

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 2
REVIEW STRUKTUR KONTROL**



Oleh:

WILDAN DAFFA' HAKIM PUTRA ANTARA

2311102055

S1IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Dalam program utama di golang, selalu terdapat dua komponen

- **package main** merupakan bahwa ini merupakan program utama
- **func main** merupakan fungsi yang akan dieksekusi Ketika program berjalan.

Dalam memberikan komentar ke program golang memiliki beberapa cara antara lain :

- “//” untuk memberikan komentar pada baris tersebut

```
// ini cara komentar
```

- “/*” ini untuk memberikan komentar multi line. Komentar ini bisa dituliskan dalam beberapa baris. Untuk mengakhiri komentar menggunakan “*/”

```
/* ini merupakan contoh komentar  
Beberapa line */
```

Program dalam Bahasa golang disimpan dalam ekstensi .go yang nantinya akan dikompilasi saat dijalankan / di build. Terdapat beberapa utilitas dalam golang

- go build : untuk melakukan kompilasi pada program golang yang berada pada folder menjadi program.
- go build <namafile>.go : untuk mengkompilasi file golang.
- go fmt : melakukan formatting file golang pada folder golang agar sesuai standar.
- go clean : membersihkan file pada folder golang sehingga sisa source code

1. Tipe data dan instruksi dasar

Dalam golang terdapat beberapa tipe data yang dapat digunakan antara lain :

Notasi Tipe data	Dalam Go
Interger	int int8 int32 //rune int64 uint

	uint8 //byte uint32 uint64
Boolean	bool
String	string
Real	float32 float64
Character	Byte rune

Untuk mengakses Alamat memori pada suatu variable bisa menambahkan & didepan nama variable misalkan : &luas dan Jika variabel berisi alamat memori, prefiks * pada variabel tersebut akan memberikan nilai yang tersimpan dalam memori yang lokasinya disimpan dalam variabel tersebut.

2. Operator

Untuk golang sendiri operator aritmatika dan logika tidak banyak berbeda seperti pada c++ seperti (+) untuk pertambahan, (-) untuk pengurangan, (*) untuk perkalian, (/) untuk pembagian. Dan untuk operator logika (&&) untuk AND, (||) untuk OR, (!) untuk not

Golang sendiri cukup ketat dalam tipe data sehingga perlu melakukan casting jika ingin melakukan operasi pada nilai yang memiliki tipe data berbeda. Untuk melakukan casting pada golang cukup dengan “tipedata(data)”. Selain casting juga dapat memanfaatkan Sprint dan Sscan dari paket fmt. Selain itu, untuk string dapat menggunakan library strconv.

3. Konstanta

Konstanta digunakan untuk mendeklarasikan variable dengan nilai tepat dan tidak berubah – ubah. Pada golang sendiri cara membuat konstanta sebagai berikut.

<code>const pi = 3.14</code>

4. Perulangan

Dalam golang hanya mempunyai notasi for untuk perulangan dengan beberapa jenis penggunaan seperti :

<code>for inisiasi; kondisi; update { //Do any }</code>	Perulangan biasa pada loop for
---	--------------------------------

for kondisi { //Do any }	Perulangan bergantung pada kondisi seperti while loop
for { //Do any }	Perulangan terus menerus hanya bisa berhenti dengan kondisi if + break
for n, var := array { //Do any }	iterator mengunjungi seluruh isi slice/array. Seperti foreach

5. Percabangan

Pada golang, notasi untuk percabangan if else tidak jauh berbeda. Seperti berikut :

```
If kondisi {
    // do if the condition met
} else if kondisi {
    // do if the condition met
} else {
    // do if none of the condition met
}
```

Sedangkan pada switch case sebagai berikut :

```
switch ekspresi {
    case ex1:
        //do if met
    case ex2:
        //do if met
    default:
        // do if none of the case match
}
```

Kita juga dapat membuat switch statement tanpa ekspresi seperti berikut :

```
switch {
    case kondisi_1:
        //do if met
    case kondisi_2:
```

```
//do if met
default:
    // do if none of the case match
}
```

II. GUIDED

A. 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
    var 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)
}
```

