LAPORAN PRAKTIKUM

ALGORITMA DAN PEMROGRAMAN 2

MODUL 2

“STRUKTUR KONTROL”

DISUSUN OLEH:

RAIHAN ADI ARBA

103112400071

S1 IF-12-01

DOSEN:

Dimas Fanny Hebrasianto Permadi, S.ST., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024/2025

DASAR TEORI

Struktur kontrol dalam pemrograman adalah konsep yang memungkinkan program untuk mengatur alur eksekusi kode berdasarkan kondisi atau logika tertentu. Struktur ini terdiri dari dua jenis utama, yaitu percabangan dan perulangan. Percabangan memungkinkan program untuk memilih aksi yang akan dijalankan berdasarkan evaluasi kondisi, sementara perulangan memungkinkan program untuk mengeksekusi blok kode secara berulang selama kondisi tertentu terpenuhi. Keduanya merupakan elemen fundamental dalam pengembangan algoritma dan logika program.

Percabangan sering diimplementasikan menggunakan pernyataan if-else atau switch-case. Jika kondisi yang diberikan bernilai benar (true), blok kode terkait akan dieksekusi. Jika kondisi bernilai salah (false), program akan mengabaikan blok tersebut atau beralih ke blok alternatif (seperti `else` atau `else if`). Dengan ini, program dapat merespons berbagai skenario secara dinamis, seperti validasi input atau pengambilan keputusan berdasarkan hasil perhitungan.

Perulangan, di sisi lain, memungkinkan program untuk mengeksekusi blok kode berulang kali selama kondisi tertentu terpenuhi. Beberapa bentuk perulangan yang umum digunakan adalah `for`, `while`, dan `do-while`. Perulangan ini sangat berguna untuk tugas-tugas seperti pengolahan data dalam array, pencarian elemen, atau penghitungan iterasi. Dengan perulangan, program dapat menghindari penulisan kode yang berulang dan meningkatkan efisiensi eksekusi.

Secara keseluruhan, struktur kontrol memungkinkan program untuk membuat keputusan berbasis kondisi dan mengulangi proses tertentu, sehingga memfasilitasi pengembangan algoritma yang lebih kompleks dan efisien. Kemampuan ini menjadikan struktur kontrol sebagai komponen penting dalam menciptakan aplikasi yang interaktif, adaptif, dan responsif terhadap berbagai kondisi selama runtime. Pemahaman mendalam tentang struktur kontrol merupakan kunci dalam pemrograman dan pengembangan perangkat lunak.

A. GUIDED

1. Source code :

```go
//Nama  : Raihan Adi Arba
//NIM   : 103112400071
//KELAS : IF-12-01

package main

import "fmt"

func main() {
    var greeting string = "Selamat datang di dunia DAP"
    var a, b int
    fmt.Println(greeting)
    fmt.Scanln(&a, &b)
    fmt.Printf("%v + %v = %v\n", a, b, a+b)
}

```

Output :

```
raihan@Raihans-MacBook-Pro minggu 9 % go run p1.go
10
10
raihan@Raihans-MacBook-Pro minggu 9 % go run p1.go
-3
3
raihan@Raihans-MacBook-Pro minggu 9 % go run p1.go
0
0
```

Deskripsi :

Program ini akan menampilkan sebuah pesan sambutan kepada pengguna, kemudian meminta pengguna untuk memasukkan dua bilangan bulat. Setelah kedua bilangan dimasukkan, program akan menjumlahkan kedua bilangan tersebut dan menampilkan hasilnya dalam format yang jelas.Program ini menggunakan fungsi fmt.Println untuk menampilkan pesan sambutan dan menerima input dari pengguna menggunakan fmt.Scanln. Setelah mendapatkan input, program akan menjumlahkan kedua bilangan dan menampilkan hasilnya menggunakan fmt.Printf dengan format yang sesuai. Program ini tidak melakukan validasi input, sehingga pengguna diharapkan memasukkan dua bilangan bulat agar program dapat berjalan dengan benar.

