

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 2  
REVIEW STRUKTUR KONTROL**



Oleh:

NAMA : Alberta Aurora Adiswari

NIM : 2311102220

KELAS : S1-IF-11-02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## I. DASAR TEORI

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.

Go diimplementasikan sebagai kompilator. Berikut adalah contoh sesi yang biasa dilakukan saat mengkompilasi dan mengeksekusi program dalam Bahasa Go:

- Panggil shell atau terminal (program/utility cmd.exe di Windows)
- Masuk ke dalam (cd) folder program (normalnya ada di C:\Users\go\src\ atau yang sejenis)
- Kemudian panggil perintah go build atau go build file.go untuk mengkompilasi file.go
- Jika gagal, akan muncul pesan eror yang sesuai, pelajari dengan baik pesan tersebut, perbaiki teks program sumber, kemudian ulangi proses buildnya.
- Jika berhasil maka pada folder tersebut akan dibuat program dengan nama yang sama dan diakhiri dengan .exe (untuk Windows)
- Panggil program eksekutabel tersebut dari terminal yang sama. Jangan memanggil program tersebut dengan mengklik eksekutabel tersebut dari folder karena program kalian hanya berbasis teks, bukan/belum dirancang dengan tampilan Windows.

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:** membaca 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 sumber nya saja.

Variabel adalah nama dari suatu Lokasi di memori, yang data dengan tipe tertentu dapat disimpan.

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 <sup>9</sup> ..10 <sup>9</sup> 64 bit: -10 <sup>19</sup> ..10 <sup>19</sup> bergantung platform 0..255 0..4294967295 0..(2 <sup>64</sup> -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 rune	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 <b>per-bit</b> 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 <b>boolean</b> AND, OR, dan NOT
* &	variabel apasaja	mendapatkan data dari lokasi memori dan mendapatkan lokasi dari variabel

Bahasa Go menggunakan kesesuaian tipe data yang ketat. Tipe data yang berbeda tidak dapat dicampur dalam satu ekspresi, termasuk tipe yang sejenis semial sama – sama integer. Untuk menyesuaikan tipe data, ada beberapa cara yang dapat dilakukan:

- Casting, **tipe(data)**, mengubah tipe dari data yang diberikan menjadi tipe data yang diinginkan.
- Memanfaatkan fungsi **Sprint** dan **Sscan** pada paket **fmt**.

- o Memanfaatkan fungsi pada paket **strconv**, seperti **Atol**, **Itoa** dan **ParseBool**.

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

Sebelum digunakan variable harus dideklarasikan dan diinisialisasi terlebih dahulu agar nilai yang tersimpan diketahui dengan jelas dan eksekusi algoritma menjadi terprediksi. Dalam Bahasa Go, variable yang tidak diinisialisasi lebih dahulu otomatis diisi dengan nilai default yang ekuivalen dengan bit 0.

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 <b>implisit</b> , tipe variabel a ditentukan dari nilai inisialisasinya

## Instruksi Dasar

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

Go hanya mempunyai kata kunci `for` untuk semua jenis perulangan yang kita pelajari dalam notasi algoritma. Dua bentuk yang kita gunakan di sini adalah struktur `while-loop` dan `repeat-until`.

Bentuk perulangan dalam Bahasa Go

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

Bentuk `while-loop` memastikan setiap kali memasuki loop, ada kondisi yang harus terpenuhi, yang berarti saat keluar dari loop nilai kondisi tersebut pasti salah.

	Notasi algoritma	Penulisan dalam bahasa Go
1	<code>while (kondisi) do</code>	<code>for kondisi {</code>
2		
3	<code>... kode yang diulang</code>	<code>.. kode yang diulang</code>
4	<code>endwhile</code>	<code>}</code>

Bentuk `repeat-until` di perulangan dilakukan terus menerus sampai kondisi keluar terpenuhi. Artinya selama kondisi belum terpenuhi maka perulangan akan terus dilakukan. Pada saat keluar dari loop maka nilai kondisi pasti benar.

