

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL II
REVIEW STRUKTUR KONTROL



Oleh:

ANISSA FAUZIA ISYANTI

2311102219

S1IF-11-02

S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024

I. DASAR TEORI

A. Struktur Program Go

Dalam kerangka program yang ditulis dalam bahasa pemrograman Go, program utama selalu mempunyai dua komponen berikut:

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

Untuk membuat komentar pada pemrograman Go dapat menggunakan garis miring ganda (“//”) atau beberapa baris dimulai dengan pasangan karakter “/*” dan diakhiri “*/”.

Semua proses terkait bahasa Go dilakukan melalui utilitas go. Beberapa opsi dengan utilitas go:

- **go build**: mengkompilasi program sumber yang ada dalam folder menjadi sebuah program.
- **go build file.go**: mengkompilasi program sumber file.go saja
- **go fmt**: membacaa semua program sumber dalam folder dan mereformat penulisannya agar sesuai dengan standar penulisan program sumber Go.
- **go clean**: membersihkan file-file dalam folder sehingga tersisa program sumbernya saja.

B. Tipe data

Notasi tipe dasar	Tipe dalam Go	Keterangan
integer	int int8 int32 //rune int64 uint uint8 //byte uint32 uint64	bergantung platform 8 bit: -128..127 32 bit: -10 ⁹ ..10 ⁹ 64 bit: -10 ¹⁹ ..10 ¹⁹ bergantung platform 0..255 0..4294967295 0..(2 ⁶⁴ -1)
real	float32 float64	32bit: -3.4E+38 .. 3.4E+38 64bit: -1.7E+308 .. 1.7E+308
boolean (atau logikal)	bool	false dan true
karakter	byte //uint8 rune //int32	tabel ASCII/UTF-8 tabel UTF-16
string	string	

Operasi yang dapat dilakukan terhadap tipe data di atas adalah:

Operator dalam Go	Tipe data terkait	Keterangan
+	string integer dan real	konkatenasi 2 string operasi penjumlahan
- * /	integer dan real	operasi pengurangan, perkalian, dan pembagian
%	integer	operasi sisa pembagian integer (modulo)
& ^ &^	integer	operasi per-bit AND, OR, XOR, AND-NOT
<< >>	integer dan unsigned integer	operasi geser bit kiri/kanan sebanyak unsigned integer yang diberikan
< <= >= > == !=	selain boolean	komparasi menghasilkan nilai boolean komparasi karakter sesuai dengan posisi karakter tersebut dalam tabel ASCII/UTF-16 komparasi string sesuai dengan operasi karakter per karakter, dimulai dari karakter paling kiri (awal)
&& !	boolean	operasi boolean AND, OR, dan NOT
* &	variabel apa saja	mendapatkan data dari lokasi memori dan mendapatkan lokasi dari variabel

C. Struktur Kontrol Perulangan

for inisialisasi; kondisi; update { // .. for-loop ala C // .. ke-3 bagian opsional, tetapi ";" tetap harus ada }
for kondisi { // .. ulangi kode di sini selama kondisi terpenuhi // .. sama seperti "for ; kondisi; {" }
for { // .. tanpa kondisi, berarti loop tanpa henti (perlu if-break) }
for ndx, var := range slice/array { // .. iterator mengunjungi seluruh isi slice/array // .. pada setiap iterasi ndx diset indeks dan var diisi nilainya }

D. Struktur Kontrol Percabangan

Bentuk If-Else

	Notasi algoritma	Penulisan dalam bahasa Go
1 2 3	if (kondisi) then .. kode untuk kondisi true endif	if kondisi { .. kode untuk kondisi true }
4 5 6 7 8	if (kondisi) then .. kode untuk kondisi true else .. kode untuk kondisi false endif	if kondisi { .. kode untuk kondisi true } else { .. kode untuk kondisi false }
9 10 11 12 13 14 15 16 17	if (kondisi-1) then .. kode untuk kondisi-1 true else if (kondisi-2) then .. kode untuk kondisi-2 true .. dst. dst. else .. kode jika semua kondisi .. di atas false endif	if kondisi_1 { .. kode untuk kondisi_1 true } else if kondisi_2 { .. kode untuk kondisi_2 true .. dst. dst. } else { .. kode jika semua kondisi .. di atas false }

