

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



Oleh:

MUHAMMAD FAUZAN

103112400064

12 IF 01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## I. DASAR TEORI

### 1. Struktur Program dalam Bahasa Go

Bahasa pemrograman Go memiliki struktur program yang khas, di mana setiap program utama terdiri dari dua komponen utama:

- **package main:** Menandakan bahwa file ini adalah program utama yang dapat dieksekusi.
- **func main():** Fungsi utama tempat eksekusi program dimulai. Selain itu, komentar dalam Go dapat dibuat dengan dua cara:
  - `//` untuk komentar satu baris.
  - `/*` untuk komentar lebih dari satu baris.

### 2. Koding, Kompilasi, dan Eksekusi Program Go

- Program Go harus disimpan dalam file berekstensi **.go**.
- Kompilasi dilakukan menggunakan perintah **go build**.
- Setelah dikompilasi, program dapat dieksekusi langsung melalui terminal.
- Bahasa Go adalah bahasa yang dikompilasi, sehingga program harus melewati tahap kompilasi sebelum dijalankan.

### 3. Tipe Data dan Variabel dalam Go

Variabel dalam Go harus dideklarasikan sebelum digunakan dan memiliki tipe data yang ketat. Beberapa tipe data yang umum digunakan antara lain:

#### 3.1 Tipe Data dalam Go dan Fungsinya

1. **Integer (int, int8, int16, int32, int64, uint, uint8, uint16, uint32, uint64)**
  - Digunakan untuk menyimpan bilangan bulat.
  - Contoh penggunaan:
    - `var angka int = 10`
    - `var kecil int8 = 127`
    - `var besar int64 = 1000000000`
2. **Float (float32, float64)**
  - Digunakan untuk menyimpan bilangan desimal.
  - Contoh penggunaan:
    - `var suhu float64 = 36.5`
3. **Boolean (bool)**
  - Digunakan untuk menyimpan nilai **true** atau **false**.
  - Contoh penggunaan:
    - `var aktif bool = true`

#### 4. Karakter (byte, rune)

- byte adalah alias dari uint8 dan digunakan untuk menyimpan karakter dalam format ASCII.
- rune adalah alias dari int32 dan digunakan untuk menyimpan karakter Unicode.
- Contoh penggunaan:
- `var huruf byte = 'A' // ASCII`

`var simbol rune = '€' // Unicode`

#### 5. String

- Digunakan untuk menyimpan teks.
- Contoh penggunaan:

`var nama string = "Go Language"`

- String bisa diakses sebagai array karakter:

`fmt.Println(nama[0]) // Menampilkan karakter pertama`

### 3.2 Deklarasi dan Inisialisasi Variabel dalam Go

Variabel harus dideklarasikan sebelum digunakan. Ada beberapa cara untuk mendeklarasikan variabel:

- **Deklarasi tanpa inisialisasi**

`var a int // Nilai default adalah 0`

- **Deklarasi dengan inisialisasi**

- `var b int = 10`

`var c = 20 // Go secara otomatis menentukan tipe data`

- **Deklarasi menggunakan shorthand :=**

`d := 30 // Sama dengan var d int = 30`

### 3.3 Konversi Tipe Data

Konversi tipe data dalam Go dilakukan secara eksplisit dengan menuliskan tipe yang diinginkan:

`var x int = 10`

`var y float64 = float64(x) // Konversi dari int ke float64`

`var z string = strconv.Itoa(x) // Konversi dari int ke string`

#### 4. Struktur Kontrol dalam Go

Struktur kontrol dalam Go terdiri dari dua kategori utama:

##### 1. Perulangan (Looping):

- **For-loop** digunakan untuk mengulang eksekusi kode selama suatu kondisi terpenuhi.
- **While-loop** dalam Go diimplementasikan dengan `for` tanpa inisialisasi dan pembaruan.
- **Repeat-until** dapat diimplementasikan menggunakan `for` dengan kondisi penghentian di dalam loop.

##### 2. Percabangan (Branching):

- **If-else** digunakan untuk menjalankan kode berdasarkan suatu kondisi.
- **Switch-case** memungkinkan eksekusi kode berdasarkan nilai tertentu.

