

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
MODUL 2  
MATERI RIVIEW STRUKTUR KONTROL**



Oleh: Dimas Fanny Hebrasianto Permadi

NAMA: Dimas Ramadhani

NIM: 103112400065

KELAS: IF-12-02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

# I. DASAR TEORI

## 1. STRUKTUR PROGRAM GO

Dalam membuat program menggunakan Bahasa GO, ada dua komponen utama:

- package main merupakan penanda bahwa file ini program utama
- func main() berisi kode utama dalam program

Untuk memberi komentar dapat menggunakan:

- // dapat digunakan sampai dengan satu baris saja
- /\*...\*/ dapat digunakan sampai lebih dari satu baris

## 2. Koding, Kompilasi, dan Eksekusi Go

- File program go harus diberi format atau ekstensi .go
- Untuk mengkompilasi dapat dilakukan di terminal/shell dengan mengetik go build file.go

## 3. Tipe Data dan Variabel Dalam Go

- Integer: int, int8, int32, int64
- Unsigned Integer: uint, uint8, uint32, uint64
- Float: float32, float64
- Boolean: bool (true/false)
- String: string
- Karakter: rune, byte

Deklarasi variabel dapat dilakukan:

```
var a int = 10
    b := 20
```

## 4. Struktur Kontrol Go

### a) Perulangan (Looping)

- For inisialisasi, kondisi, dan update:

```
for i := 0; i < count; i++ {
    fmt.Print(i)
}
```

- While-loop menggunakan for:

```
i := 0
for i < 10 {
    fmt.Println(i)
    i++}
```

- Infinite loop:

```
for {
    fmt.Println("Looping tanpa henti")
}
```

b) Percabangan (Conditional Statement)

- If-Else:

```
if x > 10 {
    fmt.Println("Lebih besar dari 10")
} else {
    fmt.Println("Kurang dari atau sama dengan 10")
}
```

- Switch-Case:

```
switch nilai {
case "A":
    fmt.Println("Sangat baik")
case "B":
    fmt.Println("Baik")
default:
    fmt.Println("Cukup")
}
```

5. Konversi Tipe Data:

- `int(float64(angka))` untuk mengonversi float ke integer.
- `strconv.Itoa(angka)` untuk mengonversi integer ke string.
- `strconv.Atoi(StringAngka)` untuk mengonversi string ke integer.

6. Input dan Output dalam Go

a. Input:

```
var nama string
fmt.Scanln(&nama)
```

b. Output:

```
fmt.Println("Halo, ", nama)
```

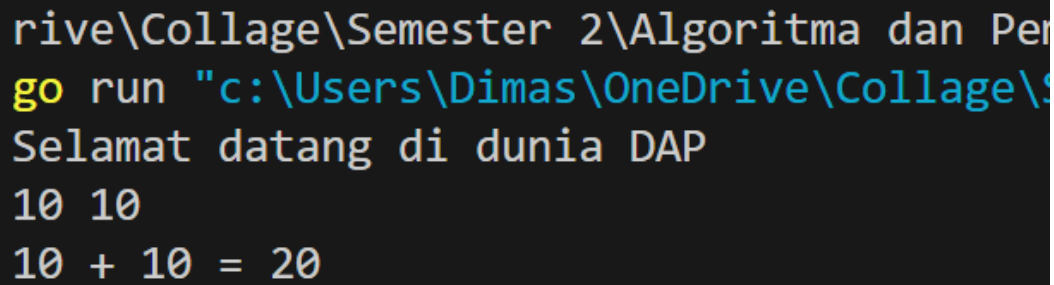
## II. GUIDED

### 1. Program 1

- Source Code:

```
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)
}
```

- Screenshot hasil program:



```
rive\Collage\Semester 2\Algoritma dan Per
go run "c:\Users\Dimas\OneDrive\Collage\
Selamat datang di dunia DAP
10 10
10 + 10 = 20
```

- Penjelasan:

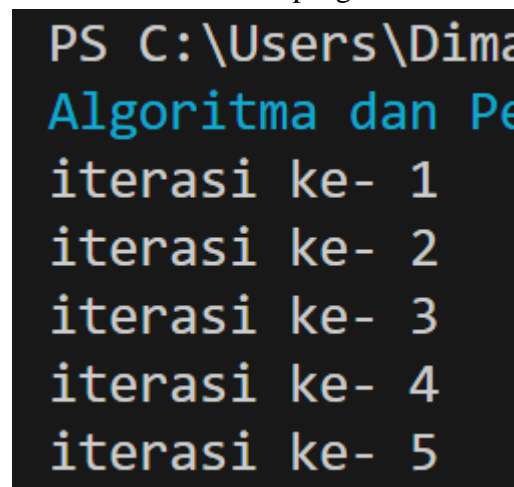
Dalam program ini ada beberapa variabel seperti variabel greetings bertipekan string dan didalam variabel ini ada kalimat “Selamat datang di dunia DAP”, lalu variabel a dan b bertipekan integer yang akan diinputkan oleh pengguna. Lalu akan tercetak variabel greetings tadi lalu pengguna harus menginputkan dua buah angka, setelah itu program akan menjumlahkan kedua angka tersebut dan mendapat hasilnya.