Bentuk Switch-Case

	Notasi algoritma	Penulisan dalam bahasa Go
1 2 3 4 5 6 7 8 9 10	depend on ekspresi nilai_1: .. kode jika ekspresi bernilai_1 nilai_2: .. kode jika ekspresi bernilai_2 .. dst. dst. }	switch ekspresi { case nilai_1: .. kode jika ekspresi bernilai_1 case nilai_2: .. kode jika ekspresi bernilai_2 .. dst. dst. default: .. kode jika tidak ada nilai .. yang cocok dengan ekspresi }
11 12 13 14 15 16 17 18 19 20	depend on (daftar variabel) kondisi_1: .. kode jika ekspresi_1 true kondisi_2: .. kode jika ekspresi_2 true .. dst. dst. }	switch { case kondisi_1: .. kode jika ekspresi_1 true case kondisi_2: .. kode jika ekspresi_2 true .. dst. dst. default: .. jika tidak ada ekspresi .. yang bernilai true }

II. GUIDED

A. Soal Latihan Modul 2A

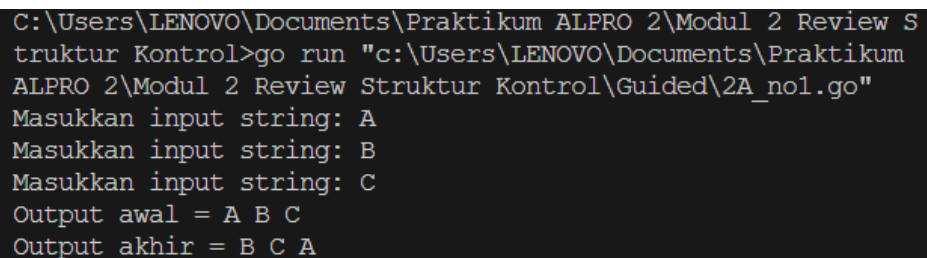
1. Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

```
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)
}
```

Screenshot Program



```
C:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review S
truktur Kontrol>go run "c:\Users\LENOVO\Documents\Praktikum
ALPRO 2\Modul 2 Review Struktur Kontrol\Guided\2A_no1.go"
Masukkan input string: A
Masukkan input string: B
Masukkan input string: C
Output awal = A B C
Output akhir = B C A
```

Di awal program meminta pengguna untuk memasukkan tiga buah string dan disimpan di variabel satu, dua, tiga. Setelah input diterima, program akan menampilkan ketiga string tersebut sebagai output awal. Kemudian,

program melakukan pertukaran nilai string, string pertama diubah menjadi string kedua, string kedua diubah menjadi string ketiga, dan string ketiga diubah menjadi string pertama. Setelah pertukaran, program akan menampilkan hasilnya sebagai output akhir.

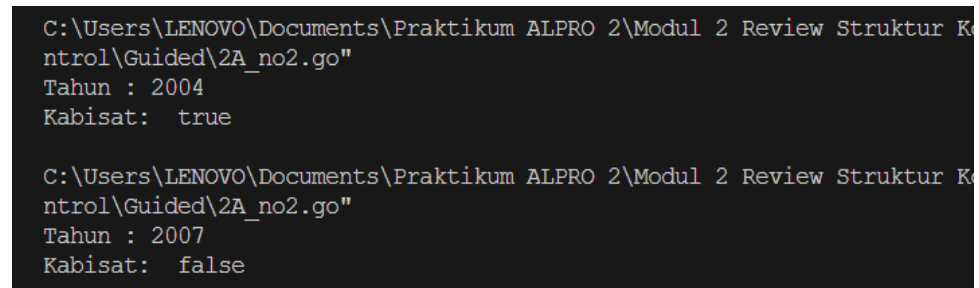
2. Tahun Kabisat adalah tahun yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat (**true**) atau bukan (**false**).

```
package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Tahun : ")
    fmt.Scan(&tahun)
    fmt.Println("Kabisat: ", (tahun%400 == 0 ||
tahun%4 == 0 && tahun%100 != 0))
}
```