#### 5. Konstanta Simbolik

Konstanta dalam Go dideklarasikan menggunakan kata kunci **const**.

Contoh:

```
const PI = 3.1415926535
```

```
const MARKER = "AKHIR"
```

#### 6. Operasi Dasar dalam Go

Operasi yang dapat dilakukan pada tipe data di Go antara lain:

- **Aritmatika:** `+`, `-`, `*`, `/`, `%`
- **Logika:** `&&`, `||`, `!`
- **Perbandingan:** `==`, `!=`, `<`, `>`, `<=`, `>=`
- **Bitwise:** `&`, `|`, `^`, `<<`, `>>`

#### 7. Implementasi dalam Praktikum

Modul ini berisi berbagai latihan untuk mengimplementasikan konsep dasar pemrograman dalam Go, seperti:

- Membuat program untuk membaca dan menampilkan data.
- Menentukan apakah suatu tahun merupakan tahun kabisat.
- Menghitung volume dan luas bola.
- Mengonversi temperatur dari Celsius ke Fahrenheit, Reamur, dan Kelvin.
- Menampilkan faktor dari suatu bilangan dan menentukan apakah bilangan tersebut prima.

## II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

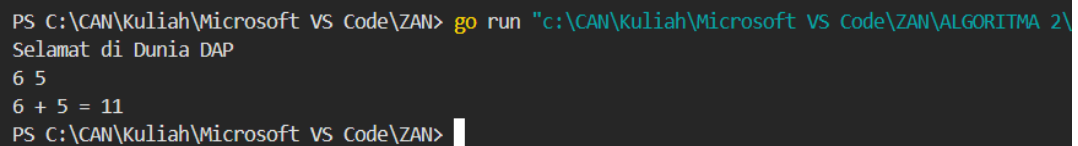
Contoh 1

```
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)
}
```

Screenshots Output



```
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\ALGORITMA 2\
Selamat di Dunia DAP
6 5
6 + 5 = 11
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> |
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini memulai dengan menampilkan pesan sapaan "SELAMAT DATANG DI DUNIA DAP", lalu meminta untuk memasukkan dua angka. Setelah angka dimasukkan, program akan menjumlahkan kedua angka tersebut dan menampilkan hasilnya.

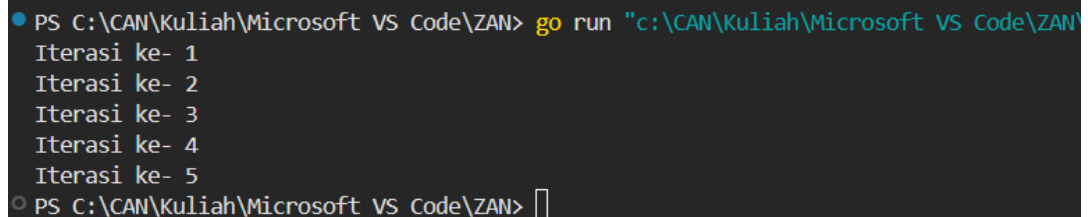
## Contoh 2

```
package main

import "fmt"

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

## Screenshots Output



```
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\
Iterasi ke- 1
Iterasi ke- 2
Iterasi ke- 3
Iterasi ke- 4
Iterasi ke- 5
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> 
```

// Foto hasil dari menjalankan code

Deskripsi: Program diatas adalah implementasi perulangan **for** Perulangan ini digunakan untuk menampilkan teks **"Iterasi ke-"** diikuti oleh nomor iterasi yang sedang berjalan, mulai dari 1 hingga 5.

### Contoh 3

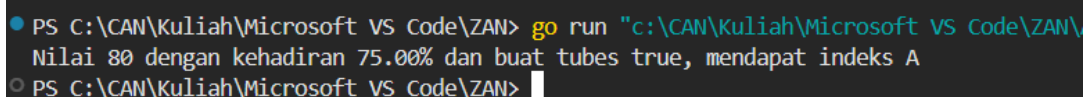
```
package main

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)
}
```

### Screenshots Output



```
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\
Nilai 80 dengan kehadiran 75.00% dan buat tubes true, mendapat indeks A
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN>
```

// Foto hasil dari menjalankan code

Deskripsi: Program diatas adalah Program untuk menentukan **indeks nilai mahasiswa** berdasarkan **nilai ujian, kehadiran, dan tugas besar** menggunakan struktur if-else.