	Notasi Algoritma	Penulisan dalam bahasa Go
1	repeat	for selesai:=false; !selesai; {
2	.. kode yang diulang	.. kode yang diulang
3	until (kondisi)	selesai = kondisi
4		}
5		
6		for selesai:=false; !selesai;
7		selesai=kondisi {
8		.. kode yang diulang
9		}

#### Bentuk if-else

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

#### Bentuk Switch-case

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

## II. GUIDED

### 1. Guided 1

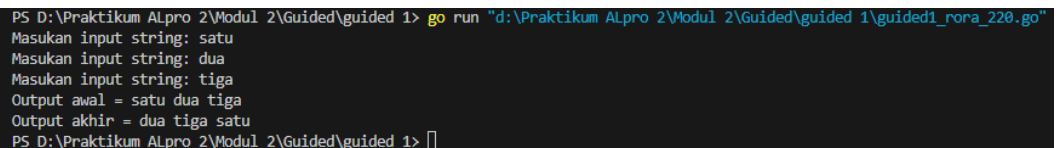
#### Source code

```
package main

import "fmt"

func main() {
    var (
        satu, dua, tiga string
        temp                string
    )
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("Masukan 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)
}
```

#### Screenshoot program



```
PS D:\Praktikum ALpro 2\Modul 2\Guided\guided 1> go run "d:\Praktikum ALpro 2\Modul 2\Guided\guided 1\guided1_rora_228.go"
Masukan input string: satu
Masukan input string: dua
Masukan input string: tiga
Output awal = satu dua tiga
Output akhir = dua tiga satu
PS D:\Praktikum ALpro 2\Modul 2\Guided\guided 1> 
```

#### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program meminta pengguna untuk menginputkan 3 inputan string yang disimpan ke dalam variable satu, dua, dan tiga. Lalu menghasilkan outputan awal dengan mengurutkan ketiga inputan string sesuai dengan urutan inputannya.

### 2. Guided 2

#### Source code

```
package main
```

```
import "fmt"

func main() {
    year := 0
    fmt.Print("Tahun: ")
    fmt.Scan(&year)
    fmt.Println("Kabisat: ", year%4 == 0 && year%1000 !=
0)
}
```

### Screenshoot program

```
PS D:\Praktikum ALpro 2\Modul 2\Guided\guided 2> go run "d:\Praktikum ALpro 2\Modul 2\Guided\guided 2\guided2_rora_220.go"
Tahun: 1978
Kabisat: false
PS D:\Praktikum ALpro 2\Modul 2\Guided\guided 2> go run "d:\Praktikum ALpro 2\Modul 2\Guided\guided 2\guided2_rora_220.go"
Tahun: 2004
Kabisat: true
PS D:\Praktikum ALpro 2\Modul 2\Guided\guided 2> []
```

### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini digunakan untuk menentukan benar tidaknya perhitungan tahun kabisat. Pada program pengguna diminta untuk menginputkan tahun yang nantinya program akan mengecek dengan menggunakan rumus yang telah dirancang untuk menentukan apakah tahun yang diinputkan habis dibagi 4, apabila kondisi ini terpenuhi, maka tahun yang diinputkan adalah tahun kabisat.

## 3. Guided 3

### Source code

```
package main

import (
    "fmt"
    "math"
)

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

### Screenshoot program



```
PS D:\Praktikum ALpro 2\Modul 2\Guided\guided 3> go run "d:\Praktikum ALpro 2\Modul 2\Guided\guided 3\guided3_rora_220.go"
Jejari = 10
Bola dengan jejari 10 memiliki volume 4188.7902 dan luas kulit 1256.6371
PS D:\Praktikum ALpro 2\Modul 2\Guided\guided 3> []
```

## Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini menghitung volume dan luas permukaan bola berdasarkan jari-jari yang diinputkan oleh pengguna. Pengguna akan diminta untuk menginputkan jari-jari yang lalu program akan menghitung volume bola dengan rumus  $\frac{4}{3} * \pi * r^3$  dan luas permukaan bola menggunakan rumus  $4 * \pi * r^2$ .

### III. UNGUIDED

#### 1. Unguided 2b1

##### Source code

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    cairanWarna_220 := []string{"merah", "kuning",
    "hijau", "ungu"}
    scanner := bufio.NewScanner(os.Stdin)
    success := true

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)
        if !scanner.Scan() {
            fmt.Println("Input error")
            return
        }
        warna := strings.Fields(scanner.Text())

        if len(warna) != 4 {
            success = false
            break
        }

        for j, warna := range warna {
            if warna != cairanWarna_220[j] {
                success = false
                break
            }
        }

        if !success {
            break
        }
    }

    fmt.Printf("BERHASIL: %v\n", success)
}
```

##### Screenshoot program

```

PS D:\Praktikum ALpro 2\Modul 2\Unguided> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\2b1_rora_220.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:\Praktikum ALpro 2\Modul 2\Unguided> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\2b1_rora_220.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
BERHASIL: false
PS D:\Praktikum ALpro 2\Modul 2\Unguided> █

```

### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini digunakan untuk memeriksa apakah input dari pengguna sesuai dengan urutan warna yang diharapkan dalam lima percobaan. Pada program dilakukan lima kali percobaan sehingga pengguna diminta untuk menginputkan empat warna dalam satu baris, setiap inputan warna dipisahkan menggunakan spasi dan dibandingkan dengan warna target. Apabila urutan yang diinputkan tidak sesuai atau kurang dari 4 warna, maka program akan menampilkan BERHASIL: false. Namun jika urutan yang diinputkan sesuai maka program akan menampilkan BERHASIL: true sebagai outputan.