```
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Doc
Masukkan input string: a
Masukkan input string: b
Masukkan input string: c
Output awal = a b c
Output akhir = b c a
```

Program ini melakukan penggeseran nilai variable . pada awal program program menerima tiga input melalui fmt.Scan dan disimpan pada variable satu, dua, dan tiga. Lalu nilai variable awal dicetak. Setelah itu, variable dibalik dengan cara variable temp diisi dengan nilai a, variable a diisi dengan b, variable b diisi dengan c, dan variable c diisi dengan nilai temp sehingga Ketika output akhir dicetak nilai akan bergeser menjadi b dulu lalu c baru a.

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.Printf("Kabisat : %t \n", (tahun%4 ==
0 && (tahun%100 != 0 || tahun%400 == 0)))
}
```

```
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/Kuliah/SMT 3/AL
Tahun : 2000
Kabisat : true
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/Kuliah/SMT 3/AL
Tahun : 2016
Kabisat : true
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/Kuliah/SMT 3/AL
Tahun : 2018
Kabisat : false
```

Program ini merupakan program untuk mengecek tahun kabisat. Pada awal program akan diminta inputan user yang akan disimpan pada variable tahun setelah itu akan dicetak menggunakan printf dengan %t menandakan bahwa tipenya adalah Boolean pengecekan dilakukan dengan memodulus tahun dengan 4 lalu memberikan operator AND yang didalam operasi akan dicek apakah tahun tidak habis dibagi seratus atau habis dibagi 400 ini dilakukan karena **“Tidak semua tahun yang bisa dibagi dengan 100 adalah tahun kabisat, kecuali tahun tersebut bisa dibagi dengan 400”**. Misalkan kita input 2000 maka akan berniali false jika tanpa dicek dengan modulus 400.

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

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jari, volume, luasKulit float64
    fmt.Print("Jejari = ")
    fmt.Scan(&jari)
    volume = math.Pi * (4.0 / 3.0) *
    math.Pow(jari, 3)
    luasKulit = 4 * math.Pi * math.Pow(jari, 2)
    fmt.Print("Bola dengan jejari ", jari, "
    memiliki volume ", volume, " dan luas kulit
    ", luasKulit)
}
```

```
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/Kuliah
Jejari = 5
Bola dengan jejari 5 memiliki volume 523.5988 dan luas kulit 314.1593
λ MacBook-Air-Daffa Modul 2 →
```

Program ini digunakan untuk menghitung volume dan luas kulit dari sebuah bola. Pada awal, program ini akan menerima inputan jari jari yang disimpan pada variable jari. Lalu akan dilakukan operasi matematika sesuai rumus volume bola dan disimpan pada variable volume. Setelah itu, dilakukan operasi untuk luaskulit bola dan disimpan pada variable luasKulit. Lalu hasilnya dicetak dengan printf, "%0.4f" pada printf digunakan untuk membatasi 4 digit setelah koma.

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

A. Modul 2A

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

```
package main
```

```

import "fmt"

func main() {
    var fahrenheit, celcius float64
    fmt.Print("Temperatur Celcius : ")
    fmt.Scan(&celcius)

    fahrenheit = 32 + celcius*9.0/5.0
    fmt.Println("Derajat Fahrenheit ",
        fahrenheit)
}

```

```

λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/
go"
Temperatur Celcius : 50
Derajat Fahrenheit 122

```

Program diatas melakukan konversi terhadap suhu celcius menjadi Fahrenheit. Pada awal program pengguna diminta menginputkan tempratur celcius dan disimpan pada variable celcius. Setelah itu dilakukan operasi matematika sesuai rumus koversi dan disimpan pada variable Fahrenheit lalu ditampilkan bersamaan dengan text “Derajat Fahrenheit”. (Note : terdapat kesalahan rumus konversi celcius ke Fahrenheit, seharusnya : $32 + \text{celcius} \times 9/5$).

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

```

package main

import "fmt"

func main() {
    var fahrenheit, celcius, reamur, kelvin
    float64
    fmt.Print("Temperatur Celcius : ")
    fmt.Scan(&celcius)

    fahrenheit = 32 + celcius*9.0/5.0
    reamur = celcius * 4.0 / 5.0
    kelvin = (fahrenheit + 459.67) * 5.0 /
    9.0
    fmt.Println("Derajat Reamur", reamur)
    fmt.Println("Derajat Fahrenheit ",
        fahrenheit)
}

```



