

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



Oleh :

NAMA : Felix Pedrosa Valentino

NIM : 103112400056

KELAS : IF – 12 – 01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## I. DASAR TEORI

### 1. Struktur Program Go

Dalam kerangka sebuah program yang ditulis menggunakan bahasa pemrograman Go, terdapat dua komponen utama yang selalu ada:

- package main: Menandakan bahwa file ini merupakan program utama.
- func main(): Berisi kode yang menjadi inti dari program Go tersebut.

Komentar dalam program, meskipun bukan bagian dari kode itu sendiri, dapat ditambahkan di mana saja. Komentar ini dapat ditulis dalam dua format :

- Satu baris komentar yang diawali dengan dua garis miring (//) hingga akhir baris.
- Beberapa baris komentar yang dimulai dengan pasangan karakter `/*` dan diakhiri dengan `*/`.

### 1) Koding, Kompilasi, dan Eksekusi Go

#### a. Koding

Dalam penulisan program menggunakan bahasa Go, prinsip dasar yang diikuti tidak jauh berbeda dengan bahasa pemrograman lain. Program Go harus ditulis menggunakan penyunting teks dan disimpan dalam format teks, bukan dalam format dokumen seperti .doc atau .docx.

Setiap program Go disimpan dalam file dengan ekstensi \*.go, dan nama file tersebut sebaiknya mencerminkan nama program yang bersangkutan. Selain itu, penting untuk menyimpan setiap program Go dalam folder tersendiri, di mana nama folder harus sama dengan nama program. Dengan begitu, secara prinsip, sebuah program Go dapat dibagi menjadi beberapa file dengan ekstensi \*.go, asalkan semuanya disimpan dalam folder yang sama.

#### b. Kompilasi

Beberapa bahasa pemrograman dirancang untuk diimplementasikan sebagai interpreter, sementara yang lainnya sebagai kompilator.

Interpreter akan membaca setiap baris instruksi dan langsung mengeksekusinya, dengan minimal pemeriksaan terhadap kesalahan penulisan program secara keseluruhan. Di sisi lain, kompilator akan memeriksa seluruh program sumber terlebih dahulu dan mengubahnya menjadi program eksekutabel, sehingga konsistensi penulisan, seperti penggunaan tipe data, sudah terverifikasi sebelum eksekusi dilakukan. Selain itu, dengan menjadikan program sebagai eksekutabel terlebih dahulu, proses optimasi dapat dilaksanakan, sehingga program dapat berjalan dengan sangat efisien.

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

1. Buka shell atau terminal (seperti cmd. exe di Windows).
2. Masuk ke dalam folder program (biasanya terletak di C:\Users\go\src\ atau lokasi yang serupa).
3. Ketik perintah ``go build`` atau ``go build file. go`` untuk mengkompilasi file. go.
4. Jika terjadi kesalahan, akan muncul pesan error yang sesuai. Pelajari dengan saksama pesan tersebut, perbaiki kesalahan pada program sumber, lalu ulangi proses kompilasi.
5. Jika berhasil, program dengan nama yang sama akan dihasilkan di folder tersebut, dengan ekstensi . exe (untuk Windows).
6. Jalankan program eksekutabel tersebut dari terminal yang sama. Hindari menjalankannya dengan mengklik ikon program dari folder, karena program ini berbasis teks dan belum dirancang untuk tampilan Windows.

## 2. Tipe Data & Instruksi Dasar

Variabel adalah sebuah nama yang merujuk pada lokasi di dalam memori, di mana data dengan tipe tertentu dapat disimpan. Nama variabel diawali dengan huruf dan dapat diikuti oleh kombinasi jumlah huruf, angka, atau garis bawah. Tipe data yang umum digunakan meliputi integer, real, boolean, karakter, dan string.

Nilai data yang disimpan dalam variabel dapat diakses dengan merujuk langsung kepada nama variabel tersebut. Untuk mendapatkan informasi mengenai alamat atau lokasi dari variabel, kita dapat menambahkan prefiks "&" di depan nama variabel. Jika variabel tersebut menyimpan alamat memori, penggunaan prefiks "\*" pada variabel tersebut akan mengembalikan nilai yang disimpan di lokasi memori yang diacu.

Bahasa Go menerapkan aturan ketat mengenai kesesuaian tipe data. Tipe data yang berbeda tidak boleh dicampur dalam satu ekspresi, bahkan jika keduanya termasuk dalam kategori yang sama, seperti dua tipe integer (misalnya `int` dan `int32`). Selain itu, variabel harus dideklarasikan terlebih dahulu sebelum digunakan.

## II. GUIDED

### 1. UnGuided 1

Source Code :

```
// Felix Pedrosa V

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 D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week2_modul2\unguided1\unguided1.go"
Selamat datang di dunia DAP
9+9
9 + 9 = 18
```

Penjelasan Program :

Program di atas merupakan sebuah program sederhana yang ditulis menggunakan bahasa pemrograman Go. Program ini diawali dengan mendeklarasikan sebuah variabel bernama `greetings`, yang berisi pesan sambutan "Selamat datang di dunia DAP". Selanjutnya, program mendeklarasikan dua variabel bertipe integer, yaitu `a` dan `b`, yang akan digunakan untuk menyimpan input dari pengguna.