## 2. Unguided 2b2

### Source code

```

package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    pita_220 := ""
    var bunga_220List []string

    scanner := bufio.NewScanner(os.Stdin)

    fmt.Print("N: ")
    scanner.Scan()
    input := scanner.Text()
    if input == "0" {
        fmt.Println("Pita:")
        fmt.Println("Bunga: 0")
        return
    }
}

```

```

count := 0
for {
    count++
    fmt.Printf("Bunga %d: ", count)
    scanner.Scan()
    bunga_220 := scanner.Text()

    if strings.ToUpper(bunga_220) == "SELESAI" {
        count--
        break
    }

    if pita_220 == "" {
        pita_220 = bunga_220
    } else {
        pita_220 += "-" + bunga_220
    }

    bunga_220List = append(bunga_220List, bunga_220)
}

fmt.Println("Pita:", pita_220)
fmt.Printf("Bunga: %d\n", len(bunga_220List))
}

package main

import (
    "fmt"
    "strings"
)

func main() {
    var n int
    var bunga_220, pita_220 string
    for i := 0; ; i++ {
        n = i
        fmt.Print("Bunga ", i+1, ": ")
        fmt.Scan(&bunga_220)
        if strings.ToLower(bunga_220) == "selesai" {
            break
        }
        pita_220 += bunga_220 + " - "
    }
    fmt.Println("Pita: ", pita_220)
    fmt.Println("Bunga: ", n)
}

```

## Screenshoot program

```

PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2\2b2_rora_220.go"
N : 3
Bunga_220 1: Kertas
Bunga_220 2: Mawar
Bunga_220 3: Tulip
Pita_220: Kertas - Mawar - Tulip -
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2\2b2_rora_220.go"
N : 0
Pita_220:
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2>

PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2\modifikasi_2b2_220.go"
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Bunga 4: SELESAI
Pita: Kertas - Mawar - Tulip -
Bunga: 3
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2\modifikasi_2b2_220.go"
Bunga 1: SELESAI
Pita:
Bunga: 0
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b2>

```

### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini meminta pengguna untuk menginputkan nama bunga, yang kemudian nantinya program akan menambahkan setiap nama bunga ke dalam string pita, dimana setiap nama bunga akan dipisahkan dengan tanda “-“. Program pertama-tama akan meminta pengguna untuk menginputkan integer n untuk menyimpan jumlah bunga yang akan kita masukkan. Dalam loop sebanyak n kali, pengguna diminta untuk menginputkan nama bunga, lalu setiap nama bunga ditambahkan ke dalam variable pita dan diikuti dengan tanda “-“. Setelah semua bunga dimasukkan, program akan menampilkan hasil akhir string pita dengan format “bunga 1 – bunga 2 - ...” sebagai outputan.

Pada modifikasi program pada loop tidak terhingga sehingga program meminta pengguna untuk memasukkan nama bunga dan setiap nama bunga akan dimasukkan ke dalam string pita, dan dipisahkan dengan “-“. Ketika pengguna mengetik selesai program keluar dari loop sehingga program mencetak hasil akhir dari string pita yang berisi nama-nama bunga yang diinputkan serta total jumlah yang dicatat pada variable n pada outputan.

## 3. Unguided 2b3

### Source code

```

package main

import "fmt"

func main() {
    var kiri_220, kanan_220 float32
    for kiri_220 < 9 && kanan_220 < 9 {

```

```

        fmt.Print("Masukan berat belanjaan di kedua kantong:
")
        fmt.Scan(&kiri_220, &kanan_220)
    }
    fmt.Println("Proses selesai.")
}

package main

import (
    "fmt"
    "math"
)

func main() {
    var kiri_220, kanan_220 float32

    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong:
")
        fmt.Scan(&kiri_220, &kanan_220)

        if kiri_220 < 0 || kanan_220 < 0 || (kiri_220+kanan_220)
> 150 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := math.Abs(float64(kiri_220 - kanan_220))
        oleng := selisih >= 9

        fmt.Println("Sepeda motor pak Andi akan oleng:", oleng)
    }
}

```

## Screenshoot program

```

PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b3> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b3\2b3_rora_220.go"
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.
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b3>

PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b3> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b3\2b3_rora_220.go"
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
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.
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b3>

```

## Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menerima inputan berjalan dari dua kantong, kiri dan kanan dan untuk menentukan apakah sepeda motor pak Andi akan oleng

berdasarkan perbandingan berat kedua kantong tersebut. Program meminta pengguna untuk menginputkan berat belanjaan di kedua kantong secara berulang selama total berat belanjaan pada kantong kurang dari 150.

