

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II**

**Review Struktur Kontrol**



**Disusun Oleh :**

**Nadhif Atha Zaki / 2311102007**

**IF-11-05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### Apa itu Go?

Go adalah bahasa pemrograman yang dikembangkan oleh Google. Dikenal dengan sintaks yang bersih, sederhana, dan efisien, Go dirancang untuk membangun aplikasi yang handal, berskala, dan berkinerja tinggi. Go sering digunakan untuk membangun aplikasi web, layanan microservices, dan infrastruktur cloud.

### Konsep Dasar Go

- **Tipe Data:** Go memiliki tipe data yang kuat, termasuk int, float, string, bool, dan beberapa tipe data yang lebih kompleks seperti array, slice, map, dan struct.
- **Variabel:** Deklarasi variabel menggunakan var atau inferensi tipe dengan :=.
- **Kontrol Aliran:** Go mendukung struktur kontrol seperti if, else, for, dan switch untuk mengontrol alur eksekusi program.
- **Fungsi:** Fungsi adalah blok kode yang dapat dipanggil berulang kali. Go mendukung fungsi dengan parameter dan nilai kembalian.
- **Paket:** Kode Go diorganisir dalam paket. Setiap file Go harus berada dalam satu paket.
- **Concurrency:** Go mendukung concurrency dengan goroutine (lightweight thread) dan channel untuk komunikasi antar goroutine.
- **Interface:** Interface mendefinisikan sekumpulan metode yang harus diimplementasikan oleh tipe data tertentu.

### Contoh Sederhana

```
Go

package main

import "fmt"

func main() {

    var name string
```

```
    fmt.Print("Masukkan nama Anda: ")

    fmt.Scanln(&name)

    fmt.Println("Halo,", name)

}
```

Use code [with caution](#).

Kode di atas akan meminta pengguna memasukkan nama dan kemudian mencetak salam.

### Fitur Unggulan Go

- **Kompilasi Cepat:** Go mengkompilasi kode menjadi executable yang efisien.
- **Garbage Collection:** Go memiliki garbage collector otomatis untuk mengelola memori.
- **Concurrency yang Mudah:** Goroutine dan channel membuat pemrograman concurrent menjadi lebih mudah.
- **Standar Library yang Kaya:** Go memiliki standar library yang lengkap untuk berbagai keperluan.
- **Statis Typing:** Go adalah bahasa yang statis, sehingga kesalahan tipe data dapat ditemukan saat kompilasi.

### Penggunaan Go dalam Dunia Nyata

Go banyak digunakan dalam berbagai bidang, seperti:

- **Pengembangan Web:** Framework seperti Gin dan Echo populer untuk membangun aplikasi web dengan Go.
- **Cloud Computing:** Go digunakan untuk membangun layanan cloud dan infrastruktur.
- **DevOps:** Go digunakan untuk membuat tools dan utilitas DevOps.
- **Data Science:** Go dapat digunakan untuk membangun aplikasi data science yang efisien.

## II. GUIDED

1.

### Sourcecode

```
package main

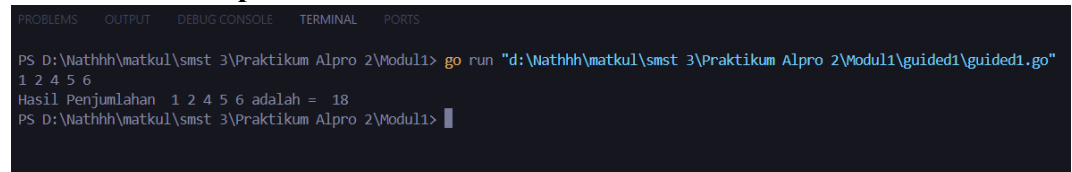
import "fmt"

func main(){

    var a, b, c, d, e int
    var hasil int
    fmt.Scanln(&a, &b, &c, &d, &e)

    hasil = a+b+c+d+e
    fmt.Println("Hasil Penjumlahan ", a,b,c,d,e, "adalah = ", hasil)
}
```

