

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 2
REVIEW STRUKTUR KONTROL



Disusun Oleh:

NAMA: MARIA DWI A

NIM: 2311102228

KELAS : S1- 1F – 11 - 02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

1. 1 Stuktur Program Go

Dalam bahasa pemograman Go, program utama memiliki dua komponen yaitu :

- **package main** : merupakan indicator yang menyatakan bahwa file berisi program utama
- **func main()** : isi dari kode utama dalam program Go.

Dalam bahasa Go, proses dijalankan melalui utilitas antara lain :

- **go build** : melakukan kompilasi program sumber yang berada pada folder menjadi sebuah program
- **go build file.go** : melakukan kompilasi program sumber file.go saja
- **go fmt** : untuk membaca semua program sumber di dalam folder dan mengubah format penulisannya agar sesuai dengan standar penulisan program sumber Go.
- **go clean** : untuk membersihkan file-file dalam folder agar tersisa program sumber nya saja.

1.2 Tipe Data dan Instruksi Dasar

1) Data dan Variabel

Tipe data merupakan pengelompokan data di dalam pemograman komputer sehingga dapat melakukan proses komputasi.

- Nama variabel diawali dengan huruf dan diikuti dengan sejumlah huruf, angka, atau garis bawah.

Nama Type	Nilai	Alokasi memori	Jangkauan ukuran
boolean	true or false	1 byte	not applicable
char	single character (unicode)	2 bytes	all unicodes characters
byte	integer	1 byte	-128 to 127
short	integer	2 bytes	-32768 to 32767
int	integer	4 bytes	-2147483648 to 2147483647
long	integer	8 bytes	-9223372036854775808 to 9223372036854775807
float	floating-point number	4 bytes	$-3.40282347 \times 10^{+38}$ to $1.40239846 \times 10^{-45}$
double	floating-point number	8 bytes	$\pm 1.76769313486231570 \times 10^{+308}$ to $\pm 4.94065645841246544 \times 10^{-324}$

- Tipe data yang umum digunakan adalah integer, boolean, char, dan string.
- **Nilai data** yang disimpan dalam variabel dapat diperoleh dengan menyebutkan langsung variabelnya.
- **Informasi alamat** atau Lokasi dari sebuah variabel dapat diperoleh dengan menambahkan prefix **&** di depan nama variabel.
contoh : **&found** akan mendapatkan alamat memori untuk menyimpan data pada **found**.
- Jika variabel berisi alamat memori, prefix ***** pada variabel tersebut akan menampilkan nilai yang tersimpan dalam memori yang lokasinya disimpan dalam variabel tersebut.

Contoh : ***mem** akan mendapatkan data di memori yang alamatnya tersimpan di **mem**.

- Operasi yang terdapat di dalam tipe data :

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 apasaja	mendapatkan data dari lokasi memori dan mendapatkan lokasi dari variabel

Contoh :

Operasi	Hasil
"non suffi" + "cit mundo"	"non sufficit mundo"
2019.01 + 1.0102	2020.0202
2020 / 20	22.22
20.2 * 1.1	101
2020 % 1999	21
2020 & 1111	2104
2020 ^ 1111	1663
2020 >> 2	505
"minutus" < "magnus"	false
2020 >= 1234	true
! false && true	true

- Tipe data yang berbeda tidak boleh dicampur dalam satu ekspresi, bahkan tipe data yang masih sejenis. Misalnya masih sama-sama integer (**int** dan **int32**). Untuk menyesuaikan tipe data, ada beberapa cara yang dapat dilakukan:
 - Casting, **tipe(data)**, mengubah tipe dari data yang diberikan ke tipe data yang diinginkan.
 - Memanfaatkan fungsi **Sprint** dan **Scan** dari **fmt**
 - Memanfaatkan fungsi-fungsi dalam paket **strconv**, seperti **Atoi**, **Itoa**, dan **ParseBool**.