```
fmt.Println("Derajat Kelvin ",
int(kelvin))
```

```
}
```

```
Derajat Fahrenheit 122
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Docume
mod.go"
Temperatur Celcius : 50
Derajat Reamur 40
Derajat Fahrenheit 122
Derajat Kelvin 323
λ MacBook-Air-Daffa Modul 2 →
```

Prosesnya sama seperti pada program sebelum dimodifikasi, hanya saja ditambahkan beberapa operasi matematika dengan rumus konversi suhu yang disimpan ke variabel reamur dan kelvin dengan tipe data float64

5. Buat program ASCII yang akan membaca 5 buah data integer dan mencetaknya dalam format karakter. Kemudian membaca 3 buah data karakter dan mencetak 3 buah karakter setelah karakter tersebut (menurut tabel ASCII).

```
package main

import "fmt"

func main() {
    var char1, char2, char3, char4, char5,
        char6, char7, char8 byte

    fmt.Scanf("%d%d%d%d%d", &char1, &char2,
        &char3, &char4, &char5)
    fmt.Scanf("%c%c%c", &char6, &char7,
        &char8)
    fmt.Printf("%c%c%c%c%c \n", char1, char2,
        char3, char4, char5)
    fmt.Printf("%c%c%c \n", (char6 + 1), (char7
        + 1), (char8 + 1))
}
```

```
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/
go"
66 97 103 117 115
SNO
Bagus
TOP
```

Program ini merubah code ASCII menjadi karakter dan melakukan manipulasi karakter dengan menambahkan dengan 1. Pada awal program user diminta menginput ascii code sebanyak 5 berupa interger dan pada input kedua diminta menginputkan character. Program ini memanfaatkan scanf yang untuk mengambil inputan digit kita dan disimpan dalam byte (ini bisa dilakukan karena byte alias dari uint8). Hal yang sama terjadi pada character hanya saja inputan yang diminta adalah character ditandai dengan “%c”. lalu inputan interger diawal dicetak dengan printf disertai format verbs “%c” agar yang muncul adalah character. Ini dilakukan juga untuk mencetak inputan kedua hanya saja setiap char yang disimpan dalam bentuk byte ditambah denan 1 terlebih dahulu.

B. 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 informasi yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna lainnya.

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var ujiKimia = [5][4]string{}
    var warna = [4]string{"merah", "kuning",
    "hijau", "ungu"}
    var beda bool = true
    for i := 0; i < 5; i++ {
        fmt.Print("Percobaan ", i+1, ": ")
        fmt.Scan(&ujiKimia[i][0],
        &ujiKimia[i][1],          &ujiKimia[i][2],
        &ujiKimia[i][3])
    }

    for i := 0; i < 5; i++ {
        for j := 0; j < 4; j++ {
            beda
            strings.ToLower(ujiKimia[i][j]) == warna[j]
            if !beda {
                i = 5
            }
        }
    }
}
```

```

                                break
                            }
                        }
                    }
                fmt.Println("Berhasil : ", beda)
            }
        }
    }
}

```

```

λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/Ku
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
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/Ku
go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hij ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
Berhasil : false
λ MacBook-Air-Daffa Modul 2 → 

```

Program ini digunakan untuk memeriksa apakah hasil percobaan suda sesuai. Pada awal program, pengguna diminta menginputkan hasil percobaan pada ke 4 gelas sebanyak 5 kali. Ini disimpan pada array 2 dimensi dengan nama variable ujiKimia. Setelah selesai menginputkan akan digunakan nested loop untuk mengecek setiap inputan. Inputan percobaan yang seharusnya telah dipetakan berdasarkan index pada variable warna. Nested loop paling luar mengakses percobaan ke-n dan yang dalamnya mengakses isi dari gelas. Jika terdapat satu saja yang tidak sesuai nilai beda akan berubah menjadi false yang akan memicu if pada nested loop dalam untuk menghentikan loop bagian dalam dan merubah nilai i pada loop terluar menjadi 5 (untuk menghentikan loop karena $i < 5$, alternatif bisa menggunakan break). Sehingga program akan mencetak Berhasil: false jika berbeda, dan Berhasil: true jika sama semua dari percobaan 1 – 5. Break disini dilakukan agar proses enjadi lebh cepat mengingat jika satu saja berbeda maka percobaan langsung gagal, maka jika ada 1 saja yang berbeda untuk menentukan percobaan berhasil atau tidak kita tidak perlu mengecek bagian setelahnya.