Setelah mencetak pesan sambutan ke layar menggunakan `fmt. Println()`, program siap menerima input dari pengguna. Pengguna diminta untuk memasukkan dua angka yang akan disimpan dalam variabel `a` dan `b` menggunakan fungsi `fmt. Scanln()`. Setelah kedua angka tersebut dimasukkan, program akan menghitung jumlahnya dan menampilkannya dalam format yang terstruktur menggunakan `fmt. Printf()`, yang menunjukkan hasil penjumlahan dalam bentuk "a + b = hasil".

### 2. UnGuided 2

Source Code :

```
// Felix Pedrosa V
```

```
package main

import "fmt"

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

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week2\unguided2\unguided2.go"
Iterasi ke- 1
Iterasi ke- 2
Iterasi ke- 3
Iterasi ke- 4
Iterasi ke- 5
```

Penjelasan Program :

Program di atas adalah sebuah program sederhana yang ditulis menggunakan bahasa pemrograman Go. Program ini memanfaatkan struktur kontrol for untuk melakukan iterasi pada angka 1 hingga 5. Di dalam blok for, variabel i diinisialisasi dengan nilai 1 dan akan bertambah satu pada setiap iterasi hingga mencapai angka 5. Pada setiap iterasi, program mencetak pesan ke layar yang menunjukkan nomor iterasi dalam format "Iterasi ke- X", di mana X merupakan nilai dari variabel i.

### 3. UnGuided 3

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"
func main() {
    // Contoh nilai (bisa diganti sesuai kebutuhan)
    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\n", nilai, pctHadir*100, adaTubes, indeks)
}

```

#### Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week2_modul2\unguided3\unguided3.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes = true memiliki indeks A

```

#### Penjelasan Program :

Program di atas adalah program sederhana yang ditulis menggunakan bahasa pemrograman Go. Program ini bertujuan untuk menentukan indeks nilai mahasiswa berdasarkan berbagai kriteria yang ditetapkan. Dalam fungsi main, program mendeklarasikan beberapa variabel, yaitu nilai, pctHadir, dan adaTubes. Masing-masing variabel ini merepresentasikan nilai akademik, persentase kehadiran, dan status keberadaan tugas tambahan (tubes) bagi mahasiswa.

Program ini kemudian menggunakan struktur kontrol if-else untuk menentukan indeks berdasarkan nilai dan kondisi yang berlaku. Apabila nilai mahasiswa lebih dari 75 dan dia memiliki tugas tambahan, maka indeks yang diberikan adalah "A". Jika nilai berada antara 65 hingga 75, indeksnya adalah "B". Selanjutnya, jika nilai lebih dari 50 dan persentase kehadirannya melebihi 70%, indeks yang diberikan adalah "C". Jika tidak ada kondisi yang terpenuhi, maka indeksnya akan menjadi "F".

Setelah menentukan indeks, program ini mencetak hasilnya dengan format yang jelas, menampilkan nilai, persentase kehadiran, status tugas tambahan, dan indeks yang diperoleh oleh mahasiswa.

### III. UNGUIDED

#### 1. Guided 1

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Tahun: ")
    fmt.Scan(&tahun)

    if tahun%400 == 0 {
        fmt.Printf("Kabisat: true\n")
    } else if tahun%100 == 0 {
        fmt.Printf("Kabisat: false\n")
    } else if tahun%4 == 0 {
        fmt.Printf("Kabisat: true\n")
    } else {
        fmt.Printf("Kabisat: false\n")
    }
}
```

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week2_modul2\guided1\guided1.go"
Tahun: 2016
Kabisat: true
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week2_modul2\guided1\guided1.go"
Tahun: 2000
Kabisat: true
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week2_modul2\guided1\guided1.go"
Tahun: 2018
Kabisat: false
```

Penjelasan Program :

Program di atas adalah sebuah program sederhana yang ditulis dalam bahasa pemrograman Go, yang bertujuan untuk menentukan apakah suatu tahun termasuk tahun kabisat atau bukan. Di dalam fungsi main, program akan meminta pengguna untuk memasukkan tahun. Setelah menerima input tersebut, program menggunakan struktur kontrol if-else untuk mengevaluasi kondisi tahun yang dimasukkan.



Jika tahun yang dimasukkan dapat dibagi habis oleh 400, maka program akan mencetak "Kabisat: true", yang berarti tahun tersebut adalah tahun kabisat. Jika tahun dapat dibagi habis oleh 100 tetapi tidak oleh 400, program akan menampilkan "Kabisat: false". Selanjutnya, jika tahun dapat dibagi habis oleh 4 tetapi tidak oleh 100, program akan mencetak "Kabisat: true" lagi. Apabila tidak ada kondisi yang terpenuhi, program akan mencetak "Kabisat: false".