Contoh :

Operasi	Hasil
2020.0 % 19	will be an illegal expression error
int(2020.0) % 19	6

Konversi tipe	Data	Tipe baru	Keterangan
tipe(data)	integer	integer	format data tidak berubah, hanya penyesuaian jumlah bit. Kekurangan bit diisi bit 0 di sebelah kiri (MSB)
	real	real	format data tidak berubah, hanya penyesuaian jumlah bit. Kekurangan bit, maka bit mantisa diisi bit 0.
	real	integer	format data disesuaikan dengan tipe data tujuan
	integer	real	format data disesuaikan dengan tipe data tujuan
fmt.Sprintf("%v", v)	any type	string	tulis output ke string
fmt.Sprintf("%c", v)	karakter	string	tulis karakter ke string
fmt.Sscanf(s, "%v", &v)	string	any type	baca string ke variabel dengan tipe tertentu
fmt.Sscanf(s, "%c", &v)	string	karakter	baca string ke variabel bertipe karakter

- Variabel harus dideklarasikan terlebih dahulu sebelum digunakan. Variabel juga harus diinisialisasikan dulu (diisi data) supaya nilai yang disimpan diketahui dengan jelas dan dieksekusi algoritma menjadi terprediksi. Di bahasa Go, variabel yang tidak diinisialisasikan lebih dahulu otomatis akan diisi dengan nilai default yang ekuivalen dengan bit 0.
 - Nilai 0 untuk bilangan integer
 - 0.0E+0 untuk bilangan real
 - **false** untuk boolean
 - Karakter NUL untuk karakter
 - "" (string kosong, string dengan panjang 0) untuk string
 - **nil** untuk alamat memori

Notasi deklarasi variabel	Penulisan dalam Go	Keterangan
kamus a : tipe	var a tipe	a diinisialisasi dengan nilai default
kamus a : tipe algoritma a <- nilai_awal	var a tipe = nilai_awal var a = (tipe)nilai_awal	a diinisialisasi dengan nilai_awal
	a := nilai_awal a := (tipe)nilai_awal	secara implisit , tipe variabel a ditentukan dari nilai inisialisasinya

Contoh :

```

1 func main() {
2     var a int
3     a = 2019
4     r := 2019.0707
5     b := false
6     c := 'x'
7     s := "a string is a string"
8 }

```

2) Instruksi Dasar

Notasi instruksi dasar	Penulisan dalam bahasa Go	Keterangan
v1 <- e1 v1 <- v1 + e1 v1 <- v1 - e1 v1 <- v1 + 1 v1 <- v1 - 1	v1 = e1 v1 += e1 // atau v1 = v1 + e1 v1 -= e1 // atau v1 = v1 - e1 v1++ // atau v1 = v1 + 1 v1-- // atau v1 = v1 - 1	operasi assignment, mengisi data ke lokasi memori (variabel)
input(v1, v2)	fmt.Scan(&v1, &v2) fmt.Scanln(&v1, &v2) fmt.Sprintf("%v %v", &v1, &v2)	Pembacaan data memerlukan alamat memori ke mana data akan disimpan.
output(e1, e2)	fmt.Print(e1, e2) fmt.Println(e1, e2) fmt.Printf("%v %v\n", e1, e2)	Penulisan data memerlukan nilai data yang akan ditulis.

Contoh:

```

1 package main
2 import "fmt"
3
4 func main() {
5     var a, b, c float64
6     var hipotenusa bool
7
8     fmt.Scanln( &a, &b, &c )
9     hipotenusa = (c*c) == (a*a + b*b)
10    fmt.Println( "Sisi c adalah hipotenusa segitiga a,b,c: ", hipotenusa)
11 }

```

3) Konstanta Simbolik

Konstanta dapat diberi nama untuk memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut.

Seperti PI untuk merepresentasikan konstantan π .

```
1 const PI = 3.1415926535897932384626433
2 const MARKER = "AKHIR"
```

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

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

Jawab :

Source Code

```
package main

import "fmt"

func main(){
    var (
        satu, dua, tiga string
        temp string
    )

    fmt.Println("\nMasukkan input string: ")
    fmt.Scanln(&satu)
    fmt.Println("Masukkan input string: ")
    fmt.Scanln(&dua)
    fmt.Println("Masukkan input string: ")
    fmt.Scanln(&tiga)
    fmt.Println("\nOutput awal = " + satu + " " + dua + " " + tiga)
    temp = satu
    satu = dua
    dua = tiga
    tiga = temp

    fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga )
    fmt.Println("\n")
}
```