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.

```

package main

import "fmt"

func main() {
    var pita string
    var bunga string
    var banyak int
    fmt.Print("N : ")
    fmt.Scan(&banyak)
    for i := 0; i < banyak; i++ {
        fmt.Print("Bunga ", i+1, " : ")
        fmt.Scan(&bunga)
        pita += bunga + " - "
    }
    fmt.Println("Pita : ", pita)
}

```

```

Dennis : Tutse
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahak
go"
N : 3
Bunga 1 : Kertas
Bunga 2 : Mawar
Bunga 3 : Tulip
Pita : Kertas - Mawar - Tulip -

```

Program ini akan membuat pita dengan bunga yang kita inputkan ditambah dengan "-". Pada awal akan diminta inputan banyaknya Bunga yang akan dimasukkan dengan `fmt.Scan`. Lalu disimpan pada variable `banyak`. Setelah itu dilakukan perulangan sebanyak nilai variable `banyak`. Selama perulangan akan diminta menginputkan nilai bunga yang disimpan pada variable `bunga` sekaligus melakukan penggabungan string yang disimpan pada variable `pita`. Lalu setelah perulangan selesai, nilai `pita` akan dicetak.

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 pita string
    var bunga string
    var count int
    for {
        fmt.Print("Bunga ", count+1, " : ")
        fmt.Scan(&bunga)

        if bunga == "SELESAI" {
            break
        }
        pita += bunga + " - "
        count++
    }
    fmt.Println("Pita : ", pita)
    fmt.Println("Bunga : ", count)
}

```

```

λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Document
mod.go"
Bunga 1 : Kertas
Bunga 2 : Mawar
Bunga 3 : Tulip
Bunga 4 : SELESAI
Pita : Kertas - Mawar - Tulip -
Bunga : 3
λ MacBook-Air-Daffa Modul 2 →

```

Modifikasi pada program ini yaitu merubah perulangan menjadi while loop Dimana perulangan akan berhenti jika kita menginputkan SELESAI. Selama belum diketik SELESAI maka akan dilakukan penambahan count dengan bilangan 1 dan penggabungan bunga.

3. 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.

```

package main

import (
    "fmt"
)

func main() {

```

```

var berat1, berat2 float64

for {
    fmt.Print("Masukkan berat belanjaan
di kedua kantong : ")
    fmt.Scan(&berat1, &berat2)
    if berat1 >= 9 || berat2 >= 9 {
        fmt.Println("Proses Selesai")
        break
    }
}
}

```

```

λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/Kuliah/SMT
go"
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
Proses Selesai

```

Program ini digunakan untuk mengetahui apakah selisihnya dua kantong kurang dari Sembilan dan apakah salah satu kantong lebih dari sama dengan Sembilan. Pada awal program akan menerima inputan berat1 dan berat2. Jika berat salah satu kantong lebih dari 9 maka perulangan akan berhenti. Disini kita tidak perlu mengecek apakah selisih lebih dari 9, karena beban dibatasi hanya 9 kg setiap kantong sehingga tidak mungkin memiliki selisih lebih dari 9.

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 berat1, berat2 float64

    for {

```

```

        fmt.Print("Masukkan berat belanjaan
di kedua kantong : ")
        fmt.Scan(&berat1, &berat2)
        if !((berat1+berat2 > 150) || berat1
< 0 || berat2 < 0) {
            fmt.Println("Sepeda motor pak
andi akan oleng : ", math.Abs(berat1-
berat2) >= 9)
        } else {
            fmt.Println("Proses Selesai")
            break
        }
    }
}

```

```

λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Docume
mod.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

```

Pada modifikasi ini dilakukan perubahan pada statement Dimana jika kedua beratnya kurang dari 150 atau berat kantong1 dan kantong2 lebih dari 0 maka akan akan menampilkan motor oleng atau tidak, untuk mengeceknya dengan menghitung selisih kedua berat apakah lebih dari sama dengan 9 memanfaatkan library math untuk mendapatkan nilai absolut untuk mengatasi apabila selisih menghasilkan minus. Program akan berhenti jika total berat lebih dari 150 dan berat salah satu kantong negatif.