### Screenshoot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\guided1\guided1.go"
1 2 4 5 6
Hasil Penjumlahan  1 2 4 5 6 adalah =  18
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █
```

### Deskripsi Program

Program mendeklarasikan variabel a, b, c, d, e dan hasil dengan tipe data int. Lalu user diminta menginputkan value dari variabel a, b, c, d, e dengan fmt.Scanln. Lalu mengisikan variabel hasil dengan rumus  $(a + b + c + d + e)$  lalu tinggal di outputkan hasilnya

2.

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Percobaan 1:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 2:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 3:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 4:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 5:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
BERHASIL: true				
Percobaan 1:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 2:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 3:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 4:	<u>ungu</u>	<u>kuning</u>	<u>hijau</u>	<u>merah</u>
Percobaan 5:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
BERHASIL: false				

### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    // Warna yang diharapkan untuk setiap percobaan
    expected := []string{"merah", "kuning", "hijau", "ungu"}

    // Menampung status keberhasilan dari setiap percobaan
    success := true

    // Loop untuk 5 kali percobaan
```

```

for i := 1; i <= 5; i++ {
    var warna1, warna2, warna3, warna4 string
    fmt.Printf("Percobaan %d: ", i)
    fmt.Scan(&warna1, &warna2, &warna3, &warna4)

    // Memeriksa apakah input sama dengan yang
    // diharapkan
    if warna1 != expected[0] || warna2 != expected[1] ||
    warna3 != expected[2] || warna4 != expected[3] {
        success = false
    }
}

// Menampilkan hasil akhir
fmt.Printf("BERHASIL: %v\n", success)
}

```

### Screenshoot Output

```

PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\guided2\guided2.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1>

```

### Deskripsi Program

Program membuat variabel warna yang diharapkan dalam percobaan bernama `expected`, lalu penampung keberhasilan dalam setiap percobaan bernama `success`, lalu membuat perulangan untuk memasukkan percobaan 1-5. Lalu membuat pemeriksaan jika warna tidak sesuai dengan `expected` maka `success` bernilai `false`. True atau falsenya akan ditampilkan di akhir program dengan `printf`.

3.

Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian nilai mata kuliah (NMK) sebagai berikut:

NAM	NMK
NAM > 80	A
72.5 < NAM <= 80	AB

9 | Modul Praktikum Algoritma dan Pemrograman 2

65 < NAM <= 72.5	B
57.5 < NAM <= 65	BC
50 < NAM <= 57.5	C
40 < NAM <= 50	D
NAM <= 40	E

Program berikut menerima input sebuah bilangan riil yang menyatakan NAM. Program menghitung NMK dan menampilkannya.

```

1 package main
2 import "fmt"
3 func main() {
4     var nam float64
5     var nmk string
6     fmt.Print("Nilai akhir mata kuliah: ")
7     fmt.Scanln(&nam)
8     if nam > 80 {
9         nmk = "A"
10    }
11    if nam > 72.5 {
12        nmk = "AB"
13    }
14    if nam > 65 {
15        nmk = "B"
16    }
17    if nam > 57.5 {
18        nmk = "BC"
19    }
20    if nam > 50 {
21        nmk = "C"
22    }
23    if nam > 40 {
24        nmk = "D"
25    } else if nam <= 40 {
26        nmk = "E"
27    }
28    fmt.Println("Nilai mata kuliah: ", nmk)
29 }

```

ablah pertanyaan-pertanyaan berikut:

- Jika **nam** diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?
- Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!
- Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

## Sourcecode

```

package main
import "fmt"

```

```

func main(){
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nam)
    if nam > 80{
        nam = "A"
    }
    if nam > 72.5{
        nam = "AB"
    }
    if nam > 65{
        nam = "B"
    }
    if nam > 57.5{
        nam = "C"
    }
    if nam > 50{
        nam = "C"
    }
    if nam > 40{
        nam = "D"
    }
    else if nam <= 40{
        nam "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

## Screenshoot Output