Screenshoot Program

```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO\tempCodeRunnerFile.go"
Masukkan input string : hari
Masukkan input string : ini
Masukkan input string: Kamis
Output awal = hari ini Kamis
Output akhir = ini Kamis hari
```

Deskripsi Program

Program ini merupakan implementasi dari bahasa pemrograman Go. Pada program diatas, pengguna diminta untuk menginputkan kata sebanyak 3 kali

yang akan disimpan pada string `satu`, `dua` dan `tiga`. Lalu setelah kata yang dimasukkan oleh pengguna, maka program akan melakukan swapping terhadap kata yang diinputkan. Pertama kata pada string `satu` disimpan pada variabel `temp`. Lalu kalimat `satu` ditukar nilainya ke string `dua` . Lalu mengganti nilai string `dua` dengan nilai `tiga`. Dan mengganti nilai string `tiga` dengan nilai yang disimpan di `temp` (nilai string `satu`). Kemudian program akan mencetak hasil dari kalimat inputan awal sebelum ditukar dan setelah di tukar.

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**).

Jawab :

Source Code

```
package main
import (
    "fmt"
)

func tahunKabisat(tahun int) bool{
    if tahun % 400 == 0{
        return true
    } else if tahun%100 == 0{
        return false
    } else if tahun%4 == 0{
        return true
    }
    return false
}

func main(){
    var year int
    for {

        fmt.Printf("\nMasukkan tahun (0 untuk keluar) : ")
        fmt.Scanln(&year)

        if (year) == 0 {
            fmt.Println("\nProgram Selesai!\n")
            break
        }

        if tahunKabisat(year){
```

```

        fmt.Println("Kabisat : true")
    } else {
        fmt.Println("Kabisat: false")
    }

    }
    fmt.Println("\n")
}

```

Screenshoot Program

```

PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO\guided2.go"

Masukkan tahun (0 untuk keluar) : 2016
Kabisat : true

Masukkan tahun (0 untuk keluar) : 2000
Kabisat : true

Masukkan tahun (0 untuk keluar) : 2018
Kabisat: false

Masukkan tahun (0 untuk keluar) : 0

Program Selesai!

```

Deskripsi Program

Program diatas merupakan program yang menerima input sebuah bilangan bulat dan akan memeriksa apakah bilangan tersebut merupakan tahun kabisat atau bukan. Jika tahun yang dimasukkan adalah kabisat, maka program akan menampilkan pesan (**true**) dan jika bukan tahun kabisat maka program akan menampilkan pesan (**false**).

3. Buat program bola yang menerima input jari-jari bola (bilangan bulat).
Tampilkan Volume dan Luas kulit Bola.

Jawab :

Souce Code

```

package main

import (
    "fmt"
    "math"
)

```

```

func main(){
    var r int
    fmt.Printf("\nMasukkan jari-jari bola : ")
    fmt.Scanln(&r)

    volume := (4.0 / 3.0) * math.Pi * math.Pow(float64(r), 3)
    luasPermukaan := 4 * math.Pi * math.Pow(float64(r), 2)

    // menampilkan hasil
    fmt.Printf("\nVolume bola : %.2f\n", volume)
    fmt.Printf("Luas permukaan bola: %.2f\n", luasPermukaan)
    fmt.Println()
}

```

Screenshoot Program

```

PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\

```

Masukkan jari-jari bola : 5

Volume bola : 523.60
Luas permukaan bola: 314.16

Deskripsi Program

Program diatas merupakan program untuk menghitung volume bola dan luas permukaan bola. Program akan meminta user untuk memasukkan jari-jari bola yang akan dihitung, kemudian program akan menghitung volume dan luas permukaan bola dengan menggunakan library math untuk memudahkan perhitungan.

III. UNGUIDED Soal Latihan Modul 2B

1. 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 urutan yaitu : ‘merah’, ‘kuning’, ‘hijau’, ‘ungu’, dan **false** untuk urutan warna lainnya.

