

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2  
MODUL 2  
REVIEW STRUKTUR KONTROL**



Oleh:

PANDIA ARYA BRATA

2311102076

IF – 11 - 02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2024**

## I. DASAR TEORI

### 2.1 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.

Komentar, bukan bagian dari kode program, dan dapat ditulis di mana saja di dalam program:

- Satu baris teks yang diawali dengan garis miring ganda ("//") s.d. akhir baris, atau.
- Beberapa baris teks yang dimulai dengan pasangan karakter "/\*" dan diakhiri dengan "\*/"

```
package main
import "fmt"
func main() {
    var greetings = "Selamat datang di dunia DAP"
    var a, b int

    fmt.Println(greetings)
    fmt.Scanln(&a, &b)
    fmt.Printf("%v + %v = %v\n", a, b, a+b)
}
```

### Koding, Kompilasi, dan Eksekusi Go

#### Koding

- Tidak berbeda dengan penulisan program sumber dalam bahasa lain, program Go harus dibuat menggunakan penyunting teks dan disimpan dalam format teks, bukan dalam format dokumen (doc, docx, atau lainnya)
- Setiap program go disimpan dalam file teks dengan ekstensi .go, dengan nama bebas. Sebaiknya nama file adalah nama untuk program tersebut,
- Setelah satu program lengkap Go disimpan dalam satu folder tersendiri. Nama folder merupakan nama program tersebut, Karena itu secara prinsip, satu program Go dapat dipecah dalam beberapa file dengan ekstensi \*.go selama disimpan dalam folder yang sama:

## Kompilasi

Beberapa bahasa pemrograman dirancang untuk diimplementasikan sebagai interpreter dan lainnya sebagai kompilator. Interpreter akan membaca setiap baris intruksi dan kemudian langsung mengeksekusinya, dengan hanya sedikit pemeriksaan apakah penulisan keseluruhan program sudah benar atau belum. Kompilator akan memeriksa keseluruhan program sumber dan kemudian mengubahnya menjadi program eksekutabel, sehingga konsistensi penulisan (seperti penggunaan tipe data) sudah diperiksa sebelum eksekusi. Selain itu karena program dibuat menjadi eksekutabel lebih dahulu, proses optimasi dapat dilakukan sehingga program menjadi sangat efisien. Catatan : 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

Variabel adalah nama dari suatu lokasi di memori, yang data dengan tipe tertentu dapat disimpan. ' Nama variabel dimulai dengan huruf dan dapat diikuti dengan sejumlah huruf, angka, atau garisbawah

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)	<b>bool</b>	<b>false</b> dan <b>true</b>
karakter	byte	tabel tabel
string	string	

- Tipe data yang umum tersedia adalah integer, real, boolean, karakter, dan string. Lihat tabel berikut ini untuk variasi tipe data yang disediakan dalam bahasa Go.
- Nilai data yang tersimpan dalam variabel dapat diperoleh dengan menyebutkan langsung nama variabelnya. Contoh: Menyebutkan nama found akan mengambil nilai tersimpan dalam memori untuk variabel found, pastinya.
- Informasi alamat atau lokasi dari variabel dapat diperoleh dengan menambahkan prefiks & di depan nama variabel tersebut. Contoh: &found akan mendapatkan alamat memori untuk menyimpan data pada found.
- Jika variabel berisi alamat memori, prefiks \* pada variabel tersebut akan memberikan nilai yang tersimpan dalam memori yang lokasinya disimpan dalam variabel tersebut.

## Struktur Kontrol Perulangan

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

1	
2	
3	
4	}
5	for kondisi {
6	// .. ulangi kode di sini selama kondisi terpenuhi
7	// .. sama seperti "for ; kondisi; {"
8	}
9	for {
10	// .. tanpa kondisi, berarti loop tanpa henti (perlu if-break)
11	}
12	for ndx,
13	
14	
15	}

Bentuk While-Loop Bentuk while-loop memastikan setiap kali memasuki loop, ada kondisi yang harus terpenuhi (benar>true). Inijuga berarti saat keluar dari loop, maka nilai kondisi tersebut pasti salah>false!

