

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**Modul 2 : Review Struktur Kontrol**



Oleh:

ANGGUN WAHYU WIDIYANA

103112480280

12-IF-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
TAHUN AJARAN 2025/2026**

## I. DASAR TEORI

Struktur kontrol merupakan elemen fundamental dalam algoritma pemrograman yang memungkinkan pengembang untuk mengatur aliran eksekusi program. Struktur kontrol menentukan urutan perintah yang akan dijalankan, kapan perintah diulang, dan kapan perintah dilewati berdasarkan kondisi tertentu.

### A. Jenis-Jenis Struktur Kontrol

#### Struktur Sekuensial

Struktur sekuensial adalah struktur paling dasar dimana perintah dieksekusi secara berurutan dari atas ke bawah. Setiap perintah dijalankan tepat satu kali dalam urutan yang ditentukan. Dalam bahasa Go, contoh struktur sekuensial adalah:

```
package main
import "fmt"

func main() {
    var nilai1 = 10
    var nilai2 = 20
    var hasil = nilai1 + nilai2
    fmt.Println("Hasil penjumlahan:", hasil)
}
```

#### Struktur Seleksi/Percabangan

Struktur seleksi memungkinkan program untuk memilih jalur eksekusi berdasarkan kondisi tertentu. Terdapat beberapa bentuk struktur seleksi:

##### a. If-Else

```
if kondisi {
    // kode jika kondisi benar
} else {
    // kode jika kondisi salah
}
```

##### b. If-Else If-Else

```
if kondisi1 {
    // kode jika kondisi1 benar
} else if kondisi2 {
    // kode jika kondisi2 benar
} else {
    // kode jika semua kondisi salah
}
```

##### c. Switch-Case

```
switch variabel {
case nilai1:
    // kode jika variabel == nilai1
```

```

case nilai2:
    // kode jika variabel == nilai2
default:
    // kode jika tidak ada case yang cocok
}

```

## Struktur Perulangan

Struktur perulangan memungkinkan program untuk mengulang eksekusi blok kode selama kondisi tertentu terpenuhi. Beberapa jenis perulangan:

### a. For Loop

```

for inisialisasi; kondisi; perubahan {
    // blok kode yang diulang
}

```

### b. While Loop

```

for kondisi {
    // blok kode yang diulang selama kondisi benar
}

```

### c. Do-While Loop

```

for {
    // blok kode yang diulang
    if !kondisi {
        break
    }
}

```

### d. Foreach Loop

```

for indeks, nilai := range koleksi {
    // blok kode yang diulang untuk setiap elemen koleksi
}

```

## B. Kontrol Aliran dalam Perulangan

### Break

Perintah `break` digunakan untuk keluar dari perulangan sebelum kondisi terminasi normal tercapai.

```

for i := 0; i < 10; i++ {
    if i == 5 {
        break // keluar dari loop ketika i == 5
    }
    fmt.Println(i)
}

```

### Continue

Perintah `continue` digunakan untuk melewati iterasi saat ini dan melanjutkan ke iterasi berikutnya.

```

for i := 0; i < 10; i++ {
    if i%2 == 0 {
        continue // lewati iterasi jika i genap
    }
    fmt.Println(i) // hanya mencetak bilangan ganjil
}

```

### C. Struktur Kontrol Bersarang (Nested Control Structures)

Struktur kontrol dapat disusun secara bersarang untuk menangani logika yang lebih kompleks:

```

for i := 0; i < 5; i++ {
    for j := 0; j < 5; j++ {
        if i == j {
            fmt.Print("* ")
        } else {
            fmt.Print("- ")
        }
    }
    fmt.Println()
}

```

### D. Implementasi dalam Bahasa Go

```

package main

import "fmt"

func main() {
    // Contoh penggunaan berbagai struktur kontrol

    // For loop standar
    fmt.Println("For loop standar:")
    for i := 0; i < 5; i++ {
        fmt.Println(i)
    }

    // While loop (menggunakan for)
    fmt.Println("\nWhile loop:")
    j := 0
    for j < 5 {
        fmt.Println(j)
        j++
    }

    // Infinite loop dengan break
    fmt.Println("\nInfinite loop dengan break:")
}