Jawab :

Source code

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    urutanBenar := []string{"merah", "kuning", "hijau", "ungu"}
    percobaan := make([][]string, 5)
    berhasil := true

    // Input dan tampilkan 5 percobaan
    for i := 0; i < 5; i++ {
        percobaan[i] = make([]string, 4)
        fmt.Printf("Percobaan %d:", i+1)
        for j := 0; j < 4; j++ {
            //fmt.Printf("Masukkan warna gelas %d: ", j+1)
            fmt.Scan(&percobaan[i][j])
            percobaan[i][j] = strings.ToLower(percobaan[i][j])
        }
    }

    for i := 0; i < 5; i++ {
        for j := 0; j < 4; j++ {
            if percobaan[i][j] != urutanBenar[j] {
                berhasil = false
            }
        }
    }

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

    // Jika gagal, tampilkan percobaan kedua (gagal)
    if !berhasil {
        berhasil = true

        for i := 0; i < 5; i++ {
            berhasil = false
        }
    }
}
```

Screenshoot Program

```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO
Percobaan 1:merah kuning hijau ungu
Percobaan 2:merah kuning hijau ungu
Percobaan 3:merah kuning hijau ungu
Percobaan 4:merah kuning hijau ungu
BERHASIL: true

PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO
Percobaan 1:merah kuning hijau ungu
Percobaan 2:merah kuning hijau ungu
Percobaan 3:merah kuning hijau ungu
Percobaan 4:ungu kuning hijau merah
Percobaan 5:merah kuning hijau ungu
BERHASIL: false
```

Deskripsi Program

Program tersebut merupakan program untuk meminta inputan dari user untuk memasukkan warna zat cairan pada gelas 1 hingga gelas ke 4 secara berurutan adalah “merah”, “kuning”, “ hijau”, dan “ungu” dan percobaan dilakukan berulang selama 5 kali.

Program akan menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali. Kemudian program akan menampilkan **true** apabila urutan warna sesuai dengan urutan percobaan yang ditentukan sebelumnya, dan program akan menampilkan **false** untuk urutan warna yang tidak sesuai.

2. Buatlah sebuah program yang menerima inputan sebuah bilangan bulat positif dan tidak nol. Kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita. Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetikkan ‘SELESAI’. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita.

Jawab:

Source Code

```
package main

import (
    "fmt"
    "strings"
)

func main(){

    var bunga string
    var pita string
    var jumlah int

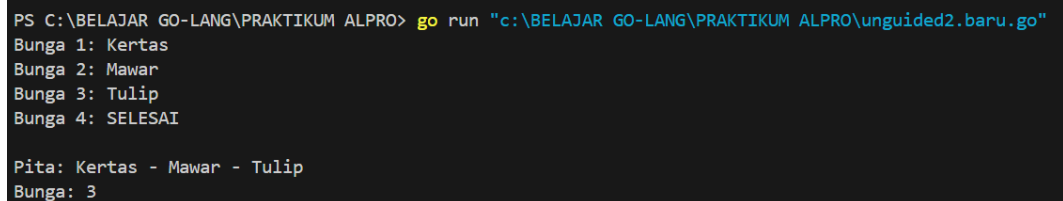
    for {
        fmt.Printf("Bunga %d: ", jumlah+1)
        fmt.Scan(&bunga)

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

        if jumlah > 0{
            pita += " - "
        }
        pita += bunga
        jumlah++
    }

    fmt.Printf("\nPita: %s\n", pita)
    fmt.Printf("Bunga: %d\n", jumlah)
}
```

Screenshoot Program



```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO\unguided2.baru.go"
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Bunga 4: SELESAI

Pita: Kertas - Mawar - Tulip
Bunga: 3
```

Deskripsi Program

Program ini merupakan sebuah program yang meminta pengguna menginputkan nama bunga secara berulang ulang dan nama bunga tersebut akan di dimpan dalam pita. Proses input nama Bungan akan berhenti apabila pengguna mengetikkan “SELESAI”. Kemudian program akan menampilkan isi pita beserta banyak bunga yang ada di dalam pita.

3. Buat program untuk 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. Modifikasi program agar 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 negative.

Jawab :

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var beratKiri, beratKanan float64

    for {

        fmt.Print("Masukkan berat belanjaan di kedua kantong: ")
        _, err := fmt.Scanln(&beratKiri, &beratKanan)

        if err != nil {
            fmt.Println("\nInput tidak valid. Mohon masukkan dua angka.")
            continue
        }

        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("\nProses selesai. Berat tidak bisa negatif.")
            break
        }

        totalBerat := beratKiri + beratKanan
        if totalBerat > 150 {
            fmt.Println("Proses selesai. Total berat melebihi batas.")
            break
        }
        selisihBerat := math.Abs(beratKiri - beratKanan)
        akanOleng := selisihBerat >= 9

        fmt.Printf("Sepeda motor pak Andi akan oleng: %t\n", akanOleng)

        if beratKiri >= 90 || beratKanan >= 90 {
            fmt.Println("\n\nProses selesai. Berat salah satu kantong melebihi batas 90.\n")
            break
        }
    }
}
```