Screenshot Program



```
C:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review Struktur K
ontrol\Guided\2A_no2.go"
Tahun : 2004
Kabisat: true

C:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review Struktur K
ontrol\Guided\2A_no2.go"
Tahun : 2007
Kabisat: false
```

Program ini merupakan program pengecekan apakah tahun tersebut merupakan tahun kabisat atau bukan. Di awal program, user diminta untuk menginputkan tahun yang disimpan di variabel **tahun**. Selanjutnya program akan melakukan pengecekan, tahun kabisat adalah tahun yang habis dibagi 4 atau 400, tetapi tidak habis dibagi 100. Setelah memeriksa kondisi tersebut, program akan menampilkan `true` jika tahun tersebut kabisat, dan `false` jika tidak.

3. Buatlah 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 $luasbola = 4\pi r^2$ ($\pi = 3.1415926535$).

```
package main

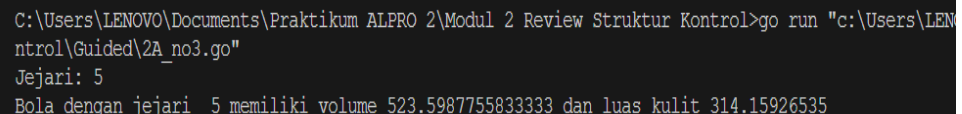
import (
    "fmt"
    "math"
)

func main() {
    const phi = 3.1415926535
    var jari int
    fmt.Print("Jejari: ")
    fmt.Scan(&jari)

    volumeBola := (4.0 / 3.0) * phi *
math.Pow(float64(jari), 3)
    luasBola := 4 * phi * math.Pow(float64(jari), 2)

    fmt.Println("Bola dengan jejari ", jari,
"memiliki volume", volumeBola, "dan luas kulit",
luasBola)
}
```

Screenshot Program



```
C:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review Struktur Kontrol>go run "c:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review Struktur Kontrol\Guided\2A_no3.go"
Jejari: 5
Bola dengan jejari 5 memiliki volume 523.5987755833333 dan luas kulit 314.15926535
```

Program ini digunakan untuk menghitung volume dan luas permukaan bola berdasarkan jari-jari yang diinputkan user. User diminta menginputkan jari-jari dalam bilangan bulat, kemudian program akan melakukan perhitungan sesuai rumus volume bola dan luas kulit. Selanjutnya program akan mencetak hasil perhitungan.

III. UNGUIDED

A. Soal Latihan Modul 2B

1. 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 berurutan 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.

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    urutanBenar := [4]string{"merah", "kuning",
    "hijau", "ungu"}
    berhasil := true

    for i := 0; i < 5; i++ {
        fmt.Printf("Percobaan %d: ", i+1)
        for j := 0; j < 4; j++ {
            var warna string
            fmt.Scan(&warna)
            if strings.ToLower(warna) !=
urutanBenar[j] {
                berhasil = false
            }
        }
    }

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


Screenshot Program

```
C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Ani
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning ungu hijau
Percobaan 5: merah kuning hijau ungu
BERHASIL: false

C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Ani
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
```

Program ini meminta pengguna untuk memasukkan susunan warna dari empat tabung reaksi dalam lima percobaan. Urutan warna yang benar adalah "merah", "kuning", "hijau", dan "ungu". Program akan memeriksa setiap percobaan, dan jika semua percobaan memiliki urutan warna yang tepat, program akan menampilkan bahwa percobaan berhasil. Jika ada satu percobaan dengan urutan warna yang salah, program akan menampilkan bahwa percobaan tidak berhasil. Hasil akhir ditampilkan dalam bentuk "BERHASIL: true" jika semua percobaan benar, atau "BERHASIL: false" jika ada yang salah.

2. 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.

