

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 1 & 2
REVIEW STRUKTUR KONTROL**



**DISUSUN OLEH:
ANASTASIA ADINDA NARENDRA INDRIANTO
103112400085
S1 IF-12-01**

**DOSEN:
Dimas Fanny Hebrasianto Permadi, S.ST., M.KOM**

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

1. Pengertian Bahasa Pemrograman Golang

Golang adalah bahasa pemrograman yang memiliki sejumlah kelebihan yang tidak dimiliki bahasa pemrograman yang lainnya. Hadirnya bahasa pemrograman Go Language (Golang) semakin dirasakan oleh para pengembang. Tidak heran jika banyak orang yang mulai belajar bahasa pemrograman yang satu ini.

Golang merupakan bahasa pemrograman yang dibuat Google dan tujuannya untuk menyempurnakan bahasa pemrograman yang ada, seperti C, Python dan yang lainnya. Golang bisa jadi pilihan yang tepat saat membuat aplikasi baru.

2. Pengertian Input dan Output

- i. *Input* atau masukan adalah data yang diberikan ke dalam program. Masukan ini bisa berasal dari berbagai sumber, seperti pengguna melalui keyboard, mouse, atau suara, dan juga bisa berasal dari sensor atau perangkat lain yang terhubung ke komputer. Dalam pemrograman, input diperlakukan sebagai bahan baku yang akan diproses oleh program.
- ii. *Output* atau keluaran adalah hasil yang diproduksi oleh program setelah mengolah input. Output bisa berbentuk tampilan pada layar, cetakan pada printer, suara, ataupun penyimpanan data ke file. Inti dari program yang kita tulis sebenarnya adalah menghasilkan output yang bermakna dari input yang diberikan.

3. Pengertian Tipe Data

Tipe data adalah kategori data yang akan diproses oleh sebuah program komputer. Dengan tipe data, program dapat mengetahui cara menangani data dengan benar, seperti melakukan operasi matematika pada angka atau memanipulasi string untuk tujuan tertentu. Secara sederhana, pengertian tipe data adalah kategori data yang akan diproses oleh sebuah program komputer. Dengan tipe data, program dapat mengetahui cara menangani data dengan benar, seperti melakukan operasi matematika pada angka atau memanipulasi string untuk tujuan tertentu. Contoh paling sederhana dari tipe data adalah tipe data **integer** yang digunakan untuk menyimpan angka bulat atau tipe data **string** yang digunakan untuk menyimpan rangkaian karakter. Penggunaan tipe data yang benar dalam suatu program memastikan bahwa data diolah secara tepat dan mengurangi risiko kesalahan atau bug dalam program tersebut.

4. Fungsi Tipe Data

- a. **Menentukan Jenis Nilai:** Tipe data memberi tahu program jenis nilai yang akan disimpan dalam variabel. Misalnya, jika kita ingin menyimpan angka, kita menggunakan tipe data integer atau float, sedangkan untuk menyimpan teks, kita menggunakan tipe data string. Ini membantu program memahami bagaimana cara menangani data tersebut.
- b. **Efisiensi Penggunaan Memori:** Setiap tipe data memerlukan jumlah memori yang berbeda. Jika kita memilih tipe data yang tepat, program bisa menggunakan memori lebih efisien. Misalnya, menggunakan tipe data yang lebih kecil untuk angka yang tidak terlalu besar akan menghemat ruang di memori.
- c. **Menjamin Konsistensi Data:** Dengan menentukan tipe data, kita memastikan bahwa variabel hanya bisa menyimpan jenis nilai yang sesuai. Misalnya, jika kita mendeklarasikan variabel sebagai tipe integer, program tidak akan bisa memasukkan teks atau jenis data lainnya ke dalamnya. Ini membantu menjaga agar data tetap konsisten dan sesuai dengan yang diharapkan.
- d. **Memudahkan Operasi pada Data:** Tipe data juga menentukan operasi apa saja yang bisa dilakukan pada data. Misalnya, kita bisa melakukan perhitungan

matematika pada angka, tetapi kita tidak bisa melakukan hal yang sama pada teks. Jadi, dengan menentukan tipe data yang tepat, kita bisa melakukan operasi yang sesuai dengan jenis data yang kita miliki.

