

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



Oleh:

M.HANIF AL FAIZ

103112400042

12 IF 01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

Teori dasar mengenai pemrograman dalam bahasa Go mencakup beberapa konsep fundamental yang perlu dipahami oleh programmer, terutama bagi yang baru mulai belajar bahasa Go. Berikut adalah beberapa teori dasar terkait bahasa pemrograman Go yang dibahas dalam modul ini:

1. Struktur Program Go

- **Package main:** Setiap program utama dalam bahasa Go dimulai dengan deklarasi package main. Ini menunjukkan bahwa file ini adalah program utama yang dapat dieksekusi.
- **Fungsi main():** Fungsi main() adalah titik masuk dari program Go. Kode yang ada dalam fungsi ini adalah yang pertama kali dieksekusi ketika program dijalankan.
- **Komentar:** Komentar dalam Go digunakan untuk menjelaskan kode dan tidak akan dieksekusi oleh program. Komentar satu baris diawali dengan //, sementara komentar multi-baris diawali dengan /* dan diakhiri dengan */.

2. Kompilasi dan Eksekusi Program Go

- **Kompilasi:** Go adalah bahasa yang dikompilasi. Proses kompilasi akan memeriksa program dan mengubahnya menjadi file eksekusi yang bisa dijalankan. Perintah go build digunakan untuk mengkompilasi program.
- **Eksekusi:** Setelah program dikompilasi, file eksekusi (.exe untuk Windows) dapat dijalankan melalui terminal atau command prompt dengan mengetikkan nama file eksekusi.

3. Tipe Data dalam Go

Go menyediakan berbagai tipe data dasar yang sering digunakan dalam pemrograman. Tipe data ini termasuk:

- **Integer:** Untuk bilangan bulat. Ada beberapa jenis integer yang bisa digunakan, seperti int, int8, int32, int64, serta uint untuk bilangan bulat positif.
- **Real (Float):** Untuk bilangan desimal. Tipe ini mencakup float32 dan float64.
- **Boolean:** Untuk nilai logika true atau false.

- **Character:** Dalam Go, karakter bisa berupa tipe byte (alias uint8) atau rune (alias int32).
- **String:** Untuk urutan karakter yang membentuk teks.

4. Deklarasi dan Inisialisasi Variabel

Variabel harus dideklarasikan sebelum digunakan dalam Go. Variabel dapat dideklarasikan dengan tipe data tertentu atau secara otomatis ditentukan berdasarkan nilai awal.

- **Deklarasi eksplisit:** Menggunakan kata kunci var, seperti var a int untuk mendeklarasikan variabel a dengan tipe int.
- **Inisialisasi:** Variabel juga bisa diinisialisasi dengan nilai awal, seperti var a = 10, atau menggunakan operator penugasan singkat := seperti a := 10.

5. Operator dalam Go

Go mendukung berbagai operator untuk operasi matematika, logika, dan bitwise. Berikut beberapa contoh operator yang umum digunakan:

- **Operator Matematika:** +, -, *, /, % (modulo).
- **Operator Perbandingan:** ==, !=, <, <=, >, >=.
- **Operator Logika:** && (AND), || (OR), ! (NOT).
- **Operator Bitwise:** &, |, ^, &^, <<, >>.

6. Konversi Tipe Data

Go memiliki sistem tipe yang sangat ketat. Tidak ada konversi otomatis antara tipe data yang berbeda, sehingga perlu dilakukan konversi eksplisit menggunakan fungsi atau casting tipe. Misalnya, konversi antara int dan float dilakukan dengan cara:

```
intVar := int(floatVar)
```

7. Fungsi Input/Output (I/O)

Go menyediakan pustaka fmt untuk menangani input dan output standar:

- **Input:** Fungsi fmt.Scan, fmt.Scanln, atau fmt.Scanf digunakan untuk membaca data dari input.
- **Output:** Fungsi fmt.Print, fmt.Println, atau fmt.Printf digunakan untuk menampilkan data ke layar.

8. Konstanta Simbolik

Dalam Go, konstanta adalah nilai tetap yang tidak dapat diubah setelah didefinisikan. Konstanta biasanya digunakan untuk nilai-nilai yang bersifat global, seperti nilai pi (PI), atau nilai tetap lainnya yang sering digunakan dalam program. Konstanta dideklarasikan dengan menggunakan kata kunci `const`, contohnya:

```
const PI = 3.14159
```

9. Pentingnya Struktur Folder dalam Go

Program Go umumnya disusun dalam folder dengan struktur tertentu. Setiap program Go disimpan dalam folder dengan nama yang relevan, dan setiap program Go disarankan untuk disimpan dalam satu folder terpisah untuk menjaga struktur yang jelas.