2. Source code :

```go
//Nama  : Raihan Adi Arba
//NIM   : 103112400071
//KELAS : IF-12-01

package main

import "fmt"

func main() {
    for i := 1; i <= 5; i++ {

        fmt.Println("Iterasi ke-", i)
    }
}
```

Output :

```
raihan@Raihans-MacBook-Pro guide % go run 103112400071_guide2.go
Iterasi ke- 1
Iterasi ke- 2
Iterasi ke- 3
Iterasi ke- 4
Iterasi ke- 5
raihan@Raihans-MacBook-Pro guide %
```

Deskripsi :

Program ini akan melakukan perulangan sebanyak lima kali dan menampilkan teks yang menunjukkan nomor iterasi saat ini. Program menggunakan perulangan `for` dengan variabel `i` yang dimulai dari 1 dan akan terus bertambah hingga mencapai 5. Pada setiap iterasi, program akan mencetak teks "Iterasi ke-" diikuti dengan nilai `i`. Struktur perulangan ini memastikan bahwa program menjalankan proses yang sama sebanyak lima kali sebelum berhenti.

3. Source code:

```go
//Nama  : Raihan Adi Arba
//NIM   : 103112400071
//KELAS : IF-12-01

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 :

```
raihan@Raihans-MacBook-Pro guide % go run 103112400071_guide3.go
nilai 80 dengan kehadiran 75.00% dan buat tubes=true, mendapat indeks A
```

Deskripsi :

Program ini menentukan indeks nilai mahasiswa berdasarkan nilai ujian, persentase kehadiran, dan keberadaan tugas besar (tubes).
Program menggunakan beberapa kondisi `if-else` untuk menilai apakah mahasiswa mendafmt.Printfeks A, B, C, atau F berdasarkan aturan berikut:

- Jika nilai lebih dari 75 dan mahasiswa memiliki tugas besar, maka mendapat indeks A.
- Jika nilai lebih dari 65, maka mendapat indeks B.
- Jika nilai lebih dari 50 dan persentase kehadiran lebih dari 70%, maka mendapat indeks C.
- Jika tidak memenuhi syarat di atas, maka mendapat indeks F.

Pada akhir program, hasil evaluasi ditampilkan dalam format yang jelas menggunakan `fmt.Printf`.

B. UNGUIDED

1. Latihan 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: 2016 Kabisat: true
2	Tahun: 2000 Kabisat: true
3	Tahun: 2018 Kabisat: false

Source Code:

```go
//Nama  : Raihan Adi Arba
//NIM   : 103112400071
//KELAS : IF-12-01

package main

import "fmt"

func main() {
    var tahun int
    var ckabisat bool

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

    if tahun%100 == 0 {
        ckabisat = tahun%400 == 0
    } else {
        ckabisat = tahun%4 == 0
    }

    fmt.Printf("Kabisat: %t\n", ckabisat)
}
```

Output :

```
raihan@Raihans-MacBook-Pro unguide % go run 103112400071_unguide1.go
Tahun: 2020
Kabisat: true
raihan@Raihans-MacBook-Pro unguide % go run 103112400071_unguide1.go
Tahun: 2021
Kabisat: false
```

Penjelasan Program:
Program ini meminta pengguna untuk memasukkan sebuah tahun, kemudian program akan melakukan pengecekan apakah tahun tersebut merupakan tahun kabisat atau bukan.
Program menggunakan struktur kondisional if-else untuk menentukan status tahun kabisat berdasarkan aturan berikut:

- Jika tahun yang dimasukkan habis dibagi 100, maka program akan mengecek apakah tahun tersebut juga habis dibagi 400. Jika iya, maka tahun tersebut adalah tahun kabisat.
- Jika tahun tidak habis dibagi 100, program akan mengecek apakah tahun tersebut habis dibagi 4. Jika iya, maka tahun tersebut juga merupakan tahun kabisat.

Program ini memanfaatkan operator modulo (%) untuk melakukan pengecekan terhadap pembagian suatu bilangan dan menyimpan hasil evaluasi dalam variabel ckabisat. Hasil akhir akan ditampilkan dengan format "Kabisat: true" jika tahun tersebut kabisat, dan "Kabisat: false" jika bukan.

2. Latihan 2

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

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 = **10** Nilai akar 2 = 1.4062058441
2	Nilai K = **100** Nilai akar 2 = 1.4133387072
3	Nilai K = **1000** Nilai akar 2 = 1.4141252651

Source Code:

```go
//Nama  : Raihan Adi Arba
//NIM   : 103112400071
//KELAS : IF-12-011

package main

import (
    "fmt"
)

func main() {
    var xy, x, y float64
    var vloop, K int

    fmt.Print("Nilai K = ")
    fmt.Scan(&K)
    xy = 1.0

    for vloop = 0; vloop <= K; vloop++ {
        x = float64((4*vloop + 2) * (4*vloop + 2))
        y = float64((4*vloop + 1) * (4*vloop + 3))
        xy *= x / y
    }