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

```

package main

import "fmt"

func main() {
    var bunga string
    var banyak int

    fmt.Print("N : ")
    fmt.Scan(&banyak)

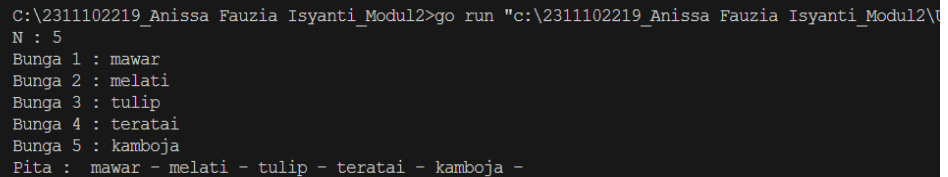
    pita := ""

    for i := 0; i < banyak; i++ {
        fmt.Print("Bunga ", i+1, " : ")
        fmt.Scan(&bunga)
        pita += bunga + " - "
    }

    fmt.Println("Pita : ", pita)
}

```

Screenshot Program



```

C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Anissa Fauzia Isyanti_Modul2\
N : 5
Bunga 1 : mawar
Bunga 2 : melati
Bunga 3 : tulip
Bunga 4 : teratai
Bunga 5 : kamboja
Pita : mawar - melati - tulip - teratai - kamboja -

```

Program ini meminta pengguna untuk memasukkan jumlah bunga yang ingin disimpan dalam pita. Setelah pengguna memasukkan jumlah bunga , program akan meminta nama setiap bunga sebanyak jumlah yang dimasukkan di awal. Nama-nama bunga yang sudah dimasukkan akan digabungkan menjadi sebuah pita yang dipisahkan dengan tanda “ – “. Di akhir program akan ditampilkan hasil gabungan nama bunga .

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

```

package main

import "fmt"

```

```

func main() {
    var bunga string
    var banyak int

    pita := ""

    for {
        fmt.Print("Bunga ", banyak+1, " : ")
        fmt.Scan(&bunga)

        if bunga == "SELESAI" {
            break
        }
        pita += bunga + " - "
        banyak++
    }

    fmt.Println("Pita: ", pita)
    fmt.Println("Jumlah bunga: ", banyak)
}

```

Screensshot Program

```

C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Anissa
Bunga 1 : mawar
Bunga 2 : melati
Bunga 3 : SELESAI
Pita: mawar - melati -
Jumlah bunga: 2

```

Program ini sama seperti program sebelumnya, hanya saja ketika pengguna memasukkan nama bunga “SELESAI” maka program akan berhenti meminta pengguna memasukkan nama bunga. Selanjutnya program akan mencetak pita dan jumlah bunga.

3. 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 9kg. 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 9kg atau lebih.

```
package main

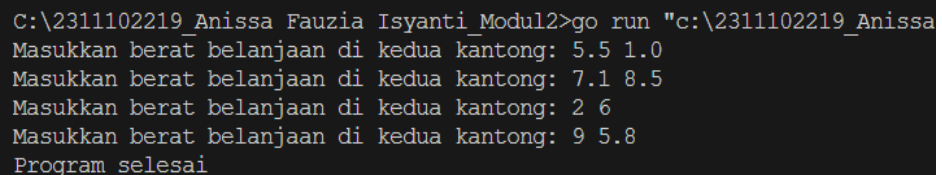
import (
    "fmt"
)

func main() {
    var kantongKiri, kantongKanan float64

    for {
        fmt.Printf("Masukkan berat belanjaan di kedua kantong: ")
        fmt.Scan(&kantongKiri, &kantongKanan)

        if kantongKiri >= 9 || kantongKanan >= 9 {
            fmt.Printf("Program selesai")
            break
        }
    }
}
```

Screenshot Program



```
C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Anissa
Masukkan berat belanjaan di kedua kantong: 5.5 1.0
Masukkan berat belanjaan di kedua kantong: 7.1 8.5
Masukkan berat belanjaan di kedua kantong: 2 6
Masukkan berat belanjaan di kedua kantong: 9 5.8
Program selesai
```

Program ini dirancang untuk membantu Pak Andi dalam mengawasi berat belanjaan yang dibawa di kantong-kantong sepeda motornya. Pengguna diminta untuk memasukkan berat barang di kantong kiri dan kanan. Program akan terus meminta input hingga berat di salah satu kantong mencapai 9 kg atau lebih. Ketika salah satu kantong mencapai berat tersebut, program akan menampilkan pesan "Program selesai" dan berhenti berfungsi.