```

```

counter := 0
for {
    fmt.Println(counter)
    counter++
    if counter >= 5 {
        break
    }
}

// Switch-case
fmt.Println("\nSwitch-case:")
nilai := 85
switch {
case nilai >= 90:
    fmt.Println("A")
case nilai >= 80:
    fmt.Println("B")
case nilai >= 70:
    fmt.Println("C")
default:
    fmt.Println("D")
}
}

```

## II. GUIDED

### Contoh Soal 1

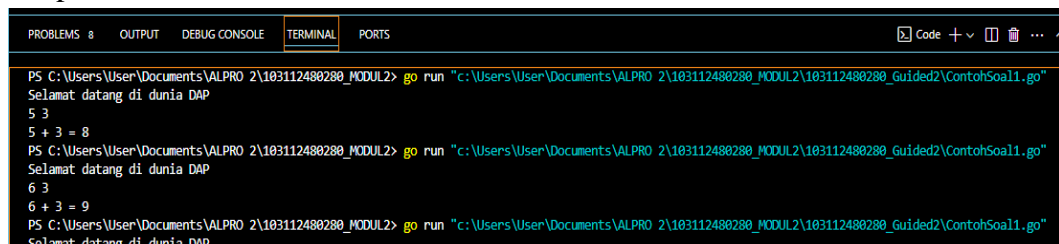
```
//Nama : Anggun Wahyu W. (103112480280)
package main

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)
}
```

Output:



```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_Guided2\ContohSoal1.go"
Selamat datang di dunia DAP
5 3
5 + 3 = 8
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_Guided2\ContohSoal1.go"
Selamat datang di dunia DAP
6 3
6 + 3 = 9
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_Guided2\ContohSoal1.go"
Selamat datang di dunia DAP
```

### Deskripsi Program

Program ini yang menyapa pengguna dan melakukan penjumlahan. Pertama, program menampilkan pesan "Selamat datang di dunia DAP". Kemudian, program meminta pengguna memasukkan dua bilangan bulat. Setelah pengguna memasukkan kedua bilangan tersebut, program akan menjumlahkan kedua bilangan tersebut dan menampilkan hasilnya ke layar dengan format: "[bilangan pertama] + [bilangan kedua] = [hasil penjumlahan]". Singkatnya, program ini adalah program sapaan dan kalkulator penjumlahan sederhana.

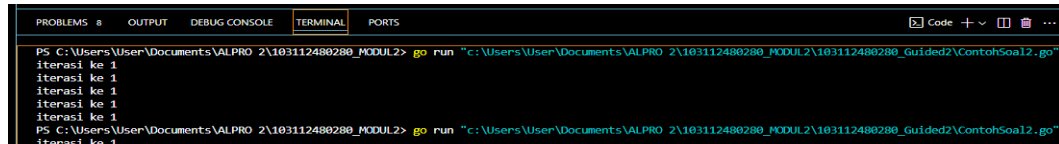
### Contoh Soal 2

```
//Nama : Anggun Wahyu W. (103112480280)
package main

import "fmt"

func main(){
    for i :=1; i<=5; i++){
        fmt.Println("iterasi ke", i)
    }
}
```

Output:

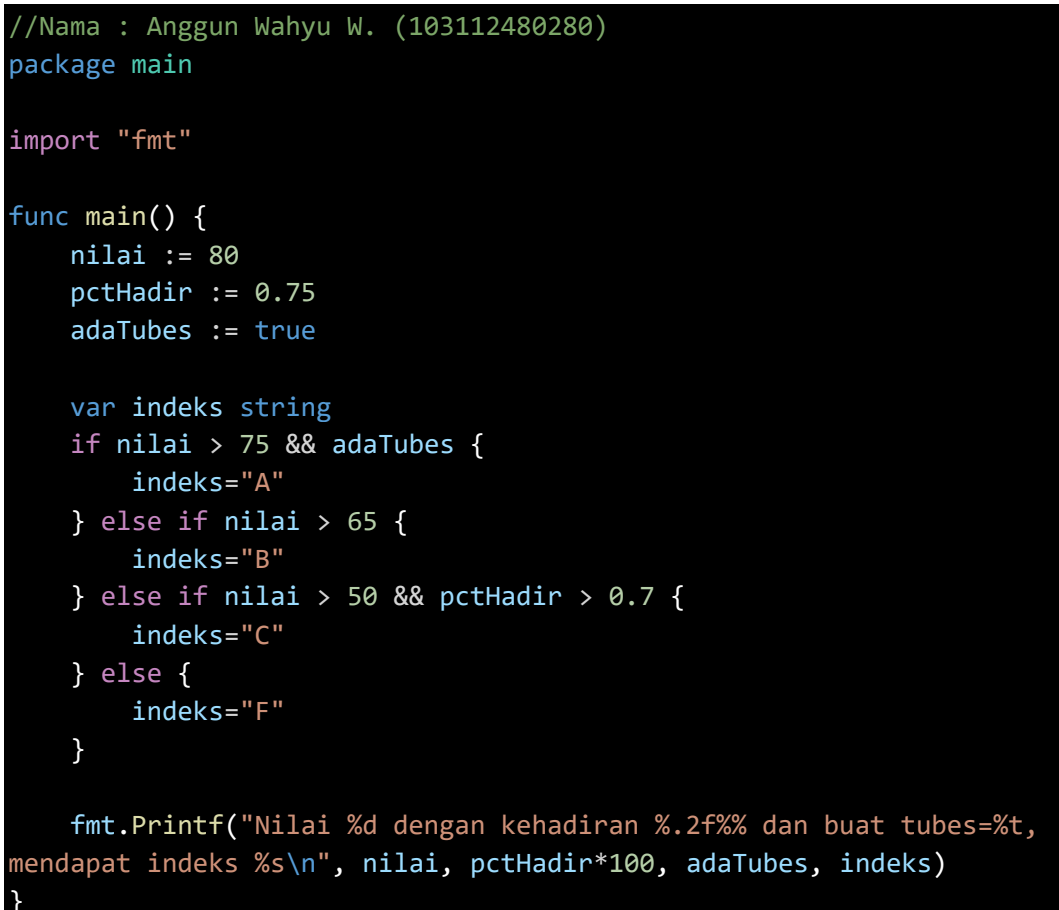


```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_Guided2\ContohSoal2.go"
iterasi ke 1
iterasi ke 1
iterasi ke 1
iterasi ke 1
iterasi ke 1
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_Guided2\ContohSoal2.go"
iterasi ke 1
```

### Deskripsi Program

Program ini mencetak teks "iterasi ke 1" sebanyak 5 kali ke layar. Program menggunakan perulangan for yang berjalan dari 1 hingga 5. Dalam setiap iterasi perulangan, program mencetak teks tersebut, sehingga teks "iterasi ke 1" muncul lima kali berturut-turut. Singkatnya, program ini adalah program perulangan yang menghasilkan output teks yang sama sebanyak lima kali.

### Contoh Soal 3



```
//Nama : Anggun Wahyu W. (103112480280)
package main

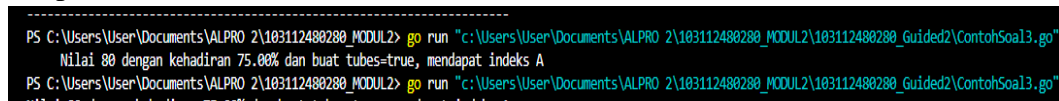
import "fmt"

func main() {
    nilai := 80
    pctHadir := 0.75
    adaTubes := true

    var indeks string
    if nilai > 75 && adaTubes {
        indeks="A"
    } else if nilai > 65 {
        indeks="B"
    } else if nilai > 50 && pctHadir > 0.7 {
        indeks="C"
    } else {
        indeks="F"
    }

    fmt.Printf("Nilai %d dengan kehadiran %.2f%% dan buat tubes=%t,
mendapat indeks %s\n", nilai, pctHadir*100, adaTubes, indeks)
}
```

Output:



```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_Guided2\ContohSoal3.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes=true, mendapat indeks A
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_Guided2\ContohSoal3.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes=true, mendapat indeks A
```

### Deskripsi Program

Program ini menentukan indeks nilai mahasiswa berdasarkan nilai ujian, persentase kehadiran, dan penyelesaian tugas (tubes). Program menetapkan indeks "A" jika

nilai lebih dari 75 dan tugas selesai, "B" jika nilai lebih dari 65, "C" jika nilai lebih dari 50 dan kehadiran lebih dari 70%, dan "F" untuk kondisi lainnya. Program kemudian mencetak informasi lengkap mengenai nilai, kehadiran, status tugas, dan indeks yang diperoleh.