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

Output:

```
raihan@Raihans-MacBook-Pro unguide % go run 103112400071_unguide2.go
Nilai K = 10
Nilai akar 2 = 1.4062058441
```

Deskripsi Program:

Program ini akan meminta pengguna untuk memasukkan nilai K, yang menentukan jumlah iterasi dalam proses aproksimasi akar kuadrat dari 2. Program menggunakan perulangan untuk melakukan perkalian rasio pecahan secara bertahap, di mana hasil akhir dari perhitungan akan semakin mendekati nilai akar 2 seiring bertambahnya jumlah iterasi. Program menginisialisasi nilai awal xy = 1.0 sebagai dasar perkalian, kemudian dalam setiap iterasi, program menghitung pembilang (x) dengan rumus ((4vloop + 2)^2) dan penyebut (y) dengan rumus (4vloop + 1) times (4vloop + 3)). Nilai xy kemudian dikalikan dengan rasio x / y untuk memperbarui hasil aproksimasi pada setiap iterasi. Setelah seluruh iterasi selesai dijalankan, program akan menampilkan hasil aproksimasi akar 2 dengan presisi hingga 10 desimal.

3. Latihan 3

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

Dari berat parsel (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):

```
Contoh #1
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
```

Source Code:

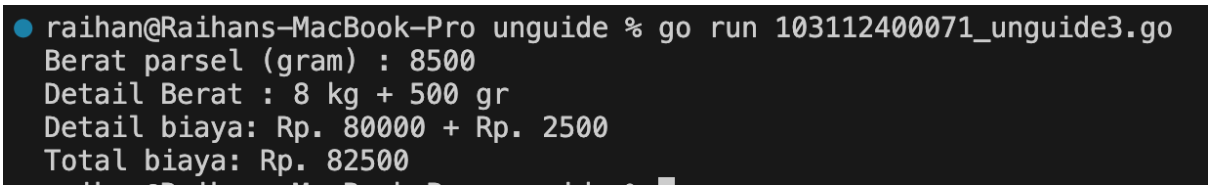
```go
//Nama  : Raihan Adi Arba
//NIM   : 103112400071
//KELAS : IF-12-011

package main

import "fmt"

func main() {
    var berat_parsel, detail_biaya, total_berat, sisa_berat, sisa int
    fmt.Print("Berat parsel (gram) : ")
    fmt.Scan(&berat_parsel)
    total_berat = berat_parsel / 1000
    sisa_berat = berat_parsel % 1000
    fmt.Printf("Detail Berat : %d kg + %d gr\n", total_berat, sisa_berat)
    detail_biaya = total_berat * 10000
    if sisa_berat >= 500 {
        sisa = sisa_berat * 5
    } else if sisa_berat < 500 {
        sisa = sisa_berat * 15
    } else if sisa_berat > 10 {
        sisa = 0
    }
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d \n", detail_biaya, sisa)
    fmt.Printf("Total biaya: Rp. %d \n", detail_biaya+sisa)
}
```

Output:



Deskripsi Program:
Program ini meminta pengguna untuk memasukkan berat parsel dalam satuan gram, kemudian program akan menghitung detail berat dalam satuan kilogram dan gram, serta menentukan biaya pengiriman berdasarkan berat tersebut. Program pertama-tama menghitung berat total dalam kilogram dengan membagi berat parsel dengan 1000, sedangkan sisa berat dalam gram diperoleh dengan operasi modulo (%). Setelah itu, program menampilkan detail berat dalam format kg + gr. Selanjutnya, program menghitung biaya pengiriman. Biaya dasar dihitung dengan mengalikan jumlah kilogram dengan Rp 10.000 per kg. Untuk sisa berat yang kurang dari 500 gram, biaya tambahan dihitung dengan mengalikan sisa berat dengan Rp 15 per gram, sedangkan jika sisa berat 500 gram atau lebih, biaya tambahan dihitung dengan mengalikan sisa berat dengan Rp 5 per gram. Program kemudian mencetak rincian biaya dan total biaya pengiriman berdasarkan perhitungan tersebut.

DAFTAR PUSTAKA

Prayogo, N. A. (2021). ***Dasar Pemrograman Go. Ebook***