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.

```
package main

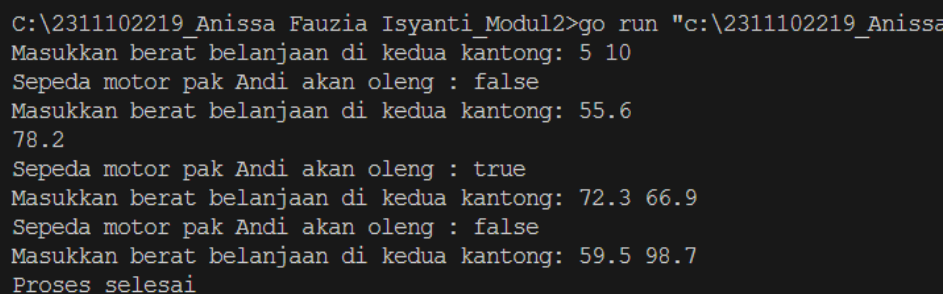
import (
    "fmt"
    "math"
)

func main() {
    var kantongKiri, kantongKanan float64

    for {
        fmt.Printf("Masukkan berat belanjaan di  
kedua kantong: ")
        fmt.Scan(&kantongKiri, &kantongKanan)

        if !((kantongKiri+kantongKanan > 150) ||  
kantongKiri < 0 || kantongKanan < 0) {
            fmt.Println("Sepeda motor pak Andi  
akan oleng :", math.Abs(kantongKiri-kantongKanan) >=
9)
        } else {
            fmt.Println("Proses selesai")
            break
        }
    }
}
```

Screenshot Program



```
C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Anissa
Masukkan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng : false
Masukkan berat belanjaan di kedua kantong: 55.6
78.2
Sepeda motor pak Andi akan oleng : true
Masukkan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng : false
Masukkan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai
```

Program ini sama seperti program sebelumnya, hanya saja pada program ini akan terus meminta input sampai total berat isi kedua kantong melebihi 150 kg atau salah satu kantong memiliki berat negatif. Jika kondisi tersebut tidak terpenuhi, program akan menghitung selisih berat antara kedua kantong dan menampilkan apakah selisih tersebut lebih besar atau sama dengan 9 kg.

Jika selisih berat mencapai atau melebihi 9 kg, program akan mengindikasikan bahwa sepeda motor Pak Andi akan oleng; jika tidak, program akan menginformasikan sebaliknya. Ketika syarat untuk menghentikan program terpenuhi, program akan menampilkan pesan "Proses selesai."

4. Diberikan sebuah persamaan sebagai berikut ini.

$$f(k) = \frac{(4k + 2)^2}{(4k + 1)(4k + 3)}$$

Buatlah sebuah rogram yang menerima input sebuah bilangan sebagai **K**, kemudian menghitung dan menampilkan nilai (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
```

```
package main

import (
    "fmt"
    "math"
)

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

    hasil := math.Pow(4*float64(k)+2, 2) /
    ((4*float64(k) + 1) * (4*float64(k) + 3))
    fmt.Printf("Nilai f(K) = %.10f \n", hasil)
}
```

Screenshot Program

```
C:\2311102219_Anissa Fauzia Isyanti_Modul2>go r
Nilai K = 100
Nilai f(K) = 1.0000061880
```

Program ini untuk menghitung nilai dari fungsi matematis berdasarkan masukan bilangan bulat “k” yang diberikan oleh pengguna. Pertama, pengguna diminta untuk memasukkan nilai “k”, yang kemudian digunakan dalam rumus tertentu. Rumus tersebut menghitung hasil dengan cara mengkuadratkan ekspresi $(4k + 2)$ dan membaginya dengan hasil kali dari $(4k + 1)$ dan $(4k + 3)$. Setelah melakukan perhitungan, program menampilkan hasilnya dengan format desimal yang presisi hingga sepuluh angka di belakang koma.

$\sqrt{2}$ merupakan bilangan irasional. Meskipun demikian, nilai tersebut dapat dihampiri dengan rumus berikut:

$$\sqrt{2} = \prod_{k=0}^{\infty} \frac{(4k + 2)^2}{(4k + 1)(4k + 3)}$$

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.

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var k int
    fmt.Print("Nilai K = ")
    fmt.Scan(&k)
    hasil := 1.0

    for i := 0; i <= k; i++ {
        hasil *= math.Pow(4*float64(i)+2, 2) /
        ((4*float64(i) + 1) * (4*float64(i) + 3))
    }

    fmt.Printf("\nNilai Akar 2 = %.10f \n", hasil)
}
```