Dengan pemahaman dasar ini, Anda dapat mulai menulis program Go, melakukan kompilasi, dan menjalankan program dengan benar.

II. GUIDED SOURCE CODE

```
package main

import "fmt"

func main() {

    var greetings = "Selamat di Dunia DAP"

    var a, b int

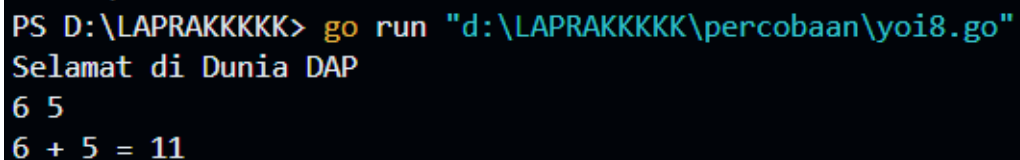
    fmt.Println(greetings)

    fmt.Scan(&a, &b)

    fmt.Printf("%v + %v = %v\n", a, b, a+b)

}
```

OUTPUT:



```
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\percobaan\yoi8.go"
Selamat di Dunia DAP
6 5
6 + 5 = 11
```

DESKRIPSI PROGRAM:

Pada kesempatan ini, kita akan membuat sebuah program dengan tujuan untuk melakukan operasi penjumlahan. Program tersebut akan diawali dengan menampilkan pesan "SELAMAT DATANG DI DUNIA DAP". Setelah itu, pengguna akan diminta untuk menginputkan dua nilai pada awal program. Nilai-nilai yang dimasukkan tersebut akan digunakan untuk melakukan operasi kalkulasi penjumlahan, dan hasil dari penjumlahan kedua nilai tersebut akan ditampilkan pada akhir program.

SOURCE CODE

```
package main
```

```
import "fmt"

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

OUTPUT:

```
PS D:\LAPRAKKKK> go run "d:\LAPRAKKKK\percobaan\yoi8.go"
Iterasi ke- 1
Iterasi ke- 2
Iterasi ke- 3
Iterasi ke- 4
Iterasi ke- 5
```

DESKRIPSI PROGRAM:

Program ini merupakan sebuah program yang menggunakan struktur perulangan (loop) dengan kondisi awal variabel **i** bernilai 1. Perulangan akan terus dilakukan selama nilai **i** masih kurang dari atau sama dengan 5. Program ini akan berhenti atau keluar dari perulangan ketika kondisi **i <= 5** tidak lagi terpenuhi, yaitu setelah nilai **i** melebihi 5.

SOURCE CODE

```
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:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\percobaan\yoi8.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes true, mendapat indeks A

```

DESKRIPSI PROGRAM:

Berikut adalah kalimat yang telah diubah menjadi bahasa formal:

Program ini adalah sebuah program yang menggunakan struktur percabangan, di mana jalannya program sudah ditentukan berdasarkan nilai-nilai yang akan diinputkan. Pada awalnya, program akan menerima nilai yang diinputkan oleh pengguna. Sebagai contoh, jika nilai yang dimasukkan adalah **a**, program kemudian akan memeriksa nilai tersebut terhadap kriteria yang telah ditentukan. Dalam kasus ini, program akan memeriksa

apakah nilai **a** lebih besar dari 75 dan apakah pengguna telah mengerjakan tugas besar (TUBES). Jika kedua kondisi tersebut terpenuhi, maka program akan memberikan nilai "A". Setelah program memeriksa apakah tugas besar telah dikerjakan, hasil akhirnya adalah bahwa program akan mengeluarkan indeks nilai "A" sesuai dengan hasil percabangan yang dilakukan.

III. UNGUIDED SOURCE CODE

```
//M.HANIF AL FAIZ  
  
//103112400042  
  
package main  
  
import "fmt"  
  
func main() {  
    var tahun int  
    var status bool  
    fmt.Scan(&tahun)  
    status = tahun%4 == 0  
    fmt.Println("TAHUN:", tahun)  
    fmt.Println("KABISAT:", status)  
}
```

OUTPUT:


```
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\percobaan\yoi10.go"
2016
TAHUN: 2016
KABISAT: true
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\percobaan\yoi10.go"
2000
TAHUN: 2000
KABISAT: true
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\percobaan\yoi10.go"
2018
TAHUN: 2018
KABISAT: false
```

DEKSRIPSI PRORGAM:

Program ini merupakan sebuah program yang dirancang untuk mendeteksi tahun kabisat. Tahun kabisat didefinisikan sebagai tahun yang dapat dibagi habis oleh 4. Program ini akan memeriksa apakah suatu tahun memenuhi kriteria tersebut dan menentukan apakah tahun tersebut termasuk tahun kabisat atau tidak.

SOURCE CODE