5. Pengertian If-Else dan Fungsinya

If-else adalah struktur percabangan dalam pemrograman yang digunakan untuk mengeksekusi kode berdasarkan suatu kondisi. Jika kondisi dalam if bernilai true, maka blok kode di dalamnya akan dijalankan. Jika tidak, program akan memeriksa kondisi dalam else if sebagai alternatif. Jika semua kondisi false, maka else akan dijalankan sebagai pilihan terakhir. Dengan menggunakan if-else, program dapat mengambil keputusan dan menjalankan instruksi yang sesuai berdasarkan kondisi yang diberikan.

If-else berfungsi untuk mengontrol alur program dengan mengeksekusi kode berdasarkan suatu kondisi. If digunakan untuk memeriksa apakah suatu kondisi bernilai true, jika ya, maka blok kode di dalamnya akan dijalankan. Jika tidak, program dapat menggunakan else-if untuk mengevaluasi beberapa kondisi tambahan secara berurutan. Jika semua kondisi false, maka else akan dieksekusi sebagai pilihan terakhir. Selain itu, terdapat if-else bersarang, yang memungkinkan pengecekan kondisi di dalam kondisi lain untuk menangani keputusan yang lebih kompleks.

6. Pengertian While Loop dan Fungsinya

While loop adalah metode perulangan yang mengeksekusi blok kode selama kondisi yang diberikan bernilai true dan akan berhenti ketika kondisi berubah menjadi false. Perulangan ini sangat berguna dalam kasus di mana jumlah iterasi belum diketahui secara pasti, seperti membaca input hingga valid atau menjalankan suatu proses hingga syarat tertentu terpenuhi. Fungsinya adalah;

- a. **Menjalankan Perulangan Berdasarkan Kondisi** – While loop memastikan suatu proses terus berjalan selama kondisi bernilai **true**.
- b. **Mengatasi Iterasi yang Tidak Diketahui Jumlahnya** – Digunakan ketika jumlah perulangan tidak bisa ditentukan sejak awal, seperti menunggu input yang valid.
- c. **Mengoptimalkan Kontrol Program** – Membantu mengelola eksekusi kode agar hanya berjalan saat kondisi tertentu terpenuhi, meningkatkan efisiensi program.

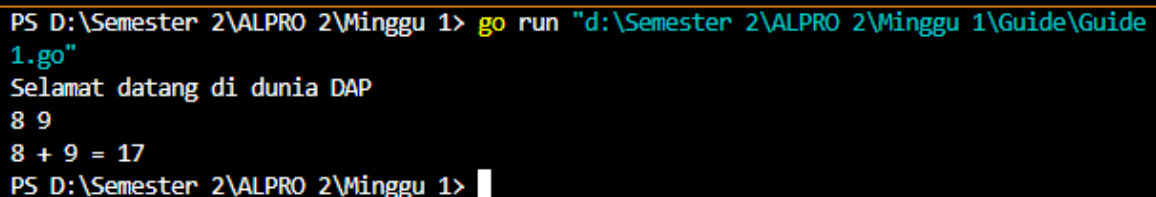
II. GUIDED

1. Guide 1

Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085, S1IF-12-01
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:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Guide\Guide
1.go"
Selamat datang di dunia DAP
8 9
8 + 9 = 17
PS D:\Semester 2\ALPRO 2\Minggu 1>
```

Deskripsi Program:

Program Guide1.go dibuat dengan bahasa pemrograman Go tujuan untuk menampilkan output berupa kalimat “Selamat datang di dunia DAP” dari variabel greetings, lalu dilanjutkan dengan memasukan input a dan b yang berupa angka karena tipe data integer, yang berarti ada 2 angka inputan yang harus dimasukan oleh pengguna dibagian terminal dan menggunakan fmt.Scanln (yang berarti mencetak kebawah). Hasil akhir program adalah hasil pertambahan dari 2 angka inputan dari pengguna menggunakan fmt.Printf yang berarti terdapat satu atau beberapa kalimat yang ada didalam hasil outputan tersebut seperti pada hasil output pada gambar diatas.

2. Guide 2

Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085, S1IF-12-01
package main
import "fmt"
func main(){
    for i := 1; i <= 4; i++ {
        fmt.Println("Iterasi ke-", i)
    }
}
```