Pada modifikasi program pada loop, setelah diinputkan berat belanjaan pada kedua kantong program mencetak pernyataan apakah sepeda motor pak andi akan oleng atau tidak, dengan menggunakan kondisi  $kiri > kanan + 9$  dan  $kanan > kiri + 9$ . Apabila satu kantong beratnya lebih dari 9kg dan melebihi berat kantong yang lain maka sepeda motor pak andi akan oleng. Setelah total berat melebihi 150, program akan keluar dari loop dan mencetak "Proses selesai" sebagai outputan.

#### 4. Unguided 2b4

##### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var k float64

    fmt.Print("Nilai K = ")
    fmt.Scan(&k)

    numerator_220 := math.Pow((4*k + 2), 2)
    denominator_220 := (4*k + 1) * (4*k + 3)
    fk := numerator_220 / denominator_220

    fmt.Printf("Nilai f(K) = %.10f\n", fk)
}
```

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var K int

    fmt.Print("Nilai K = ")
    fmt.Scan(&K)

    result := 1.0
    for k := 0; k <= K; k++ {
        numerator := math.Pow((4*float64(k) + 2), 2)
```

```

        denominator := (4*float64(k) + 1) * (4*float64(k) + 3)
        result *= numerator / denominator
    }

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

```

### Screenshot program

```

PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4\2b4_rora_220.go"
Nilai K = 100
Nilai f(K) = 1.0000061880
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4>

220.go"
Nilai K = 10
Nilai akar 2 = 1.4062058441
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4\2b4_rora_220.go"
Nilai K = 100
Nilai akar 2 = 1.4133387072
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4>
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4\2b4_rora_220.go"
Nilai K = 1000
Nilai akar 2 = 1.4141252651
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2b4>

```

### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menghitung nilai fungsi matematika berdasarkan input nilai k yang diinputkan oleh pengguna. Program meminta pengguna untuk menginputkan nilai k sehingga program menggunakan rumus pada soal untuk menghitung hasilnya. Hasil akhir dicetak dengan format yang menunjukkan 10 angka decimal setelah titik decimal.

Pada modifikasi program setelah pengguna menginputkan nilai k, program melakukan iterasi dari 0 hingga k-1 menggunakan loop for. Dimana pada setiap iterasinya program menghitung nilai berdasarkan rumus deret akar 2 dan mengalikan hasilnya dengan variable hasil yang diinisialisasi dengan 1. Setelah semua iterasi diselesaikan program mencetak hasil akhir dengan format yang sama.

## 5. Unguided 2c1

### Source code

```

package main

import (
    "fmt"
)

func main() {
    var beratParcel int

    // Meminta input berat parsel dalam gram
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&beratParcel)
}

```



```

// Menghitung total berat dalam kg dan sisa dalam gram
totalKg_220 := beratParcel / 1000
sisaGr_220 := beratParcel % 1000

// Menghitung biaya dasar
biayaDasar := totalKg_220 * 10000
var biayaSisa int

// Menghitung biaya tambahan berdasarkan sisa berat
if totalKg_220 > 10 {
    biayaSisa = 0 // Sisa berat gratis jika total lebih
dari 10kg
} else {
    if sisaGr_220 >= 500 {
        biayaSisa = sisaGr_220 * 5 // Rp. 5,- per gram
    } else {
        biayaSisa = sisaGr_220 * 15 // Rp. 15,- per gram
    }
}

// Menampilkan detail berat dan biaya
fmt.Printf("Detail berat: %d kg %d gr\n", totalKg_220,
sisaGr_220)
fmt.Printf("Detail biaya: Rp. %d Rp. %d\n", biayaDasar,
biayaSisa)
fmt.Printf("Total biaya: Rp. %d\n", biayaDasar+biayaSisa)
}