	Notasi algoritma	Penulisan dalam bahasa Go
1	e <-	e :=
2	x <-	x :=
3	y <- 0.0	y := 0.0
4	y1 <- x	y1 := x
5	while y1-y > e or y1-y < -e do	for -y > e    -y < -e {
6	y <- y1	y = y1

Bentuk Repeat-Until Bentuk repeat-until di perulangan dilakukan terus menerus sampai kondisi keluar terpenuhi. Artinya selama kondisi belum terpenuhi (salah>false) maka perulangan akan terus dilakukan. Pada saat keluar dari loop maka nilai kondisi pasti benar>true!

	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		for selesai:=false; !selesai;
6		selesai=kondisi {
7		..kode yang diulang
8		}
9		

## Struktur Kontrol Percabangan

Untuk analisa kasus, bahasa Go mendukung dua bentuk percabangan, yaitu if-else dan switch-case. 1) Bentuk If-Else Berikut ini bentuk-bentuk if-else yang mungkin dilakukan dalam bahasa Go. Semua bentuk di bawah merupakan satu instruksi if-else-endif saja (hanya satu endif). Bentuk if-else yang bersarang (dengan beberapa endif) dapat dibentuk dengan komposisi beberapa if-else-endif tersebut

	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
8	endif	}
9		if kondisi_1 {
10		.. kode untuk kondisi_1 true
11		} else if kondisi_2 {
12		.. kode untuk kondisi
13		.. 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 Dalam bahasa Co ada dua variasi bentuk switch-case. Bentuk yang biasa digunakan adalah ekspresi ditulis pada perintah switch dan nilai ditulis dalam setiap label case-nya. Bentuk yang kedua mempunyai

switch tanpa ekspresi, tetapi setiap case boleh berisi ekspresi boolean. Tentunya bentuk yang kedua lebih bersifat umum, dan merupakan penyederhanaan bentuk

	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		}

## I. GUIDED

### 1.GUIDED (No.1|2A)

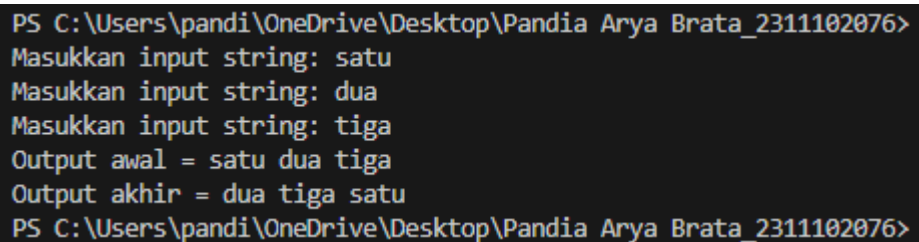
```
package main

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)
}
```

**Screenshot program:**



```
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
Masukkan input string: satu
Masukkan input string: dua
Masukkan input string: tiga
Output awal = satu dua tiga
Output akhir = dua tiga satu
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

**Deskripsi program :**

Program ini menukar urutan dari tiga kata yang dimasukkan pengguna dengan cara menyimpan nilai sementara salah satu kata, lalu menggeser nilai dari kata lainnya secara berurutan.

### 2.GUIDED(No.2|2A)

```
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)))
}
```

#### Screenshot program :

```
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
Tahun : 2005
Kabisat : false
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

#### Deskripsi program :

Program ini menentukan apakah suatu tahun yang diinputkan oleh pengguna merupakan tahun kabisat atau bukan berdasarkan aturan tahun kabisat.

### 3.GUIDED(No.3|2A)

```
package main

import (
    "fmt"
    "math"
)

func main() {

    var jari, volume, luasKulit float64
    fmt.Print("Jari-jari = ")
    fmt.Scan(&jari)
    volume = math.Pi * (4.0 / 3.0) * math.Pow(jari, 3)
    luasKulit = 4 * math.Pi * math.Pow(jari, 2)
    fmt.Printf("Bola dengan jejari %v memiliki volume %.4f
dan luas kulit %.4f \n", jari, volume, luasKulit)
}
```

#### Screenshot program :

```
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076> go run
Jejari = 10
Bola dengan jejari 10 memiliki volume 4188.7902 dan luas kulit 1256.6371
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076> o
```



**Deskripsi program :**

Program ini menghitung volume dan luas permukaan bola berdasarkan jari-jari yang dimasukkan oleh pengguna.