### III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

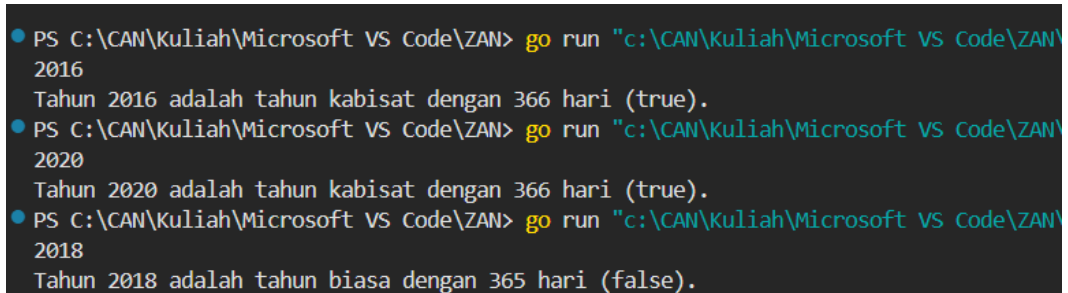
#### Soal 1

```
// MUHAMMAD FAUZAN
//103112400064
package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Masukkan tahun: ")
    fmt.Scan(&tahun)
    if (tahun%4 == 0 && tahun%100 != 0) || (tahun%400 == 0) {
        fmt.Printf("Tahun %d adalah tahun kabisat dengan 366 hari (true).\n",
tahun)
    } else {
        fmt.Printf("Tahun %d adalah tahun biasa dengan 365 hari (false).\n", tahun)
    }
}
```

#### Screenshots Output



```
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\
2016
Tahun 2016 adalah tahun kabisat dengan 366 hari (true).
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\
2020
Tahun 2020 adalah tahun kabisat dengan 366 hari (true).
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\
2018
Tahun 2018 adalah tahun biasa dengan 365 hari (false).
```

// Foto hasil dari menjalankan code

Deskripsi: Program diatas adalah Program ini digunakan untuk menentukan apakah suatu tahun merupakan **tahun kabisat** atau bukan. Tahun kabisat adalah tahun yang **habis dibagi 4**, tetapi jika juga **habis dibagi 100**, maka tidak dianggap kabisat, kecuali jika **habis dibagi 400**. Program akan meminta pengguna memasukkan tahun, mengevaluasi berdasarkan aturan tersebut, dan menampilkan hasilnya.



## Soal 2

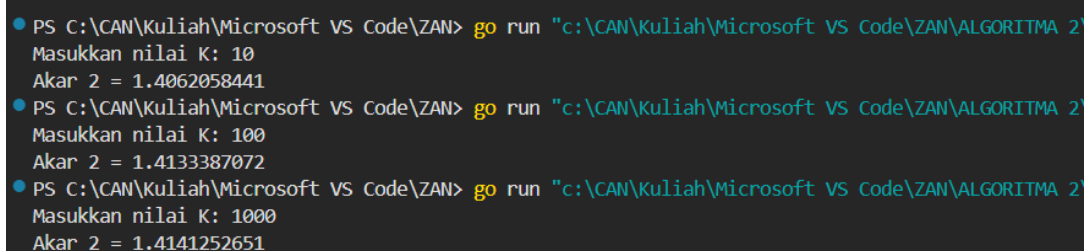
```
//MUHAMMAD FAUZAN
//103112400064
package main

import (
    "fmt"
    "math"
)

func hitungAkar2(K int) float64 {
    akar2 := 1.0
    for k := 0; k <= K; k++ {
        akar2 *= math.Pow(float64(4*k+2), 2) / float64((4*k+1)*(4*k+3))
    }
    return akar2
}

func main() {
    var K int
    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&K)
    fmt.Printf("Akar 2 = %.10f\n", hitungAkar2(K))
}
```