```

PS D:\Wathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Wathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\guided3\tempCodeRunnerFile.go"
# command-line-arguments
guided3\tempCodeRunnerFile.go:26:2: syntax error: unexpected else, expected }
PS D:\Wathhh\matkul\smst 3\Praktikum Alpro 2\Modul1>

```

## Deskripsi Program

1. Jika nam diberikan adalah 80.1 apa keluaran dari program tersebut?

Apakah eksekusi program tersebut sesuai spesifikasi soal?

= seharusnya program akan menamulkan nilai A tapi karena program memiliki beberapa kesalahan makan program tidak dapat terseksui dengan baik

2. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!

=



- Seharusnya yang menampilkan kriteria nilainya yaitu variabel nmk bukan nam
- Seharusnya menggunakan fungsi else if setelah if, bukan if terus menerus
- Penulisan else if seharusnya sejajar dengan } pada if sebelumnya

3. Kode program yang benar dan uji coba hasilnya

Code:

```
package main
import "fmt"
func main(){
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nam)
    if nam > 80{
        nmk = "A"
    }else if nam > 72.5{
        nmk = "AB"
    }else if nam > 65{
        nmk = "B"
    }else if nam > 57.5{
        nmk = "C"
    }else if nam > 50{
        nmk= "C"
    }else if nam > 40{
        nmk = "D"
    }else {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}
```

- Memasukkan 93.5

```
PS D:\Mathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Mathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\guided3\guided3.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS D:\Mathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █
```

- Memasukkan 70.6

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run  
Nilai akhir mata kuliah: 70.6  
Nilai mata kuliah: B  
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █
```

- Memasukkan 49.5

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Nathhh\mat  
Nilai akhir mata kuliah: 49.5  
Nilai mata kuliah: D  
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █
```

### III. UNGUIDED

1.

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**

3 | Modul Praktikum Algoritma dan Pemrograman 2

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

N: <u>3</u> Bunga 1: <u>Kertas</u> Bunga 2: <u>Mawar</u> Bunga 3: <u>Tulip</u> Pita: Kertas - Mawar - Tulip -	N : 0 Pita :
---	-----------------

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 di bawah ini (**teks bergaris bawah** adalah input/read):

Bunga 1: <u>Kertas</u> Bunga 2: <u>Mawar</u> Bunga 3: <u>Tulip</u> Bunga 4: <u>SELESAI</u> Pita: Kertas - Mawar - Tulip - Bunga: 3	Bunga 1: <u>SELESAI</u> Pita : Bunga: 0
---	---

#### Sourcecode I

```
package main

import (
    "fmt"
)
```

```

func main() {
    var N int

    var pita string
    var bunga string

    // Meminta input jumlah N
    fmt.Print("Masukkan jumlah N (banyak bunga): ")
    fmt.Scanln(&N)

    // Meminta input nama bunga sebanyak N kali
    for i := 1; i <= N; i++ {
        fmt.Printf("Bunga %d: ", i)
        fmt.Scanln(&bunga)

        // Menambahkan bunga ke pita, dipisahkan dengan
        // tanda " - "
        if pita == "" {
            pita = bunga
        } else {
            pita = pita + " - " + bunga
        }
    }

    // Menampilkan hasil pita setelah proses selesai
    fmt.Println("Pita: ", pita)
}

```

### Screenshoot Output

```

PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\main.go"
Masukkan jumlah N (banyak bunga): 3
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Pita: Kertas - Mawar - Tulip
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1>

```

## Deskripsi Program

Program meminta memasukkan jumlah N banyak bunga, lalu kita membuat perulangan jika  $i \leq n$  yang diinputkan kita memasukkan nama bunga sesuai banyaknya n, lalu untuk menampilkan outputnya kita menambahkan variabel "pita" dan tanda "-"

## Sourcecode II

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var pita string
    var bunga string
    var count int

    // Meminta input nama bunga hingga pengguna mengetik
    "SELESAI"
    for {
        fmt.Print("Masukkan nama bunga (ketik 'SELESAI'
untuk selesai): ")
        fmt.Scanln(&bunga)

        // Jika user mengetik 'SELESAI', hentikan proses
input
        if strings.ToUpper(bunga) == "SELESAI" {
            break
        }

        // Menambahkan bunga ke pita, dipisahkan dengan
tanda " - "
        if pita == "" {
            pita = bunga
        } else {
            pita = pita + " - " + bunga
        }
        count++ // Menghitung jumlah bunga yang dimasukkan
    }

    // Menampilkan hasil pita dan jumlah bunga setelah
proses selesai
```

```
fmt.Println("Pita: ", pita)
fmt.Printf("Jumlah bunga: %d\n", count)
}
```

### Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Na
thhh\matkul\smst 3\Praktikum Alpro 2\Modul1\guided3\guided3.go"
Masukkan nama bunga (ketik 'SELESAI' untuk selesai): kertas
Masukkan nama bunga (ketik 'SELESAI' untuk selesai): mawar
Masukkan nama bunga (ketik 'SELESAI' untuk selesai): tulip
Masukkan nama bunga (ketik 'SELESAI' untuk selesai): selesai
Pita: kertas - mawar - tulip
Jumlah bunga: 3
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █
```

### Deskripsi Program

Program mengubah sedikit algoritmanya dengan menghilangkan penginputan N di awal program. User bisa menuliskan banyak bunga sebanyak yang dia mau dan mengetikan "selesai" jika ingin menampilkan pita nya. Dan jumlah bunga akan ditampilkan di akhir program.

2.

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

---

4 | Modul Praktikum Algoritma dan Pemrograman 2

```
Masukan berat belanjaan di kedua kantong: 5.5 1.0
Masukan berat belanjaan di kedua kantong: 7.1 8.5
Masukan berat belanjaan di kedua kantong: 2 6
Masukan berat belanjaan di kedua kantong: 9 5.8
Proses selesai.
```

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

```
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
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.
```

## Sourcecode I

```
package main

import "fmt"

func main() {
```

```

    fmt.Println("Modul Praktikum Algoritma dan Pemrograman
2")

    for {
        var beratKantong1, beratKantong2 float64

        fmt.Print("Masukan berat belanjaan di kedua kantong:
")
        fmt.Scan(&beratKantong1, &beratKantong2)

        if beratKantong1 >= 9 || beratKantong2 >= 9 {
            fmt.Println("Proses selesai.")
            break
        }
    }
}

```

### Screenshoot Output

```

PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Na
thhh\matkul\smst 3\Praktikum Alpro 2\Modul1\unguided2\unguided2.go
"
Modul Praktikum Algoritma dan Pemrograman 2
Masukan berat belanjaan di kedua kantong: 3 5
Masukan berat belanjaan di kedua kantong: 4 3
Masukan berat belanjaan di kedua kantong: 9 3
Proses selesai.
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █

```

### Deskripsi Program

Program meminta memasukkan berat di masing masing kantong, intinya jika salah satu berat lebih dari sama dengan 9kg, program akan selesai

### Sourcecode II

```

package main

import (
    "fmt"

```



```

    "math"
)

func main() {
    fmt.Println("Modul Praktikum Algoritma dan Pemrograman
2")

    for {
        var beratKantong1, beratKantong2 float64

        fmt.Print("Masukan berat belanjaan di kedua kantong:
")
        fmt.Scan(&beratKantong1, &beratKantong2)

        // Validasi input
        if beratKantong1 < 0 || beratKantong2 < 0 {
            fmt.Println("Berat tidak boleh negatif. Masukkan
ulang.")
            continue
        }

        // Hitung total berat
        totalBerat := beratKantong1 + beratKantong2

        // Hentikan program jika total berat melebihi 150 kg
        if totalBerat > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        // Hitung selisih berat
        selisihBerat := math.Abs(beratKantong1 -
beratKantong2)

        // Cek apakah sepeda motor akan oleng
        if selisihBerat >= 9 {
            fmt.Println("Sepeda motor Pak Andi akan oleng:
true")
        } else {
            fmt.Println("Sepeda motor Pak Andi akan oleng:
false")
        }
    }
}

```



### Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\unguided2\unguided2.g"
"
Modul Praktikum Algoritma dan Pemrograman 2
Masukan berat belanjaan di kedua kantong: 11 9
Sepeda motor Pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor Pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █
```