**II. UNGUIDED****1. UNGUIDED(No.4|2A)**

```
package main

import "fmt"

func reamur(ce1 float64) {
    r := (4.0 / 5.0) * ce1
    fmt.Println("Derajat Reamur : ", r)
}

func fahrenheit(ce1 float64) {
    f := (9.0 / 5.0 * ce1) + 32
    fmt.Println("Derajat Fahrenheit : ", f)
}

func kelvin(ce1 float64) {
    k := ce1 + 273.15
    fmt.Println("Derajat Kelvin : ", k)
}

func main() {
    var ce1 float64
    fmt.Print("Derajat Celcius : ")
    fmt.Scan(&ce1)
```

```

    reamur(cel)
    fahrenheit(cel)
    kelvin(cel)
}

```

**Screenshot program :** =

```

PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
Derajat Celcius : 50
Derajat Reamur : 40
Derajat Fahrenheit : 122
Derajat Kelvin : 323.15
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>

```

**Deskripsi program :**

Program ini adalah sebuah konverter suhu. Program ini dirancang untuk menerima input suhu dalam derajat Celcius dari pengguna, lalu melakukan konversi ke tiga satuan suhu lainnya yaitu Reamur, Fahrenheit, dan Kelvin. Setiap konversi dilakukan oleh fungsi yang berbeda, dan hasil konversi akan ditampilkan di layar. Intinya, program ini membantu pengguna untuk dengan mudah mengubah satuan suhu dari Celcius ke satuan suhu lainnya.

## 2.UNGUIDED(No.5|2A)

```

package main

import "fmt"

func main() {
    var c1, c2, c3, c4, c5, c6, c7, c8 byte

    fmt.Scanf("%d%d%d%d", &c1, &c2, &c3, &c4, &c5)
    fmt.Scanf("\n")
    fmt.Scanf("%c%c%c", &c6, &c7, &c8)
    fmt.Printf("%c%c%c%c \n", c1, c2, c3, c4, c5)
    fmt.Printf("%c%c%c \n", (c6 + 1), (c7 + 1), (c8 + 1))
}

```

### Screenshot program :

```
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
66 97 103 117 115
SNO
Bagus
TOP
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

### Deskripsi program :

Program ini menerima input berupa lima angka desimal dan tiga karakter. Angka-angka desimal disimpan dalam variabel c1, c2, c3, c4, dan c5, sedangkan karakter-karakter disimpan dalam variabel c6, c7, dan c8. Program kemudian mencetak angka-angka desimal dan karakter-karakter yang telah diinput, serta karakter-karakter yang telah diinkremenkan (ditambah 1).

### 3.UNGUIDED(No.1|2B)

```
package main

import "fmt"

func reamur(cel float64) {
    r := (4.0 / 5.0) * cel
    fmt.Println("Derajat Reamur : ", r)
}

func fahrenheit(cel float64) {
    f := (9.0 / 5.0 * cel) + 32
    fmt.Println("Derajat Fahrenheit : ", f)
}

func kelvin(cel float64) {
    k := cel + 273.15
    fmt.Println("Derajat Kelvin : ", k)
}

func main() {
    var cel float64
    fmt.Print("Drajat Celcius : ")
    fmt.Scan(&cel)
    reamur(cel)
    fahrenheit(cel)
    kelvin(cel)
}
```

#### Screenshot program :

```
Percobaan ke-1 = merah kuning hijau ungu
Percobaan ke-2 = merah kuning hijau ungu
Percobaan ke-3 = merah kuning hijau ungu
Percobaan ke-4 = ungu kuning hijau merah
Percobaan ke-5 = merah kuning hijau ungu
BERHASIL = false
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

#### Deskripsi program :

Program ini dirancang untuk **memverifikasi urutan warna** yang dimasukkan oleh pengguna. Program akan menjalankan 5 kali percobaan, di mana pada setiap percobaan, pengguna diminta untuk memasukkan 4 warna. Warna-warna yang dimasukkan kemudian akan dibandingkan dengan urutan warna yang sudah ditentukan dalam program.