```
package main

import (
    "fmt"
    "math"
)

func hitungAkar2(K int) float64 {
    var hasil float64 = 1.0

    for k := 0; k <= K; k++ {
        nom := math.Pow(float64(4*k+2), 2)
        den := float64((4*k + 1) * (4*k + 3))
        hasil *= nom / den
    }
}
```

```

    }

    akar2 := hasil

    return akar2
}

func main() {

    var K int

    fmt.Print("Masukkan nilai K: ")

    fmt.Scan(&K)

    akar2 := hitungAkar2(K)

    fmt.Printf("Akar 2 = %.10f\n", akar2)
}

```

OUTPUT:

```

PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\percobaan\yoipart2.go"
Masukkan nilai K: 10
Akar 2 = 1.4062058441
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\percobaan\yoipart2.go"
Masukkan nilai K: 100
Akar 2 = 1.4133387072
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\percobaan\yoipart2.go"
Masukkan nilai K: 1000
Akar 2 = 1.4141252651

```

DESKRIPSI PROGRAM:

i

$$f(k) = \frac{(4k + 2)^2}{(4k + 1)(4k + 3)}$$

ii

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

Memasukkan salah satu bilangan irrasional dan menyelesaikannya dalam sebuah program yang dirancang oleh mahasiswa

SOURCE CODE

```
// M.HANIF AI FAIZ
// 103112400042

package main

import (
    "fmt"
)

func main() {
    var berat_asli int

    fmt.Print("Berat parsel (gram): ")

    fmt.Scanln(&berat_asli)

    kg := berat_asli / 1000
    gram := berat_asli % 1000

    fmt.Printf("Detail berat: %d kg + %d gr\n", kg, gram)

    biaya_kg := kg * 10000
```

```

var biaya_gram int

if gram >= 500 {

    biaya_gram = gram * 5

} else {

    biaya_gram = gram * 15

}

fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biaya_kg, biaya_gram)

fmt.Printf("Total biaya: Rp. %d", biaya_kg+biaya_gram)

}

```

OUTPUT:

```

PS D:\LAPRAK\> go run "d:\LAPRAK\percobaan\tempCodeRunnerFile.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\LAPRAK\> go run "d:\LAPRAK\percobaan\tempCodeRunnerFile.go"
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\LAPRAK\> go run "d:\LAPRAK\percobaan\tempCodeRunnerFile.go"
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 113750

```

DESKRIPSI PROGRAM:

Program yang dibuat untuk PT POS yang dimana PT POS tersebut membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parsel.

IV. KESIMPULAN

Dalam laporan praktikum ini, berbagai program yang mengimplementasikan konsep dasar dalam pemrograman dengan bahasa Go telah dikembangkan dan dijelaskan. Program-program tersebut mencakup:

1. Program Penjumlahan

Program ini meminta input dua nilai dan kemudian menghitung serta menampilkan hasil penjumlahan kedua nilai tersebut. Hal ini menunjukkan pemahaman tentang penggunaan input/output serta operasi matematika dalam Go.

2. Program Perulangan (Loop)

Program ini menggunakan struktur perulangan untuk menampilkan iterasi hingga mencapai jumlah yang diinginkan. Ini mengilustrasikan penerapan for loop dalam bahasa Go.

3. Program Percabangan (If-Else)

Program ini menggunakan percabangan untuk memberikan indeks berdasarkan nilai, kehadiran, dan status tugas besar (TUBES). Program ini juga menunjukkan penggunaan logika untuk menentukan hasil berdasarkan kondisi yang diberikan.

4. Program Deteksi Tahun Kabisat

Program ini mengimplementasikan deteksi tahun kabisat dengan memeriksa apakah suatu tahun dapat dibagi habis oleh 4. Ini menggambarkan konsep percabangan dan operasi modulus.

5. Program Perhitungan Fungsi dan Akar

Program ini menghitung nilai dari fungsi matematika dan akar kuadrat dengan menggunakan metode perulangan dan fungsi matematika. Hal ini menunjukkan penerapan matematika dalam pemrograman.

6. Program Penghitungan Biaya Kirim

Program ini menghitung biaya kirim berdasarkan berat parcel dengan membagi biaya per kilogram dan per gram. Program ini menggabungkan perhitungan matematis dan penggunaan percabangan untuk menentukan tarif biaya.

Secara keseluruhan, laporan ini menunjukkan penerapan konsep-konsep dasar pemrograman seperti struktur kontrol, perulangan, percabangan, dan pemrosesan input/output dalam bahasa Go. Semua program yang dikembangkan bertujuan untuk memberikan pemahaman yang lebih baik tentang bagaimana cara kerja bahasa Go dalam menyelesaikan masalah-masalah dasar dalam pemrograman.

V. REFERENSI

Modul Praktikum Algoritma dan Pemrograman 2, Telkom University, 2025.