## 2. Guided 2

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

func main() {
    var nilai float64

    for {
        fmt.Print("Nilai K = ")
        _, err := fmt.Scanln(&nilai)
        if err == nil {
            break
        }
    }

    f_k := (4*nilai + 2) * (4*nilai + 2) / ((4*nilai + 1) * (4*nilai + 3))

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

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week2_modul2\guided2\guided2.go"
Nilai K = 100
Nilai f(k) = 1.0000061880
```

Penjelasan Program :

Program di atas adalah sebuah program sederhana yang ditulis dalam bahasa pemrograman Go, yang bertujuan untuk meminta pengguna untuk memasukkan nilai K, lalu menghitung dan menampilkan hasil dari fungsi

$f(K)$  sesuai dengan rumus yang telah ditentukan. Proses dimulai dengan mendeklarasikan variabel `nilaiK` yang bertipe float64 untuk menyimpan input dari pengguna. Selanjutnya, terdapat sebuah loop tak terbatas yang berfungsi untuk terus meminta input hingga pengguna memberikan nilai yang valid. Dalam loop tersebut, program membaca input menggunakan fmt. Scanln(&nilaiK) dan jika tidak ada kesalahan (err == nil), loop akan berhenti.

Setelah memperoleh nilai K, program memproses dan menghitung nilai  $f(K)$  berdasarkan rumus matematika yang telah diberikan. Hasil perhitungan ditampilkan dengan ketelitian 10 angka desimal menggunakan `fmt. Printf("Nilai  $f(K)$  = %. 10f\n", f\_k). Dengan cara ini, program memastikan bahwa hanya input yang valid yang diproses dan hasil yang ditampilkan memiliki presisi yang sangat tinggi.

### 3. Guided 3

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

func main() {

    var berat, beratKg, sisaGram, biayaKg, biayaSisaGram, totalBiaya int

    fmt.Print("Masukkan berat parsel (gram): ")
    fmt.Scan(&berat)

    beratKg = berat / 1000
    sisaGram = berat % 1000

    biayaKg = beratKg * 10000

    if sisaGram > 0 {
        if beratKg > 10 {
            biayaSisaGram = 0
        } else if sisaGram >= 500 {
            biayaSisaGram = sisaGram * 5
        } else {
            biayaSisaGram = sisaGram * 15
        }
    }
}
```

```

    }
}

totalBiaya = biayaKg + biayaSisaGram

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

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\alpro2_week2_modul2\guided3\guided3.go"
Masukkan berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\alpro2_week2_modul2\guided3\guided3.go"
Masukkan berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\alpro2_week2_modul2\guided3\guided3.go"
Masukkan berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000

```

Penjelasan Program :

Program di atas adalah program untuk menghitung biaya pengiriman parcel berdasarkan berat yang dimasukkan oleh pengguna dalam satuan gram. Program dimulai dengan mendeklarasikan variabel untuk menyimpan berat parcel, berat dalam kilogram, sisa gram, biaya per kilogram, biaya untuk sisa gram, dan total biaya bertipe integer. Kemudian meminta pengguna untuk memasukkan berat parcel dalam satuan gram. Setelah meminta pengguna untuk memasukkan berat parcel, program kemudian mengonversi berat tersebut dari gram ke kilogram dan menghitung sisa gram yang tidak terpakai. Biaya pengiriman dihitung berdasarkan berat dalam kilogram, di mana setiap kilogram dikenakan biaya Rp. 10.000. Untuk sisa gram, program menerapkan tarif yang berbeda: jika berat kilogram lebih dari 10, biaya untuk sisa gram adalah nol; jika sisa gram lebih dari atau sama dengan 500, dikenakan biaya Rp. 5 per gram; dan jika kurang dari 500 gram, dikenakan biaya Rp. 15 per gram. Setelah semua perhitungan dilakukan, program mencetak detail berat dalam kilogram dan gram, rincian biaya untuk setiap kategori, serta total biaya pengiriman.

#### **IV. KESIMPULAN**

Sebagai kesimpulan, bahasa pemrograman Go adalah pilihan yang efisien dan sederhana dengan struktur yang jelas. Setiap program dalam Go wajib memiliki paket utama (`package main`) dan fungsi utama (`func main()`) yang berfungsi sebagai titik awal eksekusi. Dalam penerapannya, bahasa Go memanfaatkan mekanisme kompilasi untuk memastikan bahwa kode yang dijalankan bebas dari kesalahan sintaksis. Tipe data dalam Go sangat presisi, sehingga setiap variabel harus dideklarasikan dengan tipe yang sesuai sebelum digunakan. Selain itu, kontrol alur seperti perulangan (`for`) dan percabangan (`if-else`) memungkinkan program untuk menjalankan instruksi berulang kali atau membuat keputusan berdasarkan kondisi tertentu. Dari berbagai program yang telah diuji, dapat disimpulkan bahwa Go sangat cocok untuk berbagai keperluan pemrograman, termasuk perhitungan matematis dan pengembangan aplikasi yang efisien dan terstruktur.

## **V. REFERENSI**

Modul 2 - Praktikum Alpro 2

Dibantu oleh AI