#### 4.UNGUIDED(No.2|2B)

```
package main

import (
    "fmt"
)

func main() {
    var j int
    fmt.Print("Masukan Jumlah : ")
    fmt.Scan(&j)

    if j == 0 {
        fmt.Print("Pita : ")
    } else {
        bunga := make([]string, j)
        for i := 0; i < j; i++ {
            fmt.Printf("Bunga %v : ", i+1)
            fmt.Scan(&bunga[i])
        }
        var gabung string
        for i := 0; i < j; i++ {
            gabung += bunga[i] + " - "
        }
        fmt.Print(gabung)
    }
}
```

#### Screenshot program :

```
Masukan Jumlah : 3
Bunga 1 : tulip
Bunga 2 : mawar
Bunga 3 : kertas
tulip - mawar - kertas -
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

**Deskripsi program :**

Program ini dirancang untuk **mengumpulkan nama-nama bunga dan menggabungkannya menjadi satu string**. Program akan meminta pengguna untuk memasukkan jumlah bunga yang ingin diinputkan. Jika jumlah bunga yang dimasukkan adalah 0, maka program akan langsung mencetak "Pita". Jika tidak, program akan meminta pengguna untuk memasukkan nama-nama bunga satu per satu, kemudian menggabungkan semua nama bunga tersebut menjadi satu string yang dipisahkan oleh tanda "-".

**5.UNGUIDED(No.2|2B|Modif)**

```
package main

import (
    "fmt"
)

func main() {
    var input string
    var gabung string
    var j = 0
    fmt.Printf("Bunga %v= : ", j+1)
    fmt.Scan(&input)
    if input == "selesai" {
        fmt.Println("Pita : ")
        fmt.Println("Bunga : 0")
    } else {
        for input != "selesai" {
            if input != "selesai" {
                gabung += input + " - "
            }
            fmt.Printf("Bunga %v= : ", j+2)
            fmt.Scan(&input)
            j++
        }

        fmt.Println(gabung)
        fmt.Print("bunga : ", j)
    }
}
```

#### Screenshot program :

```
Bunga 1= : mawar
Bunga 2= : kertas
Bunga 3= : ungu
Bunga 4= : selesai
mawar - kertas - ungu -
bunga : 3
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

#### Deskripsi program :

Program ini dirancang untuk **mengumpulkan nama-nama bunga hingga pengguna memasukkan kata "selesai"**. Program akan terus meminta input nama bunga sampai pengguna memasukkan kata "selesai". Setelah itu, program akan menampilkan semua nama bunga yang telah diinputkan dalam satu baris, dipisahkan oleh tanda "-". Selain itu, program juga akan menampilkan jumlah bunga yang telah diinputkan.

### 6.UNGUIDED(No.3|2B)

```
package main

import "fmt"

func main() {
    var v1, v2 float64
    var ulang = true
    for ulang != false {
        fmt.Print("Masukan belanjaan di kedua kantong : ")
        fmt.Scan(&v1, &v2)
        if v1 == 9 || v2 == 9 {
            ulang = false
        }
    }
    fmt.Print("Proses selesai")
}
```

#### Screenshot program :

```
Masukan belanjaan di kedua kantong : 5.5 1.0
Masukan belanjaan di kedua kantong : 7.1 8.5
Masukan belanjaan di kedua kantong : 2 6
Masukan belanjaan di kedua kantong : 9 5.8
Proses selesai
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

#### Deskripsi program :

Program ini dirancang untuk **menerima input berupa dua bilangan desimal secara berulang hingga salah satu bilangan tersebut bernilai**

9. Program akan terus meminta input dari pengguna sampai kondisi tersebut terpenuhi.

## 7.UNGUIDED(No.3|2B|Modif)