### III. UNGUIDED

#### Latihan Soal 1

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).

(Contoh input/output, Teks bergaris bawah adalah input dari user):

1.	Tahun: <u>2016</u> Kabisat: true
2.	Tahun: <u>2000</u> Kabisat: true
3.	Tahun: <u>2018</u> Kabisat: false

```
//Nama : Anggun Wahyu W. (103112480280)
package main

import "fmt"

func main() {
    var tahun int

    fmt.Print("Tahun: ")
    fmt.Scan(&tahun)

    kabisat := false

    if tahun%400 == 0 {
        kabisat = true
    } else if tahun%100 != 0 && tahun%4 == 0 {
        kabisat = true
    }

    fmt.Println("Kabisat:", kabisat)
}
```

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal1.go"
Tahun: 2016
Kabisat: true
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal1.go"
Tahun: 2000
Kabisat: true
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal1.go"
Tahun: 2018
Kabisat: false
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal1.go"
```

#### Deskripsi Program

Program ini dirancang untuk menentukan apakah suatu tahun adalah tahun kabisat atau bukan, berdasarkan aturan yang berlaku dalam kalender Gregorian.

Pengguna diminta untuk memasukkan tahun sebagai input. Program kemudian memeriksa tahun tersebut dengan menggunakan beberapa kondisi. Jika tahun tersebut habis dibagi 400, maka itu pasti tahun kabisat. Namun, jika tahun tersebut habis dibagi 100 tetapi tidak habis dibagi 400, maka itu bukan tahun kabisat. Selain itu, jika tahun tersebut habis dibagi 4 tetapi tidak habis dibagi 100, maka itu adalah tahun kabisat. Jika tidak memenuhi salah satu dari kondisi di atas, maka tahun tersebut bukan tahun kabisat.

Setelah melakukan pemeriksaan, program akan mencetak hasil berupa true jika tahun tersebut adalah kabisat dan false jika bukan. Dengan demikian, program ini memberikan cara yang sederhana dan efektif untuk mengevaluasi status tahun kabisat berdasarkan input pengguna.

## Latihan Soal 2

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

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

Nilai K = 100  
 Nilai  $f(K)$  = 1.0000061880

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

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

Modifikasi program sebelumnya yang menerima input integer K dan menghitung  $\sqrt{2}$  untuk K tersebut. Hampiran  $\sqrt{2}$  dituliskan dalam ketelitian 10 angka di belakang koma.

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

1.	Nilai K = <u>10</u> Nilai akar 2 = 1.2062058441
2.	Nilai K = <u>100</u> Nilai akar 2 = 1.4133387072
3.	Nilai K = <u>1000</u> Nilai akar 2 = 1.4141252651

```
//Nama : Anggun Wahyu W. (103112480280)
package main
```

```

import "fmt"

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

    FK := func(k int) float64 {
        atas := float64((4*k + 2) * (4*k + 2))
        bawah := float64((4*k + 1) * (4*k + 3))
        return atas / bawah
    }

    akar2 := func(k int) float64 {
        hasil := 1.0
        for i := 0; i <= k; i++ {
            atas := float64((4*i + 2) * (4*i + 2))
            bawah := float64((4*i + 1) * (4*i + 3))
            hasil *= atas / bawah
        }
        return hasil
    }

    fmt.Printf("Nilai f(K) = %.10f\n", FK(k))
    fmt.Printf("Nilai akar 2 = %.10f\n", akar2(k))
}

```

Output:

