dari atau sama dengan 9. Program akan menentukan apakah sepeda motor oleng atau tidak (true atau false).

```

| fmt.Println("Proses selesai.")

```

- Setelah perulangan berhenti, program mencetak teks berupa “Proses selesai.”

6.) 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.

Source Code:

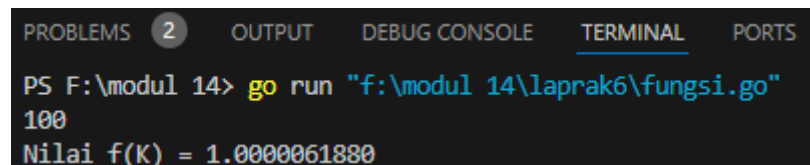
```
package main

import (
    "fmt"
    "math"
)

func main() {
    var k float64
    fmt.Scan(&k)

    fk := math.Pow((4 * k + 2), 2) / ((4 * k + 1) * (4 * k + 3))
    fmt.Printf("Nilai f(K) = %.10f\n", fk)
}
```

Output:



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS F:\modul 14> go run "f:\modul 14\laprak6\fungsi.go"
100
Nilai f(K) = 1.0000061880
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menghitung dan menampilkan nilai f(K) sesuai persamaan yang diberikan. Program meminta kita memasukkan nilai k.

```
fk := math.Pow((4 * k + 2), 2) / ((4 * k + 1) * (4 * k + 3))
fmt.Printf("Nilai f(K) = %.10f\n", fk)
```

- Program akan menghitung f(k) sesuai dengan rumus yang ditulis. Kemudian program akan mencetak nilai f(k) dengan 10 angka di belakang koma.

## DAFTAR PUSTAKA

Mathur, A. (2024). *What is Composition in Java With Examples*.

<https://www.upgrad.com/blog/what-is-composition-in-java-with-examples/>

Prasti Eko Yunanto, S. T. , M. Kom. (2004). *MODUL PRAKTIKUM 14 - KOMPOSISI ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA* .