Screenshoot Program

```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO\unguided3.go"

Masukkan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false

Masukkan berat belanjaan di kedua kantong: 55.6 70.2
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. Total berat melebihi batas.
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO>
```

Deskripsi Program

Program tersebut merupakan sebuah program yang meminta pengguna untuk memasukkan 2 inputan bilangan bulat yang merupakan berat pada kedua kantong. Lalu program akan menghitung berat sesuai bilangan yang diinputkan apakah akan membuat sepeda oleng atau tidak. Program akan menampilkan **true** jika selisih kedua isi kantong lebih dari atau sama dengan 9kg. Lalu program akan berhenti memproses apabila total berat isi kedua kantong melebihi 150 kg atau salah satu kantong beratnya negatif.

4. Buatlah program yang menerima input sebuah bilangan sebagai **K**, kemudian menghitung dan menampilkan nilai $f(K)$ sesuai persamaan diatas. Lalu modifikasi program sebelumnya yang menerima input integer **K** dan menghitung $\sqrt{2}$ dari bilangan **K**.

Jawab :

```
package main

import(
    "fmt"
    "math"
)

func hitungAkar2(K int) float64{
    hasil := 1.0
```



```

    for k := 0; k <= K; k++{
        hasil *= hitungF(float64(k))
    }
    return hasil
}

func hitungF(k float64) float64{
    pembilang := math.Pow(4*k+2, 2)
    penyebut := (4*k + 1) * (4*k + 3)
    return pembilang / penyebut
}

func main(){
    var K int

    fmt.Print("Masukkan nilai K : ")
    _, err := fmt.Scanf("%d", &K)

    if err != nil{
        fmt.Println("Input tidak valid. Harap masukkan bilangan bulat.")
        return
    }

    hasil := hitungAkar2(K)

    fmt.Printf("Nilai K = %d\n", K)
    fmt.Printf("Nilai akar 2 = %.10f\n", hasil)
}

```

Screenshoot Program

```

PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-L
Masukkan nilai K : 10
Nilai K = 10
Nilai akar 2 = 1.4062058441

```

Deskripsi Program

Program tersebut merupakan program yang digunakan untuk menerima input sebuah bilangan sebagai **K**, kemudian program akan menghitung nilai $f(k)$ dengan menggunakan fungsi `hitungF` lalu program juga akan menghitung akar 2 dari bilangan yang dimasukkan dengan menggunakan fungsi `hitungakar2`.

Soal Latihan Modul 2C

1. Buatlah program BiayaPos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut :

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

Source Code

```
package main

import (
    "fmt"
    //"os"
)

func hitungBiayaPos(beratGram int) (int, int, int, int){ // int 4 kali akan
menunjukkan apa yg direturn
    beratKg := beratGram / 1000
    sisaGram := beratGram % 1000

    biayaPerKg := beratKg * 10000

    var biayaSisa int
    if beratKg >= 10{
        biayaSisa = 0
    } else if sisaGram >= 500{
        biayaSisa = sisaGram * 5
    } else {
        biayaSisa = sisaGram * 15
    }

    totalBiaya := biayaPerKg + biayaSisa

    return beratKg, sisaGram, biayaPerKg, totalBiaya
}

func main(){
    for {
        var berat int
        fmt.Print("Masukkan berat parcel dalam gram (0 untuk keluar) : ")
        _, err := fmt.Scanln(&berat) // _ untuk mengabaikan nilai yang dikembalikan
oleh fungsi
        if err != nil{ // memeriksa apakah error atau tidak
            fmt.Println("Input tidak valid. Mohon masukkan angka.")
            continue
        }

        if berat == 0{
            fmt.Println("Terima kasih telah menggunakan program ini.")
        }
    }
}
```

```

        break
    }
    if berat < 0{
        fmt.Println("Berat tidak boleh negatif. Silahkan coba lagi")
        continue
    }

    beratKg, sisaGram, biayaPerKg, totalBiaya := hitungBiayaPos(berat)

    fmt.Println("\nHasil Perhitungan: ")
    fmt.Printf("Berat parcel\t: %d\n", berat)
    fmt.Printf("Detail berat\t: %d kg + %d gr\n", beratKg, sisaGram)
    fmt.Printf("Detail Biaya\t: Rp. %d + Rp. %d\n", biayaPerKg, totalBiaya-
biayaPerKg)
    fmt.Printf("Total biaya\t: Rp. %d\n", totalBiaya)
    }
}