4. Buatlah sebuah program yang menerima input sebuah bilangan sebagai K, kemudian menghitung dan menampilkan nilai $f(K)$ sesuai persamaan di modul.

```

package main

import (
    "fmt"
    "math"
)

```

```
func main() {
    var k int
    var hasil float64 = 1.0
    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 Akar 2 = %.10f \n",
        hasil)
}
```

```
Proses Setoran
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffa/
go"
Nilai K = 100
Nilai Akar 2 = 1.0000061880
```

Program ini digunakan untuk menghitung fungsi matematika seperti yang ditentukan di modul. Pada awal program pengguna diminta melakukan inputan yang akan dimasukkan ke variable k. setelah itu, nilai k dimasukkan ke persamaan sesuai yang ada di modul lalu nilainya diinsiasi ke variable hasil. Setelah itu dicetak dengan printf dengan format verbs “%.10f” untuk mengambil 10 digit angka setelah koma.

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
    var hasil float64 = 1.0
    fmt.Print("Nilai K = ")
    fmt.Scan(&k)
    for i := 0; i <= k; i++ {
        hasil *= math.Pow(4*float64(i)+2, 2)
        / ((4*float64(i) + 1) * (4*float64(i) + 3))
    }
    fmt.Printf("Nilai Akar 2 = %.10f \n",
        hasil)
```



```
}
```

```
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/D
rFile.go"
Nilai K = 10
Nilai Akar 2 = 1.4062058441
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/D
rFile.go"
Nilai K = 100
Nilai Akar 2 = 1.4133387072
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/D
rFile.go"
Nilai K = 1000
Nilai Akar 2 = 1.4141252651
λ MacBook-Air-Daffa Modul 2 →
```

Pada modifikasinya, kita diminta untuk membuat program menghitung nilai akar 2 dengan harapan sehingga kita membutuhkan perulangan sebanyak $k+1$ kali dan setiap perulangan fungsi matematika diatas nilai x nya akan diisi dengan nilai i pada perulangan lalu dikalikan dengan nilai variable hasil sebelumnya. Setelah perulangan selesai, nilai hasil dicetak dengan printf dengan format verbs “%.10f” untuk mengambil 10 digit angka setelah koma.

C. MODUL 2C

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

```
package main

import "fmt"

func main() {
    var beratParsel, parselKG, parselGram,
        biayaKg, biayaGram int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&beratParsel)
    parselKG = beratParsel / 1000
    parselGram = beratParsel % 1000
    biayaKg = parselKG * 10000
    if parselGram >= 15 {
        biayaGram = parselGram * 5
    } else {
        biayaGram = parselGram * 15
    }
}
```

```

    fmt.Printf("Detail berat: %d kg + %d gr\n",
    parselKG, parselGram)
    fmt.Printf("Detail biaya: Rp. %d + Rp.
    %d\n", biayaKg, biayaGram)
    fmt.Printf("Total biaya: Rp. %d\n",
    biayaKg+BiayaGram)
}

```

```

λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim,
go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
λ MacBook-Air-Daffa Modul 2 → █

```

Program ini digunakan untuk menghitung biaya pengiriman. Pada awal program, user diminta inputan berat parsel dalam bentuk gram yang akan disimpan ke dalam variable beratParsel. Disini akan dipisahkan berat kilo dan gram, untuk berat kilo didapat dari pembagian nilai beratParsel dengan 1000, dan untuk gram dari beratParsel di modulus dengan 1000 ini akan diinisiasi pada variable parselKG dan parselGram. Lalu untuk mendapat nilai biayaKG kita mengalikan parselKG dengan 10000, untuk penghitungan biaya gramnya menggunakan if statement jika lebih dari sama dengan 15 maka untuk nilai variable biayaGram (yang menampung biaya gram) yaitu parselGram dikali dengan 5 dan jika tidak maka dikali 15. pada detail berat yang ditampilkan adalah nilai variable parselKG dan parselGram, Pada detail biaya yang ditampilkan adalah biayaKG dan biayaGram, pada total biaya yang ditampilkan merupakan hasil penjumlahan dari nilai variable biayaKG dan biayaGram.

2. 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?

Jawab : error, alasannya karena variable nam bertipe data float64 bukan string.

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

Jawab : seharusnya hasil nilainya disimpan pada variable nmk, if statement digabungkan menjadi if else statement, dan else terakhir untuk nilai E bisa digantikan dengan else mengingat tidak ada kondisi lain setelah e. Alurnya dengan melakukan pengecekan nam dari atas atau nilai lebih dari 80, lalu untuk statement menggunakan batas bawah dari range nilai. Sehingga batas atas dari setiap nilai ikut ke nmk bawahnya. Misalkan pada statement $nam > 72.5$ untuk ab dan $nam > 65$ untuk b maka nilai b dapat dicapai jika nilainya di range $65 < nilai \leq 72.5$.

- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Jawab :