```
package main

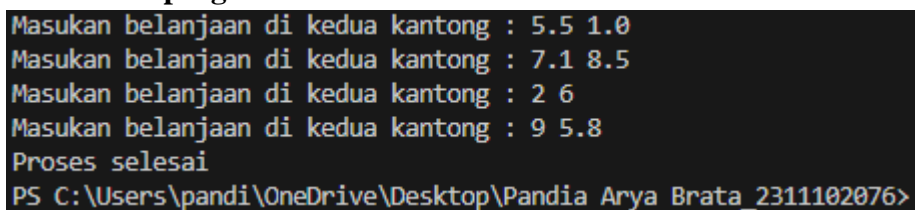
import "fmt"

func main() {
    var v1, v2 float64
    var jatuh = false
    var ulang = true
    for ulang != false {

        fmt.Print("Masukan belanjaan di kedua kantong : ")
        fmt.Scan(&v1, &v2)
        if v1-v2 >= 9 || v2-v1 >= 9 {
            jatuh = true
        } else {
            jatuh = false
        }
        if v1+v2 >= 150 || v1 < 0 || v2 < 0 {
            ulang = false
        } else {
            fmt.Println("Speda motor pak Andi akan oleng : ", jatuh)
        }

        fmt.Print("Proses selesai")
    }
}
```

### Screenshot program :



```
Masukan belanjaan di kedua kantong : 5.5 1.0
Masukan belanjaan di kedua kantong : 7.1 8.5
Masukan belanjaan di kedua kantong : 2 6
Masukan belanjaan di kedua kantong : 9 5.8
Proses selesai
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

### Deskripsi program :

Program ini dirancang untuk mensimulasikan kondisi sepeda motor Pak Andi saat membawa dua kantong belanja. Program akan terus meminta input berupa berat belanjaan di kedua kantong hingga total berat melebihi 150, salah satu kantong beratnya negatif, atau selisih berat kedua kantong mencapai 9 atau lebih.

## 8.UNGUIDED(No.4|2B)

```
package main

import "fmt"

func main() {
    var f, k float64

    fmt.Print("Nilai K : ")
    fmt.Scan(&k)
    f = ((4*k + 2) * (4*k + 2)) / ((4*k + 1) * (4*k + 3))
    fmt.Printf("Nilai f(k) : %.10f", f)
}
```

#### Screenshot program :

```
Nilai K : 100
Nilai f(k) : 1.0000061880
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

#### Deskripsi program :

Program ini dirancang untuk **menghitung nilai suatu fungsi matematika** yang telah didefinisikan dalam bentuk rumus tertentu. Rumus ini melibatkan variabel k dan digunakan untuk menghitung nilai f(k).

### 9.UNGUIDED(No.4|2B|Modif)

```
package main

import "fmt"

func main() {
    var f, k float64

    fmt.Print("Nilai K : ")
    fmt.Scan(&k)
    f = ((4*k + 2) * (4*k + 2)) / ((4*k + 1) * (4*k + 3))
    fmt.Printf("Nilai f(k) : %.10f", f)
}
```

#### Screenshot program :

```
Nilai K : 10
Nilai akar 2 : 1.4062058441
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
```

#### Deskripsi program :

Program ini dirancang untuk menghitung perkiraan nilai akar kuadrat dari 2 menggunakan sebuah rumus matematika tertentu. Rumus ini merupakan salah satu metode untuk menghitung akar kuadrat secara numerik.

### 10.UNGUIDED(No.1|2C)



```

package main

import "fmt"

func main() {
    var gr, kg, sisa, harga, harga2, total int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&gr)
    kg = gr / 1000
    sisa = gr % 1000
    fmt.Printf("Detail berat : %d kg + %d gr", kg, sisa)
    fmt.Println("")
    harga = 10000 * kg
    if sisa >= 500 {
        if kg > 10 {
            harga2 = 5 * sisa
            fmt.Printf("Detail biaya: Rp.%d kg + Rp.%d",
harga, harga2)
            total = harga
            fmt.Println("")
            fmt.Println("Total biaya : Rp.", total)
        } else {
            harga2 = 5 * sisa
            fmt.Printf("Detail biaya: Rp.%d kg + Rp.%d",
harga, harga2)
            total = harga + harga2
            fmt.Println("")
            fmt.Println("Total biaya : Rp.", total)
        }
    } else {
        if kg > 10 {
            harga2 = 5 * sisa
            fmt.Printf("Detail biaya: Rp.%d kg + Rp.%d",
harga, harga2)
            total = harga
            fmt.Println("")
            fmt.Println("Total biaya : Rp.", total)
        } else {
            harga2 = 15 * sisa
            fmt.Printf("Detail biaya: Rp.%d kg + Rp.%d",
harga, harga2)
            total = harga + harga2
            fmt.Println("")
            fmt.Println("Total biaya : Rp.", total)
        }
    }
}

```

#### Screenshot program :