```

Screenshoot Program

```

PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO\unguided5.go"
Masukkan berat parcel dalam gram (0 untuk keluar) :
8500

Hasil Perhitungan:
Berat parcel      : 8500
Detail berat      : 8 kg + 500 gr
Detail Biaya      : Rp. 80000 + Rp. 2500
Total biaya       : Rp. 82500

Masukkan berat parcel dalam gram (0 untuk keluar) : 9250

Berat parcel      : 9250
Detail berat      : 9 kg + 250 gr
Detail Biaya      : Rp. 90000 + Rp. 3750
Total biaya       : Rp. 93750

Masukkan berat parcel dalam gram (0 untuk keluar) : 0
Terima kasih telah menggunakan program ini.

```

Deskripsi Program

Program tersebut merupakan program untuk menghitung biaya pengiriman suatu toko berdasarkan berat parcel. Untuk menghitung biaya pengiriman harus melalui ketentuan sebagai berikut :

- Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam)
- Biaya jasa pengiriman adalah Rp. 10.000,- per kg
- Jika sisa berat \geq 500 gram, maka tambahan biaya adalah Rp. 5,- per gram

- Jika sisa berat < 500 gram, maka biaya tambahan yang dikenakan adalah Rp. 15,- per gram.
- Jika sisa berat (yang kurang dari 1 kg) digratiskan biayanya apabila total berat ternyata lebih dari 10 kg.

Program akan meminta pengguna menginputkan berat parcel, lalu program akan menampilkan hasil perhitungan pengiriman berdasarkan ketentuan yang ada.

2. Diberikan sebuah nilai akhir mata kuliah (NAM) [0..10] dan standar penilaian mata kuliah (NMK). Program berikut menerima input sebuah bilangan riil yang menyatakan NAM. Program menghitung NMk dan menampilkannya

Jawab :

Source Code Sebelum diperbaiki

```
package main

import "fmt"

func main(){
    var nam float64
    var nmk string
    fmt.Println("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)
}
```

Screenshoot Program

```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO\unguided6(1).go"
# command-line-arguments
.\unguided6(1).go:8:6: undefined: fmt.Println
.\unguided6(1).go:11:9: cannot use "A" (untyped string constant) as float64 value in assignment
.\unguided6(1).go:14:9: cannot use "AB" (untyped string constant) as float64 value in assignment
.\unguided6(1).go:17:9: cannot use "B" (untyped string constant) as float64 value in assignment
.\unguided6(1).go:20:9: cannot use "BC" (untyped string constant) as float64 value in assignment
.\unguided6(1).go:23:9: cannot use "C" (untyped string constant) as float64 value in assignment
.\unguided6(1).go:26:9: cannot use "D" (untyped string constant) as float64 value in assignment
.\unguided6(1).go:28:9: cannot use "E" (untyped string constant) as float64 value in assignment
```

- **Jawablah pertanyaan-pertanyaan berikut :**

- b. Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut?
apakah eksekusi program tersebut sesuai ekspektasi soal?

Jawab :

jika program sebelum diperbaiki maka program tidak dapat dijalankan karena terdapat banyak kesalahan yang ada di dalam program.

- b.** Apa saja kesalahan dari program tersebut? Mengapa demikian
jelaskan alur program seharusnya!