Output:

```
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Guide\Guide
1.go"
Iterasi ke- 1
Iterasi ke- 2
Iterasi ke- 3
Iterasi ke- 4
PS D:\Semester 2\ALPRO 2\Minggu 1> █
```

Deskripsi Program:

Program Guide1.go dibuat dengan bahasa pemrograman Go dengan tujuan perulangan dari sebuah iterasi, yang mencetak teks "Iterasi ke-" diikuti dengan nomor iterasi, sebanyak 4 kali. Program ini menggunakan perulangan for dengan variabel i yang dimulai dari 1 dan berakhir pada 4. Setiap iterasi, nilai i dicetak menggunakan `fmt.Println()` yang berarti hasil output tersebut mencetak turun kebawah.

3. Guide 3

Source Code:

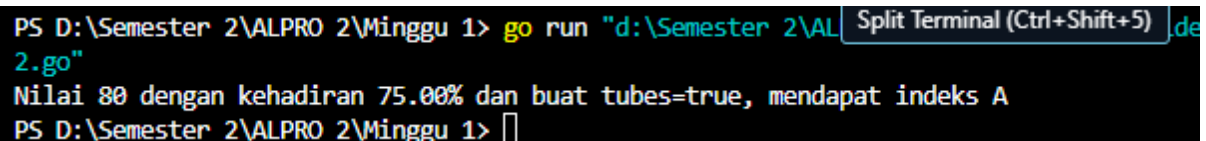
```
// Anastasia Adinda Narendra Indrianto
// 103112400085, S1IF-12-01
package main
import "fmt"
func main() {
    nilai := 80
    pctHadir := 0.75
    adaTubes := true

    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:



```
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Guide2.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes=true, mendapat indeks A
PS D:\Semester 2\ALPRO 2\Minggu 1> 
```

Deskripsi Program:

Program Guide2.go dibuat dengan bahasa pemrograman Go yang dibuat dengan tujuan menentukan indeks nilai berdasarkan kriteria tertentu. Variabel yang digunakan adalah nilai, pctHadir (persentase kehadiran), dan adaTubes (status apakah ada tugas besar). Jika nilai lebih dari 75 dan ada tugas besar (adaTubes bernilai true), indeksnya adalah "A", jika nilai lebih dari 65, indeksnya adalah "B", jika nilai lebih dari 50 dan persentase kehadiran lebih dari 70%, indeksnya adalah "C", jika tidak memenuhi kriteria di atas, indeksnya adalah "F". Program kemudian mencetak hasilnya dalam format yang jelas, menampilkan nilai, persentase kehadiran, status tugas besar, dan indeks yang didapat.

III. UNGUIDED

1. Soal 2A No.2

Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085, S1IF-12-01
package main

import "fmt"

func kabisat(tahun int) bool {
    if tahun%400 == 0 {
        return true
    } else if tahun%100 == 0 {
        return false
    } else if tahun%4 == 0 {
        return true
    }
    return false
}

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

    if kabisat(tahun) {
        fmt.Println("Kabisat : true")
    } else {
        fmt.Println("Kabisat : false")
    }
}
```

Output:

```
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Ano2.go"
Masukan Tahun: 2016
Kabisat : true
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Ano2.go"
Masukan Tahun: 2000
Kabisat : true
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Ano2.go"
Masukan Tahun: 2018
Kabisat : false
PS D:\Semester 2\ALPRO 2\Minggu 1> |
```

Deskripsi Program:

Program 2Ano2.go dibuat dengan tujuan menentukan tahun yang ingin dimasukan pengguna apakah tahun kabisat atau bukan. Jika tahun tersebut habis dibagi 400 dan habis dibagi 4 maka tahun tersebut merupakan tahun kabisat yang hasilnya akan mengeluarkan output true, namun jika tahun tersebut tidak habis dibagi 400 dan tidak habis dibagi 4 maka dan tahun tersebut habis dibagi 100 maka tahun tersebut bukanlah tahun kabisat yang akan mengeluarkan hasil output false. Secara singkat program ini menentukan apakah suatu tahun merupakan tahun kabisat dengan menggunakan fungsi kabisat(tahun int) bool. Fungsi ini mengevaluasi tahun berdasarkan aturan bahwa tahun yang habis dibagi 400 adalah kabisat, tahun yang habis dibagi 100 tetapi tidak habis dibagi 400 bukan kabisat, dan tahun yang habis dibagi 4 adalah kabisat.

2. Soal 2B No.4

Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085, S1IF-12-01
package main

import "fmt"

func hitungFk(K int) float64 {
    hasil := 1.0
    for i := 0; i < K; i++ {
        penyebut := (4*float64(i) + 2) * (4*float64(i) + 2)
        pembilang := (4*float64(i) + 1) * (4*float64(i) + 3)

        hasil *= float64(penyebut) / float64(pembilang)
    }
    return hasil
}

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

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


Output:

```
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Bno4.go"
Nilai K = 10
Nilai akar 2 = 1.4054086752
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Bno4.go"
Nilai K = 100
Nilai akar 2 = 1.4133299615
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Bno4.go"
Nilai K = 1000
Nilai akar 2 = 1.4141251768
PS D:\Semester 2\ALPRO 2\Minggu 1> █
```

Deskripsi Program:

Program 2Bno4.go ini dibuat dengan tujuan menghitung nilai pendekatan akar 2 menggunakan metode iteratif berdasarkan jumlah iterasi K yang dimasukkan oleh pengguna. Fungsi `hitungFk(K int) float64` melakukan perhitungan dengan mengalikan hasil sebelumnya dengan rasio antara penyebut $(4i + 2)^2$ dan pembilang $(4i + 1)(4i + 3)$ dalam perulangan `for` sebanyak K kali. Dalam fungsi `main()`, pengguna diminta memasukkan nilai K, lalu program memanggil fungsi `hitungFk()` untuk menghitung nilai pendekatan akar 2, yang kemudian ditampilkan dengan ketelitian 10 angka di belakang koma.

3. Soal 2C No 1

Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085, S1IF-12-01
package main

import (
    "fmt"
)

func hitungBiaya(beratGram int) (int, int, int, int, int) {
    kg := beratGram / 1000
    sisaGram := beratGram % 1000

    var biayaKg, biayaSisa int

    // Hitung biaya per kg
    if kg > 0 {
        biayaKg = kg * 10000
    } else {
        biayaKg = 0
    }

    // Hitung biaya tambahan untuk sisa gram
```

```

    if sisaGram > 0 {
        biayaSisa = (sisaGram / 100) * 500
    } else {
        biayaSisa = 0
    }

    // Total biaya
    totalBiaya := biayaKg + biayaSisa
    return kg, sisaGram, biayaKg, biayaSisa, totalBiaya
}

func main() {
    var berat int
    fmt.Print("Masukkan berat parcel (gram): ")
    fmt.Scan(&berat)

    if berat < 0 {
        fmt.Println("Berat tidak boleh negatif!")
    } else {
        kg, sisaGram, biayaKg, biayaSisa, totalBiaya := hitungBiaya(berat)

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

```

Output:

```

PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Cno1.go"
Masukkan berat parcel (gram): 8500
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Cno1.go"
Masukkan berat parcel (gram): 9250
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 1000
Total biaya: Rp. 91000
PS D:\Semester 2\ALPRO 2\Minggu 1> go run "d:\Semester 2\ALPRO 2\Minggu 1\Tugas LAPRAK\2Cno1.go"
Masukkan berat parcel (gram): 11750
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3500
Total biaya: Rp. 113500
PS D:\Semester 2\ALPRO 2\Minggu 1>

```

Deskripsi Program:

Program 2Cno1.go dibuat dengan tujuan menghitung biaya pengiriman parcel berdasarkan berat yang dimasukkan pengguna dalam satuan **gram**. Perhitungan dilakukan melalui fungsi `hitungBiaya(beratGram int)`, yang memproses berat parcel menjadi **jumlah kilogram (kg) dan sisa gram**, serta menentukan biaya pengiriman sesuai tarif yang telah ditetapkan. Biaya dasar pengiriman adalah **Rp 10.000 per kg**, sementara jika terdapat **sisa gram** yang tidak mencapai 1 kg, maka akan dikenakan biaya tambahan sebesar **Rp 500 per 100 gram**.

Di dalam fungsi `main()`, program meminta pengguna untuk memasukkan berat parcel, kemudian memeriksa apakah nilai yang dimasukkan valid (tidak negatif). Jika input valid, program akan memanggil fungsi `hitungBiaya()` untuk menghitung rincian berat dan biaya. Hasil perhitungan kemudian ditampilkan kepada pengguna, mencakup **berat total dalam kg dan gram, biaya per kg, biaya tambahan berdasarkan sisa gram, serta total biaya pengiriman**. Jika pengguna memasukkan berat negatif, program akan menampilkan pesan kesalahan dan tidak melakukan perhitungan. Dengan pendekatan ini, program memastikan bahwa perhitungan biaya pengiriman dilakukan dengan akurat dan informatif bagi pengguna.

IV. KESIMPULAN

Program Guided dan Unguided yang sudah dibuat sebelumnya bisa disimpulkan sebagai berikut secara berturut;

i. Guided

1. Program pertama menampilkan pesan dan menerima dua input angka, lalu menghitung serta menampilkan hasil penjumlahan.
2. Program kedua menggunakan perulangan for untuk mencetak teks "Iterasi ke-" sebanyak 4 kali.
3. Program ketiga menentukan indeks nilai berdasarkan nilai, kehadiran, dan tugas besar menggunakan if-else.

Semua program menggunakan bahasa Go dengan konsep dasar seperti input-output, perulangan, dan percabangan.

ii. Unguided

1. Program Penentuan Tahun Kabisat (2Ano2.go)
 - o Menentukan apakah suatu tahun merupakan tahun kabisat atau bukan.
 - o Menggunakan fungsi kabisat(tahun int) bool untuk mengevaluasi tahun berdasarkan aturan pembagian 400, 100, dan 4.
 - o Menghasilkan output true jika tahun kabisat, dan false jika bukan.
2. Program Perhitungan Pendekatan Akar 2 (2Bno4.go)
 - o Menghitung nilai pendekatan akar 2 menggunakan metode iteratif.
 - o Menggunakan fungsi hitungFk(K int) float64 dengan perulangan for sebanyak K kali.
 - o Hasil akhir ditampilkan dengan ketelitian 10 angka di belakang koma.
3. Program Perhitungan Biaya Pengiriman Parsel (2Cno1.go)
 - o Menghitung biaya pengiriman berdasarkan berat gram yang dimasukkan pengguna.
 - o Menggunakan fungsi hitungBiaya(beratGram int), dengan tarif dasar Rp 10.000 per kg dan tambahan Rp 500 per 100 gram untuk sisa gram.
 - o Memeriksa input agar tidak negatif dan menampilkan rincian berat serta total biaya dengan perhitungan yang akurat.

Dari ketiga program tersebut, dapat disimpulkan bahwa masing-masing memiliki tujuan spesifik dalam menyelesaikan masalah tertentu, yaitu penentuan tahun kabisat, perhitungan pendekatan akar 2, dan kalkulasi biaya pengiriman parsel. Semua program menggunakan fungsi terpisah untuk mengelola logika utama, memastikan efisiensi, dan mempermudah eksekusi dalam fungsi main().

V. REFRENSI

<https://sko.dev/wiki/input-dan-output>

<https://codingstudio.id/blog/golang-adalah/>

<https://dif.telkomuniversity.ac.id/tipe-data-pemrograman/>

<https://rpubs.com/maulidarahmah/829044>

https://repository.unikom.ac.id/62967/1/Materi%20Pertemuan%204_Labview%201%2BWhile%20Loop%20%2B%20Shift%20Register.pdf