```
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)
}
```

```
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/D
go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/D
go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/D
go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
```

Untuk cara kerjanya, sama seperti yang di deskripsikan pada poin b jadi program akan menerima inputan nilai dan disimpan pada variable nam. Alurnya dengan melakukan pengecekan nam dari atas atau nilai lebih dari 80, lalu untuk statement

menggunakan batas bawah dari range nilai. Sehingga batas atas dari setiap nilai ikut ke nmk bawahnya. Misalkan pada statement $nam > 72.5$ untuk ab dan $nam > 65$ untuk b maka nilai b dapat dicapai jika nilainya di range $65 < nilai \leq 72.5$. lalu nilai variable nmk dicetak bersama dengan string "Nilai mata kuliah : "

3. 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 number int

    fmt.Print("Bilangan: ")
    fmt.Scan(&number)
    if number <= 1 {
        fmt.Println("Bilangan harus lebih dari 1!")
        return
    }
    fmt.Print("Faktor :")
    for i := 1; i <= number; i++ {
        if number%i == 0 {
            fmt.Print(" ", i)
        }
    }

}
```

```
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/go"
go"
Bilangan: 12
Faktor : 1 2 3 4 6 12
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/go"
go"
Bilangan: 7
Faktor : 1 7
λ MacBook-Air-Daffa Modul 2 →
```

Program ini digunakan untuk mencari nilai bilangan prima, pada awal program pengguna diminta input angka yang akan disimpan pada variable number. Cara mencari faktornya dengan melakukan perulangan sebanyak bilangan tersebut dimulai dari angka 1. Jika

bilangan yang diinput di modulo dengan nilai i pada perulangan habis dibagi maka bilangan tersebut faktor dan akan dicetak pada terminal. Sebelum mencari faktor akan dilakukan pengecekan apakah bilangan yang diinputkan kurang dari atau sama dengan 1 jika iya maka akan dicetak bilangan harus lebih dari 1 alu program dihentikan dengan return.

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 number int
    var countFaktor int

    fmt.Print("Bilangan: ")
    fmt.Scan(&number)
    if number <= 1 {
        fmt.Println("Bilangan harus lebih dari 1!")
        return
    }
    fmt.Print("Faktor :")
    for i := 1; i <= number; i++ {
        if number%i == 0 {
            fmt.Print(" ", i)
            countFaktor++
        }
    }

    fmt.Print("\nPrima : ", countFaktor == 2,
        " \n")
}
```

```
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/mod.go"
Bilangan: 12
Faktor : 1 2 3 4 6 12
Prima : false
λ MacBook-Air-Daffa Modul 2 → go run "/Users/daffahakim/Documents/mod.go"
Bilangan: 7
Faktor : 1 7
Prima : true
```

Pada lanjutan program hanya dilakukan sedikit penambahan yaitu penambahan variable `countFaktor` lalu melakukan increment `countFaktor` setiap faktor dari bilangan ditemukan. Dan terakhir penambahan print dengan operator logika yang akan mengecek apakah banyaknya faktor hanya dua, jika iya maka bilangan tersebut prima (bilangan prima sendiri hanya memiliki faktor bilangan 1 dan bilangan itu sendiri, bilangan selain prima otomatis punya lebih dari 2 faktor karena bilangan 1 dan bilangan itu sendiri ikut terhitung).