Screenshot Program

```
C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Anissa Fauzia Isyanti_Modul2\modul2.go"
Nilai K = 10

Nilai Akar 2 = 1.4062058441
```

Program ini meminta pengguna untuk memasukkan sebuah bilangan bulat yang disebut sebagai K. Selanjutnya, program menghitung nilai hasil berdasarkan rumus matematis yang melibatkan perhitungan dengan loop dari 0 hingga K. Dalam setiap iterasi, program menghitung nilai tertentu menggunakan rumus yang melibatkan bilangan i, dan hasilnya dikalikan untuk mendapatkan nilai akhir. Setelah menyelesaikan semua perhitungan, program menampilkan nilai akhir dengan ketelitian sepuluh angka di belakang koma, yang mewakili hasil dari operasi matematika tersebut. Dengan kata lain, program ini menghitung suatu nilai yang mendekati akar dua menggunakan metode perhitungan yang spesifik.

B. Soal Latihan Modul 2C

1. 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) ddigratiskan biayanya apabila total berat ternyata lebih dari 10kg.

```
package main

import "fmt"

func main() {
```



```

var berat, Kg, gram, BiayaDasar, tambahan, total
int

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

Kg = berat / 1000
gram = berat % 1000

BiayaDasar = Kg * 10000

if Kg > 10 {
    tambahan = 0
} else {
    if gram >= 500 {
        tambahan = gram * 5
    } else {
        tambahan = gram * 15
    }
}

total = BiayaDasar + tambahan

fmt.Printf("Detail berat: %d kg + %d gr\n", Kg,
gram)
fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n",
BiayaDasar, tambahan)
fmt.Printf("Total biaya : Rp %d\n", total)
}

```

Screenshot Program

```

C:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review Struktur Kontrol>go r
ntrol\Unguided\2C_nol.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya : Rp 82500

```

Program ini dirancang untuk menghitung biaya pengiriman parsel berdasarkan berat dalam satuan gram. Pengguna diminta memasukkan berat parsel dalam gram yang disimpan di variabel **berat**, kemudian akan dikonversi menjadi kilogram(**Kg**) dan sisa gram (**gram**). Biaya dasar(**BiayaDasar**) dihitung berdasarkan kilogram (setiap kilogram dikenakan Rp.10000). Selanjutnya biaya tambahan (**tambahan**) dihitung jika berat sisa gram mencapai 500gram atau lebih, maka dikenakan Rp.5 per gram, jika kurang dari 500 gram, dikenakan Rp.15 per gram. Namun jika

berat parcel lebih dari 10 kilogram, tidak dikenakan biaya tambahan. Di akhir program ditampilkan total biaya (**total**) yang didapatkan dari biaya dasar + biaya tambahan.

2. 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
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.

```
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 = "BC"
    }
    if nam > 50 {
        nam = "C"
    }
    if nam > 40 {
        nam = "D"
    } else if nam <= 40 {
        nam = " E"
    }
}
```

```
        fmt.Println("Nilai mata kuliah: ", nmk)
    }
```

Jawablah pertanyaan-pertanyaan berikut:

- a. Jika **nam** diberikan adalah 80.1, apa keluaran dari program tersebut?
Apakah eksekusi program tersebut sesuai spesifikasi soal?
= Program akan mengalami kesalahan (error) dan tidak dapat dijalankan.
Karena kesalahan tipe data dari variabel nam yang dideklarasikan bertipe data float64, tidak dapat menyimpan nilai string seperti 'A', 'AB', 'C', 'D, dan 'E'.
- b. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!
= **Kesalahan penggunaan tipe data** dimana variabel nam dideklarasikan float64 seharusnya tidak diisi dengan nilai string, **kesalahan penggunaan kontrol perabangan** pada program diatas semua kondisi if tidak menggunakan else if sehingga ketika satu kondisi terpenuhi, program masih akan memeriksa kondisi berikutnya.
- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

```
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 = "BC"
    } else if nam > 50 {
        nmk = "C"
    } else if nam > 40 {
        nmk = "D"
    } else {
        nmk = "E"
    }
}
```

```

    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

Screenshot Program

```

r Kontrol>go run "c:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review Struk
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A

C:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review Struktur Kontrol>go run
2C no2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B

C:\Users\LENOVO\Documents\Praktikum ALPRO 2\Modul 2 Review Struktur Kontrol>go run
2C no2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D

```

Program ini digunakan untuk menentukan nilai huruf berdasarkan nilai akhir mata kuliah yang dimasukkan pengguna. Mula-mula program akan meminta pengguna memasukkan nilai akhir berupa desimal. Berdasarkan nilai tersebut, program akan mengecek nilai yang dimasukkan masuk ke dalam rentang nilai mana, jika nilai lebih dari 80 akan mendapatkan “A”, jika lebih dari 72.5, maka “AB”, jika lebih dari 65 diberi nilai “B”, lebih dari 57.5 mendapatkan “BC”, lebih dari 50 diberi “C”, lebih dari 40 diberikan “D” dan nilai dibawah atau sama dengan 40 mendapatkan “E”. Program akan melakukan pengecekan hingga salah satu kondisi terpenuhi. Setelah pengecekan, program akan menampilkan nilai huruf.

3. 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!

```

package main

import (
    "fmt"
)

func main() {
    var b int

```

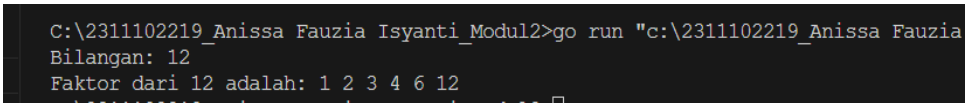
```

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

    fmt.Printf("Faktor dari %d adalah: ", b)
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Printf("%d ", i)
        }
    }
}

```

Screenshot Program



```

C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Anissa Fauzia
Bilangan: 12
Faktor dari 12 adalah: 1 2 3 4 6 12

```

Program ini menerima input berupa bilangan bulat dari pengguna, kemudian mencari dan menampilkan semua faktor dari bilangan tersebut. Faktor-faktor ini adalah bilangan yang habis membagi bilangan yang diinput tanpa sisa. Program menggunakan loop untuk memeriksa setiap angka dari 1 hingga bilangan tersebut, dan mencetak angka-angka yang merupakan faktor.

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.

```

package main

import (
    "fmt"
)

func main() {
    var b int
    var banyakFaktor int

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

    if b <= 1 {

```

```

        fmt.Println("Bilangan harus lebih dari 1.")
        return
    }

    fmt.Print("Faktor: ")
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Printf("%d ", i)
            banyakFaktor++
        }
    }
    fmt.Print("\nPrima: ", banyakFaktor == 2, "\n")
}

```

Screenshot Program

```

C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_Anissa F
Bilangan (b > 1): 7
Faktor: 1 7
Prima: true

```

```

C:\2311102219_Anissa Fauzia Isyanti_Modul2>go run "c:\2311102219_An
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false

```

Program ini sama seperti program sebelumnya, untuk mengecek faktor bilangan. Setelah itu, program akan mencari dan mencetak semua faktor bilangan, yaitu angka-angka yang dapat membagi bilangan tersebut tanpa sisa. Selain itu, program juga menghitung jumlah faktor dan menentukan apakah bilangan tersebut merupakan bilangan prima, yaitu bilangan yang hanya memiliki dua faktor (1 dan dirinya sendiri). Hasilnya adalah daftar faktor dan informasi apakah bilangan tersebut prima.