### Deskripsi Program

Program dikembangkan dengan meminta memasukkan berat kedua kantong, lalu program memvalidasi input agar tidak ada nilai negatif, program menghitung total berat kedua kantong, jika total berat lebih dari 150 kg, program berhenti, program menghitung selisih berat kedua kantong dan mengkonfirmasi apakah sepeda motor akan oleh atau tidak berdasarkan selisih berat tersebut, proses akan diulang terus menerus hingga kondisi berhenti terpenuhi

3.

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

```
Nilai K = 100
Nilai f(K) = 1.0000061880
```

### Sourcecode

```
package main

import "fmt"

func f(k int) float64 {
    return float64((4*k+2)*(4*k+2)) /
float64((4*k+1)*(4*k+3))
}

func main() {
    var k int

    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&k)

    hasil := f(k)
    fmt.Printf("Nilai f(k) = %.10f\n", hasil)
}
```

### Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d
thhh\matkul\smst 3\Praktikum Alpro 2\Modul1\unguided3\tempCode
erFile.go"
Masukkan nilai K: 100
Nilai f(k) = 1.0000061880
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █
```

### **Deskripsi Program**

Func f dibuat berisi rumus sesuai yang ada di modul, lalu di main nya kita tinggal mengisi value ke nilai K dan memanggil fungsi f

4.

PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, **buatlah program BiayaPos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut!**

Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

1	Contoh #1 Berat parcel (gram): <u>8500</u> Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500
2	Contoh #2 Berat parcel (gram): <u>9250</u> Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750
3	Contoh #3 Berat parcel (gram): <u>11750</u> Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000 + Rp. 3750 Total biaya: Rp. 110000

### Sourcecode

```
package main

import "fmt"

func hitungBiayaKirim(beratParcel int) int {
    // Menghitung berat dalam kg dan sisa gram
    beratKg := beratParcel / 1000
    sisaGram := beratParcel % 1000

    // Menghitung biaya dasar
```

```

        biayaDasar := beratKg * 10000

        // Menghitung biaya tambahan berdasarkan sisa gram
        var biayaTambahan int
        if sisaGram >= 500 {
            biayaTambahan = sisaGram * 5
        } else if sisaGram > 0 && beratKg > 10 {
            // Jika sisa gram kurang dari 500 gram dan
            total berat lebih dari 10 kg, maka gratis
            biayaTambahan = 0
        } else {
            biayaTambahan = sisaGram * 15
        }

        // Menghitung total biaya
        totalBiaya := biayaDasar + biayaTambahan
        return totalBiaya
    }

func main() {
    fmt.Println("PT POS - BlayaPos")

    for {
        var beratParsel int

        fmt.Print("Berat parsel (gram): ")
        fmt.Scan(&beratParsel)

        if beratParsel <= 0 {
            fmt.Println("Berat parsel harus
positif.")
            continue
        }

        totalBiaya := hitungBiayaKirim(beratParsel)
        fmt.Println("Total biaya: Rp.", totalBiaya)
    }
}

```

### Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\unguided4\unguided4.go"
PT POS - BlayaPos
Berat parcel (gram): 8500
Total biaya: Rp. 82500
Berat parcel (gram): 
```

### Deskripsi Program

Program ini dirancang untuk menghitung biaya pengiriman paket di PT POS. Pengguna akan diminta memasukkan berat paket dalam gram. Program kemudian akan menghitung total biaya pengiriman berdasarkan ketentuan yang telah ditetapkan oleh PT POS. Ketentuan ini mencakup biaya dasar per kilogram dan biaya tambahan untuk sisa gram yang melebihi 500 gram. Program juga mempertimbangkan pengecualian untuk paket dengan total berat lebih dari 10 kg dan sisa gram kurang dari 500 gram. Hasil perhitungan berupa total biaya pengiriman akan ditampilkan kepada pengguna. Program ini dapat digunakan berulang kali untuk menghitung biaya pengiriman paket dengan berbagai berat.