```

### Screenshoot program

```

PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c1> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c1\2c1_rora_220.go"
Berat parsel (gram): 9250
Detail berat: 9 kg 250 gr
Detail biaya: Rp. 90000 Rp. 3750
Total biaya: Rp. 93750
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c1> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c1\2c1_rora_220.go"
Berat parsel (gram): 11750
Detail berat: 11 kg 750 gr
Detail biaya: Rp. 110000 Rp. 0
Total biaya: Rp. 110000
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c1>
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c1>
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c1> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c1\tempCodeRunnerFile.go"
Berat parsel (gram): 8500
Detail berat: 8 kg 500 gr
Detail biaya: Rp. 80000 Rp. 2500
Total biaya: Rp. 82500

```

### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menghitung total biaya pengiriman berdasarkan berat parsel yang dimasukkan oleh pengguna dalam gram. Biaya pengiriman dasar dihitung sebagai original, yang merupakan biaya untuk setiap kilogram, ditetapkan pada Rp. 10.000 per kilogram. Jika berat kilogram kurang dari 10, program kemudian menentukan biaya tambahan berdasarkan berat gram: jika kurang dari 500 gram, biaya tambahan dihitung dengan mengalikan gram dengan 15; jika 500 gram atau lebih, biaya tambahan dihitung dengan mengalikan gram dengan 5. Setelah semua biaya

dihitung, program mencetak detail berat, biaya dasar, biaya tambahan, dan total biaya pengiriman.

## 6. Unguided 2c2

- a. Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?

Keluaran dari program tersebut jika nam yang diberikan adalah 80.1 adalah nilai "A". Eksekusi program ini benar apabila hanya meninjau kondisi bahwa nam lebih besar dari 80. Akan pada program ini tidak sesuai dengan spesifikasi soal karena pada table, nilai yang diberikan untuk  $80 < \text{nam} \leq 100$  adalah "A", sedangkan kondisi  $72.5 < \text{nam} \leq 80$  adalah "AB".

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

Kesalahan dari program ini pada penggunaan operator perbandingan, terutama ketika menangani batasan rentang nilai, yang mana pada program, kondisi rentang nilai terdapat yang salah.

Sebagai contoh, yang mana pada tabel seharusnya:

$72.5 < \text{NAM} \leq 80$  yaitu senilai dengan "AB" akan tetapi dalam program tidak ada kondisi untuk batas atas 80, sehingga semua nilai lebih besar dari 80 akan menghasilkan A. selain itu seharusnya program memeriksa kondisi rentang nilai dengan kondisi untuk batasnya.

- c. Perbaiki program tersebut! Ujilah dengan masukan 93.5; 70.6; dan 49,5. Seharusnya keluaran yang diperoleh adalah 'A', 'B' dan 'D'

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

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

```

### Screenshoot program

```

PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c2> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c2\2c2_rora_220.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c2> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c2\2c2_rora_220.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c2> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c2\2c2_rora_220.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c2>

```

### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menentukan nilai huruf dari nilai akhir mata kuliah berdasarkan input nilai yang diinputkan oleh pengguna. Program meminta pengguna menginputkan nilai akhir dalam bentuk angka decimal, program kemudian menggunakan serangkaian pernyataan kondisional if dan else if untuk menentukan kategori nilai huruf yang sesuai.

## 1. Unguided 2c3

### Source code

```

package main

import "fmt"

func main() {
    var bil_220 int
    var prima bool = true

```

```

fmt.Print("Bilangan: ")
fmt.Scan(&bil_220)
fmt.Print("Faktor: ")

for i := 1; i <= bil_220; i++ {
    if bil_220%i == 0 {
        fmt.Print(i, " ")
        if !(i == 1 || i == bil_220) {
            prima = false
        }
    }
}

fmt.Println()
fmt.Print("Prima: ", prima)
}

```

### Screenshoot program

```

PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c3> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c3\2c3_rora_228.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c3> go run "d:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c3\2c3_rora_228.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Praktikum ALpro 2\Modul 2\Unguided\unguided 2c3> 

```

### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menentukan faktor-faktor dari suatu bilangan dan memeriksa apakah bilangan tersebut merupakan bilangan prima. Program meminta pengguna untuk menginputkan sebuah bilangan, sehingga program akan mencetak semua factor dari bilangan dengan menggunakan loop. Di dalam loop, program memeriksa apakah factor tersebut bukan 1 atau bilangan itu sendiri. Jika ada factor lain, maka variable prima diubah menjadi false. Setelah loop selesai program akan mencetak semua factor dan menunjukkan apakah bilangan tersebut adalah bilangan prima atau tidak.