```

PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal2.go"
Nilai K = 10
Nilai f(K) = 1.0005672150
Nilai akar 2 = 1.4062058441
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal2.go"
Nilai K = 100
Nilai f(K) = 1.0000061880
Nilai akar 2 = 1.4133387072
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal2.go"
Nilai K = 1000
Nilai f(K) = 1.0000000624
Nilai akar 2 = 1.4141252651
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal2.go"
Nilai K =

```

### Deskripsi Program

Program ini berfungsi untuk melakukan dua perhitungan utama berdasarkan input sebuah bilangan bulat, K. Pertama, program menghitung nilai fungsi  $f(K)$  menggunakan rumus matematika tertentu. Kedua, program memberikan aproksimasi nilai akar kuadrat dari 2 ( $\sqrt{2}$ ) berdasarkan nilai K. Pengguna akan diminta untuk memasukkan nilai K, kemudian program akan menghitung dan menampilkan kedua nilai tersebut dengan presisi 10 angka di belakang koma. Program memberikan cara cepat untuk mengevaluasi fungsi  $f(K)$  dan menghampiri nilai akar 2.

### Latihan Soal 3

PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, **buatlah program BlayaPos untuk menghitung blaya pengiriman tersebut dengan ketentuan sebagai berikut!**

Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

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

1.	Contoh #1 Berat parcel (gram): <u>8500</u> Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500
2.	Contoh #2 Berat parcel (gram): <u>9250</u> Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750
3.	Contoh #3 Berat parcel (gram): <u>11750</u> Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000+ Rp. 3750 Total biaya: Rp. 110000

```
//Nama : Anggun Wahyu W. (103112480280)
package main

import "fmt"

func main() {
    var beratGram, biayaTambahan int

    fmt.Print("Berat parcel (gram): ")
    fmt.Scan(&beratGram)

    beratKg := beratGram / 1000
    sisaGram := beratGram % 1000
    biayaKg := beratKg * 10000

    if beratKg > 10 {
        biayaTambahan = 0
    } else if sisaGram >= 500 {
```

```

        biayaTambahan = sisaGram * 5
    } else {
        biayaTambahan = sisaGram * 15
    }

    totalBiaya := biayaKg + biayaTambahan

    fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisaGram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg,
biayaTambahan)
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}

```

Output:

```

PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal3.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal3.go"
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal3.go"
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE2> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE2\103112480280_Unguided2\LatihanSoal3.go"

```

### Deskripsi Program

Program BiayaPos berfungsi untuk menghitung total biaya pengiriman parsel berdasarkan beratnya. Pengguna memasukkan berat parsel dalam gram, dan program akan mengolah data tersebut untuk menghasilkan rincian biaya pengiriman. Program ini akan membagi berat parsel menjadi kilogram (kg) dan sisa gram, kemudian menghitung biaya dasar berdasarkan tarif per kilogram. Biaya tambahan akan dikenakan berdasarkan sisa berat (dalam gram), dengan ketentuan tarif yang berbeda tergantung pada apakah sisa berat tersebut melebihi atau kurang dari 500 gram. Jika total berat parsel melebihi 10 kilogram, biaya tambahan akan digratiskan. Program kemudian menampilkan rincian berat (kg dan gram) dan rincian biaya (biaya dasar, biaya tambahan, dan total biaya) kepada pengguna.

#### **IV. KESIMPULAN**

Struktur kontrol adalah bagian integral dari algoritma pemrograman yang memungkinkan pengendalian aliran eksekusi program berdasarkan kondisi tertentu. Pemahaman yang baik tentang struktur kontrol seperti sekuensial, seleksi, dan perulangan sangat penting untuk mengembangkan algoritma yang efisien dan efektif. Melalui contoh soal dan latihan soal saya diatas, terlihat bahwa struktur kontrol sangat penting dalam menentukan alur logika program.

## **V. REFERENSI**

School of Computing. *Modul Praktikum 2 – Review Struktur Kontrol Algoritma dan Pemrograman 2 SI Informatika*. 2024

Munir, Rinaldi. (2020). *Algoritma dan Pemrograman dalam Bahasa Go*. Bandung: Informatika.

Utami, Ema dan Sukrisno. (2019). *Konsep Dasar Pemrograman Go*. Yogyakarta: Graha Ilmu.