5. S

Sebuah bilangan bulat **b** memiliki faktor bilangan **f** > 0 jika **f** habis membagi **b**. Contoh: 2 merupakan faktor dari bilangan 6 karena 6 habis dibagi 2.

Buatlah program yang menerima input sebuah bilangan bulat **b** dan **b** > 1. Program harus dapat mencari dan menampilkan semua faktor dari bilangan tersebut!

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bilangan: <u>12</u> Faktor: 1 2 3 4 6 12	Bilangan: <u>7</u> Faktor: 1 7
---	-----------------------------------

Bilangan bulat **b** > 0 merupakan bilangan prima **p** jika dan hanya jika memiliki persis dua faktor bilangan saja, yaitu 1 dan dirinya sendiri.

Lanjutkan program sebelumnya. Setelah menerima masukan sebuah bilangan bulat **b** > 0. Program tersebut mencari dan menampilkan semua faktor bilangan tersebut. Kemudian, program menentukan apakah **b** merupakan bilangan prima.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bilangan: <u>12</u> Faktor: 1 2 3 4 6 12 Prima: <u>false</u>	Bilangan: <u>7</u> Faktor: 1 7 Prima: <u>true</u>
--	---

## Sourcecode I

```
package main

import "fmt"

func main() {
    var bilangan int

    fmt.Print("Bilangan: ")
    fmt.Scan(&bilangan)

    if bilangan <= 1 {
        fmt.Println("Bilangan harus lebih besar dari 1.")
        return
    }
}
```



```

    fmt.Print("Faktor: ")
    for i := 1; i <= bilangan; i++ {
        if bilangan%i == 0 {
            fmt.Print(i, " ")
        }
    }
    fmt.Println()
}

```

### Screenshoot Output

```

PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run main.go
Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\unguided5\unguided5.go
Bilangan: 20
Faktor: 1 2 4 5 10 20
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> 

```

### Deskripsi Program

Program ini dirancang untuk menemukan semua faktor dari sebuah bilangan bulat positif. Pengguna akan diminta memasukkan sebuah bilangan bulat. Program kemudian akan melakukan perhitungan untuk mencari bilangan-bilangan yang dapat membagi habis bilangan tersebut tanpa sisa. Bilangan-bilangan yang memenuhi kriteria tersebut disebut sebagai faktor. Semua faktor yang ditemukan akan dicetak di layar. Program ini berguna untuk memahami konsep faktorisasi bilangan bulat dan dapat digunakan sebagai alat bantu dalam mempelajari teori bilangan.

### Sourcecode II

```

package main

import "fmt"

func main() {
    var bilangan int

```

```
fmt.Print("Bilangan: ")
fmt.Scan(&bilangan)

if bilangan <= 1 {
    fmt.Println("Bilangan harus lebih besar dari 1.")
    return
}

fmt.Print("Faktor: ")
var jumlahFaktor int = 0
for i := 1; i <= bilangan; i++ {
    if bilangan%i == 0 {
        fmt.Print(i, " ")
        jumlahFaktor++
    }
}
fmt.Println()

if jumlahFaktor == 2 {
    fmt.Println("Prima: true")
} else {
    fmt.Println("Prima: false")
}
}
```

**Screenshoot Output**

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\unguided5\tempCodeRunnerFile.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> go run "d:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1\unguided5\tempCodeRunnerFile.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul1> █
```

### Deskripsi Program

Program ini dirancang untuk menentukan apakah sebuah bilangan bulat adalah bilangan prima atau bukan. Setelah pengguna memasukkan sebuah bilangan, program akan mencari dan mencetak semua faktor dari bilangan tersebut. Sembari mencari faktor, program juga menghitung jumlah faktor yang ditemukan. Jika jumlah faktor yang ditemukan hanya dua (yaitu 1 dan bilangan itu sendiri), maka bilangan tersebut dinyatakan sebagai bilangan prima. Hasil akhir yang ditampilkan adalah daftar faktor dan status apakah bilangan tersebut prima atau bukan