## Screenshots Output



```
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\ALGORITMA 2"
Masukkan nilai K: 10
Akar 2 = 1.4062058441
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\ALGORITMA 2"
Masukkan nilai K: 100
Akar 2 = 1.4133387072
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\ALGORITMA 2"
Masukkan nilai K: 1000
Akar 2 = 1.4141252651
```

// Foto hasil dari menjalankan code

Deskripsi: Program di atas adalah sebuah program yang menerima input sebuah bilangan bulat **K**, kemudian menghitung dan menampilkan **pendekatan nilai akar 2** ( $\sqrt{2}$ ) menggunakan metode perkalian tak hingga.

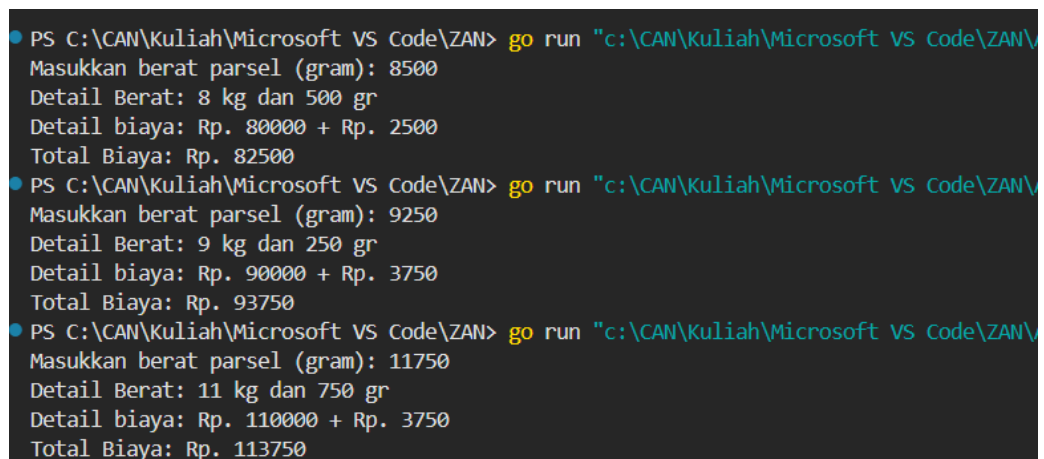
### Soal 3

```
//MUHAMMAD FAUZAN
//103112400064
package main

import (
    "fmt"
)

func main() {
    var berat, hargaGram int
    fmt.Print("Masukkan berat parsel (gram): ")
    fmt.Scanln(&berat)
    kilogram := berat / 1000
    sisaGram := berat % 1000
    fmt.Printf("Detail Berat: %d kg dan %d gr\n", kilogram, sisaGram)
    hargaKg := kilogram * 10000
    switch {
    case sisaGram >= 500:
        hargaGram = sisaGram * 5
    default:
        hargaGram = sisaGram * 15
    }
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", hargaKg, hargaGram)
    fmt.Printf("Total Biaya: Rp. %d\n", hargaKg+hargaGram)
}
```

### Screenshots Output



```
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\
Masukkan berat parsel (gram): 8500
Detail Berat: 8 kg dan 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total Biaya: Rp. 82500
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\
Masukkan berat parsel (gram): 9250
Detail Berat: 9 kg dan 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total Biaya: Rp. 93750
PS C:\CAN\Kuliah\Microsoft VS Code\ZAN> go run "c:\CAN\Kuliah\Microsoft VS Code\ZAN\
Masukkan berat parsel (gram): 11750
Detail Berat: 11 kg dan 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total Biaya: Rp. 113750
```

// Foto hasil dari menjalankan code

Deskripsi: Program diatas adalah program BiayaPos untuk menghitung biaya kirim berdasarkan berat parsel.

#### **IV. KESIMPULAN**

## **V. REFERENSI**

MODUL 2. REVIEW STRUKTUR KONTROL, Praktikum Algoritma dan Pemrograman 2