Jawab :

Kesalahan dalam program

- Variabel **nam** dideklarasikan dengan **float64**. Namun dalam pemberian value pada program tersebut memberikan nilai string. Hal ini menyebabkan kesalahan karena tipe data **float64** dan string tidak kompatibel
- Struktur penggunaan percabangan **if-else-if** tidak benar karena pada program tersebut hanya menggunakan if saja. Setelah pengecekan kondisi pertama menggunakan **if** maka pengecekan setelahnya menggunakan **if else** dan untuk kondisi terakhir menggunakan **else**, untuk memastikan bahwa satu kondisi yang dievaluasi dan yang pertama memenuhi syarat langsung digunakan.

- Variabel **nmk** sudah dideklarasikan tetapi tidak pernah dipanggil/diisi nilainya. Seharusnya nilai **nam** yang sudah diubah menjadi huruf dipanggil sebagai nilai dari **nmk**.

Alur Program yang seharusnya :

- Program meminta input dari user berupa nilai angka
- Nilai yang diinputkan tersebut akan dikonversikan menjadi huruf (A,B,B,D atau E) berdasarkan pengelompokan yang ditentukan
- Hasil nilai dari konversi nilai menjadi huruf akan ditampilkan sebagai output.

c. Perbaiki program tersebut! Ujilah dengan masukan : 93.5, 70,6; dan 49.5. Seharusnya keluaran yang diperoleh 'A', 'B', dan 'D'

o Source code (sesudah diperbaiki)

```
package main

import "fmt"

func main(){
    var nam float64
    var nmk string

    // Meminta input nilai akhir mata kuliah
    for {
        fmt.Print("Nilai akhir mata kuliah: ")
        _, err := fmt.Scanln(&nam)
        if err != nil{
            fmt.Println("Input tidak valid. Mohon masukkan angka.")
            var discard string
            fmt.Scanln(&discard)
            continue
        }

        if nam == 0{
            fmt.Println("Program Selesai")
            break
        }

        // Menentukan nilai huruf berdasarkan nilai numerik
        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"
    }

    // Menampilkan hasil nilai huruf
    fmt.Println("Nilai mata kuliah: ", nmk)
    fmt.Println()

}
}

```

Screenshoot Program

```

PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKT
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A

Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B

Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D

```

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

Jawab :

Source code

```

package main

import (
    "fmt"
    "strings"
)

func cariFaktor(bilangan int) []int{
    var faktor [] int
    for i := 1; i <= bilangan; i++){

```

```

        if bilangan%i == 0{
            faktor = append(faktor, i)
        }
    }
    return faktor
}

func isPrima(bilangan int) bool{
    if bilangan <= 1{
        return false
    }
    for i := 2; i*i <= bilangan; i++){
        if bilangan%i == 0{
            return false
        }
    }
    return true
}

func main(){
    var bilangan int
    fmt.Print("Masukkan bilangan bulat (b > 1) : ")
    _, err := fmt.Scanf("%d", &bilangan)

    if err != nil || bilangan <= 1{
        fmt.Println("Input tidak valid. Masukkan bilangan bulat lebih
besar dari 1.")
        return
    }

    faktor := cariFaktor(bilangan)
    prima := isPrima(bilangan)

    fmt.Printf("\nBilangan: %d\n", bilangan)
    fmt.Printf("Faktor: %s\n",
strings.Trim(strings.Join(strings.Fields(fmt.Sprint(faktor)), " "),
"[]"))
    fmt.Printf("Prima : %t\n", prima)
    fmt.Println("\n\n")
}

```

Screenshoot Program

```

PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO> go run "c:\BELAJAR GO-LANG\PRAKTIKUM ALPRO\unguided7.go"
Masukkan bilangan bulat (b > 1) : 12

Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima : false

```


Deskripsi Program

Program tersebut merupakan program yang meminta user untuk menginputkan sebuah bilangan bulat b dan b harus > 1 . Setelah itu, program akan mencari dan menampilkan faktor dari bilangan yang diinputkan. Lalu program juga akan menentukan apakah bilangan **b** yang diinputkan bilangan prima atau bukan. Jika bilangan tersebut prima maka program akan mencetak pesan **true** dan jika bukan bilangan prima maka program akan mencetak pesan **false**.