```

Berat parsel (gram): 8500
Detail berat : 8 kg + 500 gr
Detail biaya: Rp.80000 kg + Rp.2500
Total biaya : Rp. 82500

```

```

Berat parcel (gram): 9250
Detail berat : 9 kg + 250 gr
Detail biaya: Rp.90000 kg + Rp.3750
Total biaya : Rp. 93750
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata 2311102076>
Berat parcel (gram): 11750
Detail berat : 11 kg + 750 gr
Detail biaya: Rp.110000 kg + Rp.3750
Total biaya : Rp. 110000

```

#### **Deskripsi program :**

Program ini dirancang untuk **menghitung biaya pengiriman suatu parcel** berdasarkan beratnya dalam gram. Program ini juga memperhitungkan biaya tambahan berdasarkan berat sisa (dalam gram) dan total berat dalam kilogram.

### **11.UNGUIDED(No.2|2C)**

#### **Sourcecode sebelum :**

```

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 <=40 {
        nam = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

#### **Sourcecode sesudah :**

```

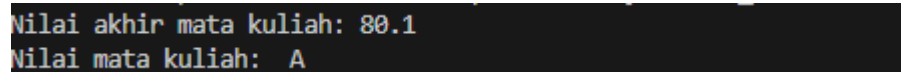
package main

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 :



```

Nilai akhir mata kuliah: 80.1
Nilai mata kuliah: A

```

#### Deskripsi program :

**A** > output A, ya output sesuai dengan program dikarenakan jika `nam > 80` maka output yang dihasilkan adalah **A**, sedangkan `nam` tersebut 80,1.

**B** > Program ini memiliki banyak kesalahan yang membuat program tidak berjalan, beberapa letak kesalahan yaitu :

- if yang berulang, seharusnya jika percabangan 2 atau lebih kondisi menggunakan `if`, `else if`, dan diakhiri `else`, dan di `else` tidak perlu memiliki kondisi.
- inisiasi variable dalam suatu percabangan salah yaitu "`nam`", seharusnya yang benar adalah "`nmk`", karena `nmk` sendiri variable yang dipanggil di output akhir.

C > SESUAI

```
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\pandi\OneDrive\Desktop\Pandia Arya Brata_2311102076>
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
```

## 12.UNGUIDED(No.3|2C)

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Bilangan: ")
    fmt.Scan(&n)
    fmt.Print("faktor : ")
    for i := 1; i <= n; i++ {
        if n%i == 0 {
            fmt.Print(" ", i)
        }
    }
}
```

**Screenshot program :**

```
Bilangan: 7
faktor : 1 7
```

**Deskripsi program :**

Program ini dirancang untuk **menampilkan semua faktor dari sebuah bilangan bulat yang diinputkan oleh pengguna**. Faktor dari suatu bilangan adalah bilangan-bilangan yang dapat membagi habis bilangan tersebut tanpa sisa.

## 13.UNGUIDED(No.3|2C|Modif)


```
package main

import "fmt"

func main() {
    var n int
    var prime bool
    fmt.Print("Bilangan: ")
    fmt.Scan(&n)
```

```
    fmt.Print("faktor : ")
    for i := 1; i <= n; i++ {
        if n%i == 0 {
            fmt.Print(" ", i)
        }
    }
    fmt.Print("\n")
    for i := 2; i*i < n; i++ {
        if n%i == 0 {
            prime = false
        } else {
            prime = true
        }
    }
    fmt.Print("prima : ", prime)
}
```

#### **Screenshot program :**

A screenshot of a terminal window with a black background and light blue text. It shows the output of a Go program for the input number 7. The output is: "Bilangan: 7", "faktor : 1 7", and "prima : true".

```
Bilangan:
7
faktor : 1 7
prima : true
```

#### **Deskripsi program :**

Program ini dirancang untuk menentukan apakah sebuah bilangan yang diinputkan oleh pengguna adalah bilangan prima atau bukan, sisanya sama seperti program sebelumnya.