## 2. Program 2

- Source Code:

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

- Screenshoot hasil program:



```
PS C:\Users\Dima
Algoritma dan Pe
iterasi ke- 1
iterasi ke- 2
iterasi ke- 3
iterasi ke- 4
iterasi ke- 5
```

- Penjelasan:

Program ini bertujuan untuk mencetak beberapa kalimat dalam satu struktur menggunakan perulangan. Memiliki satu buah variabel bertipekan integer. Jika kita menjalankan programnya maka akan langsung tercetak beberapa kalimat.

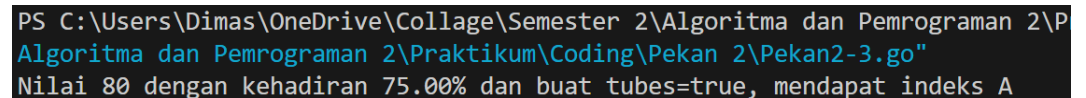
### 3. Program 3

- 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)
}
```

- Screenshoot hasil program:



```
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2\Algoritma dan Pemrograman 2\Praktikum\Coding\Pekan 2\Pekan2-3.go
Nilai 80 dengan kehadiran 75.00% dan buat tubes=true, mendapat indeks A
```

- Penjelasan:

Program ini bertujuan untuk menilai predikat mahasiswa berdasar nilai, presentase hadir, dan tugas besar. Memiliki beberapa variabel, pctHadir bertipe integer dengan nilai 0.75 bertipe floatingpoint64, adaTubes bertipe Boolean nilainya true, nilai bertipe integer dengan nilai 80, indeks bertipe string. Program ini menggunakan struktur percabangan dengan kondisi yang sudah ada pada source code. Lalu tercetak kalimat yang sesuai dengan screenshoot hasil program diatas.

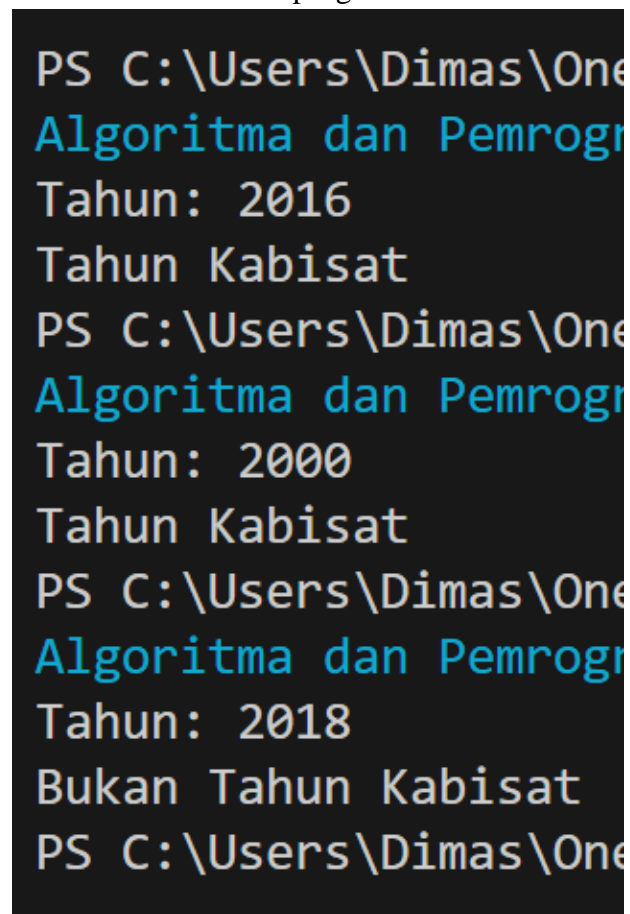
### III. UNGUIDED

#### 1. 2A nomor 2

- Source code:

```
package main
import "fmt"
func main() {
    var tahun int
    fmt.Print("Tahun: ")
    fmt.Scan(&tahun)
    if tahun%400 == 0 {
        fmt.Println("Tahun Kabisat")
    } else if tahun%4 == 0 && tahun%100 != 0 {
        fmt.Println("Tahun Kabisat")
    } else {
        fmt.Println("Bukan Tahun Kabisat")
    }
}
```

- Screenshoot hasil program:



```
PS C:\Users\Dimas\OneDrive\Desktop> go run main.go
Algoritma dan Pemrograman
Tahun: 2016
Tahun Kabisat
PS C:\Users\Dimas\OneDrive\Desktop> go run main.go
Algoritma dan Pemrograman
Tahun: 2000
Tahun Kabisat
PS C:\Users\Dimas\OneDrive\Desktop> go run main.go
Algoritma dan Pemrograman
Tahun: 2018
Bukan Tahun Kabisat
PS C:\Users\Dimas\OneDrive\Desktop>
```

- **Penjelasan:**

Program ini bertujuan untuk mengetahui apakah tahun yang kita cek merupakan tahun kabisat atau tidak. Memiliki satu variabel yaitu tahun bertipe integer yang akan sebagai tempat inputan kita. Program ini menggunakan struktur percabangan dengan kondisi seperti pada source code. Lalu untuk hasilnya seperti pada screenshot hasil pembahasan.

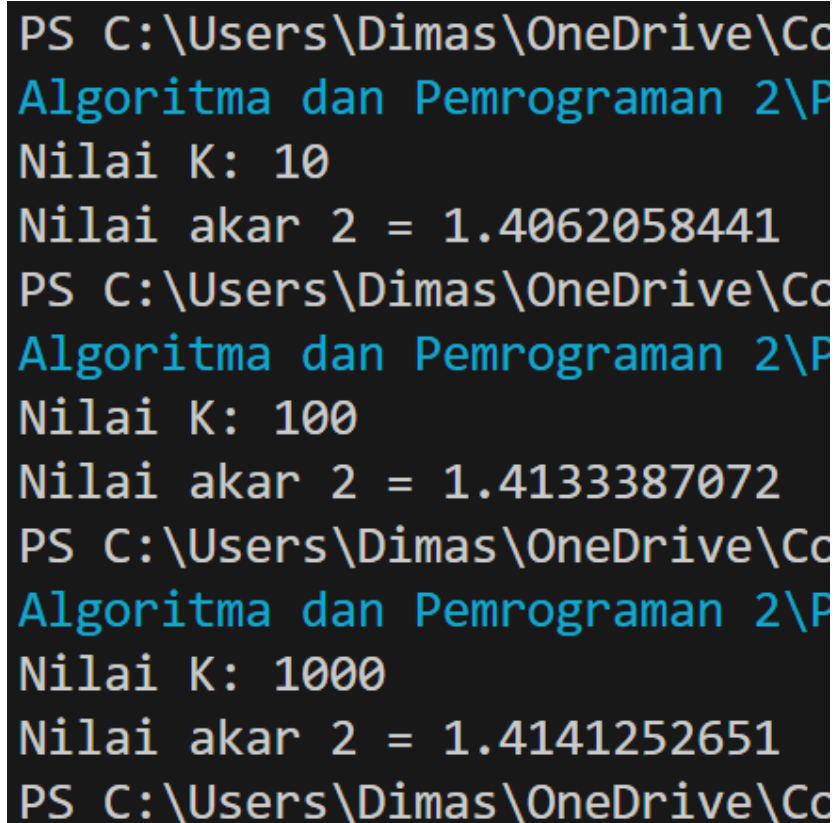


## 2. 2B nomor 4

- Source code:

```
package main
import "fmt"
func main() {
    var hasil, a float64
    var k int
    a = 0
    hasil = 1
    fmt.Print("Nilai K: ")
    fmt.Scan(&k)
    for i := 0; i <= k; i++ {
        temp := (4*a + 2) * (4*a + 2) / (4*a + 1) / (4*a + 3)
        hasil *= temp
        a++
    }
    fmt.Printf("Nilai akar 2 = %.10f", hasil)
}
```

- Screenshoot hasil program:



```
PS C:\Users\Dimas\OneDrive\Co
Algoritma dan Pemrograman 2\
Nilai K: 10
Nilai akar 2 = 1.4062058441
PS C:\Users\Dimas\OneDrive\Co
Algoritma dan Pemrograman 2\
Nilai K: 100
Nilai akar 2 = 1.4133387072
PS C:\Users\Dimas\OneDrive\Co
Algoritma dan Pemrograman 2\
Nilai K: 1000
Nilai akar 2 = 1.4141252651
PS C:\Users\Dimas\OneDrive\Co
```

- **Penjelasan:**

Program ini bertujuan untuk mengetahui pendekatan dari akar dua sesuai dengan batas yang kita masukkan. Memiliki beberapa variabel, hasil, temp, dan a beritpekan float64, k dan i bertipe integer. Program ini menggunakan struktur perulangan dengan kondisi dan iterasi sesuai dengan source code. Jika kita menjalankan program, awal akan tercetak kalimat "Niali K: " lalu dapat memasukkan angka sesuai dengan code diatas. Setelah itu masuk ke perulangan dengan kondisi sesuai dengan code. Lalu hasil sesuai dengan screenshoot hasil pembahasan.

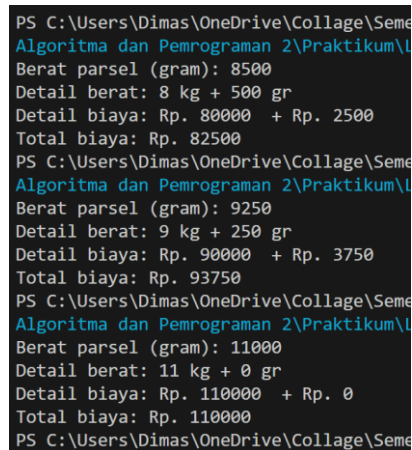
### 3. 2C nomor 1

- Source code:

```
package main
import "fmt"
func main() {
    var berat, kg, gr, harga_kg, harga_gr, totalbiaya int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&berat)
    kg = berat / 1000
    gr = berat % 1000

    if gr >= 500 {
        harga_gr = gr * 5
    } else {
        harga_gr = gr * 15
    }
    if kg <= 10 {
        harga_kg = kg * 10000
        totalbiaya = harga_kg + harga_gr
    } else {
        harga_kg = kg * 10000
        totalbiaya = harga_kg
    }
    fmt.Printf("Detail berat: %d kg + %d gr\n", kg, gr)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", harga_kg,
harga_gr)
    fmt.Printf("Total biaya: Rp. %d", totalbiaya)
}
```

- Screenshoot hasil program:



```
PS C:\Users\Dimas\OneDrive\Collage\Seme
Algoritma dan Pemrograman 2\Praktikum\l
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\Dimas\OneDrive\Collage\Seme
Algoritma dan Pemrograman 2\Praktikum\l
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Users\Dimas\OneDrive\Collage\Seme
Algoritma dan Pemrograman 2\Praktikum\l
Berat parsel (gram): 11000
Detail berat: 11 kg + 0 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS C:\Users\Dimas\OneDrive\Collage\Seme
```

- **Penjelasan:**

Program ini bertujuan untuk menghitung harga dari berat yang kita masukkan dengan satuan gram. Memiliki beberapa variabel berat, kg, gr, harga\_gr, harga\_kg, dan total\_biaya bertipe integer.

Program ini menggunakan struktur percabangan dengan kondisi sesuai pada source code. Jika kita menjalankan program, pertama akan tercetak "Berat parcel (gram): " lalu kita dapat memasukkan angka. Setelah untuk menghitung berat kg digunakan pembagian, sedangkan untuk menghitung gram digunakan modulus. Setelah itu masuk ke percabangan. Dan akan tercetak hasilnya sesuai dengan kondisi berat parcel tersebut.

#### **IV. KESIMPULAN**

Dari modul ini, dapat disimpulkan bahwa bahasa pemrograman Go memiliki struktur yang sederhana namun kuat untuk mengembangkan aplikasi. Dengan sistem kompilasi yang efisien, Go memungkinkan pengembang untuk menulis kode yang cepat dan mudah dipahami. Tipe data yang ketat dan sistem perulangan serta percabangan yang fleksibel membuat Go menjadi bahasa yang cocok untuk berbagai keperluan pemrograman.

Selain itu, konsep dasar seperti deklarasi variabel, operasi aritmatika, input-output, serta struktur kontrol dalam Go memberikan fondasi penting bagi pemrogram untuk memahami dan menguasai bahasa ini. Dengan memahami materi yang telah dibahas dalam modul ini, peserta diharapkan mampu menulis dan menjalankan program dalam bahasa Go dengan baik serta memahami cara kerja kompilasi dan eksekusi dalam lingkungan pengembangan.

## **V. REFERENSI**

**Prayogo, N. A. (2024). *Dasar Pemrograman Golang* (Versi 4.0.20240830). Retrieved from <https://github.com/novalagung/dasarpemrogramangolang>**