

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 2

MATERI



Oleh:

ABISAR FATHIR

103112400068

12-IF-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Struktur Program dalam Go

Setiap program Go memiliki komponen utama yaitu package main sebagai penanda program utama, import untuk pustaka yang dibutuhkan, dan func main() sebagai fungsi utama tempat eksekusi program dimulai.

Tipe Data dan Variabel

Go memiliki beberapa tipe data dasar, termasuk int untuk bilangan bulat, float64 untuk bilangan desimal, bool untuk nilai benar/salah, dan string untuk teks. Variabel dapat dideklarasikan menggunakan var atau langsung diinisialisasi dengan :=.

Operasi Aritmatika

Bahasa Go mendukung operasi matematika seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Operasi ini digunakan dalam berbagai perhitungan, termasuk pemrosesan data numerik dan algoritma logika.

Struktur Kontrol

Go menggunakan if-else untuk percabangan dalam pengambilan keputusan. Untuk perulangan, Go hanya memiliki for, yang bisa digunakan sebagai pengganti while atau do-while dalam bahasa lain.

Input dan Output

Go menggunakan fmt.Scan() dan fmt.Scanln() untuk menerima input dari pengguna. Sementara itu, output ditampilkan menggunakan fmt.Print(), fmt.Println(), atau fmt.Printf() untuk format yang lebih terstruktur.

Kompilasi dan Eksekusi Program

Go menggunakan metode kompilasi untuk mengeksekusi program. Program ditulis dalam file dengan ekstensi .go, lalu dikompilasi menggunakan perintah go build. Setelah itu, program dapat dijalankan melalui terminal atau command prompt.

II. GUIDED

1. Program ke 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)
}
```

Output Program:

```
Selamat datang di dunia DAP
5 5
5 + 5 = 10
```

Deskripsi Program: Program Diatas Berfungsi untuk menampilkan sebuah pesan, kemudian meminta input dua bilangan bulat dari pengguna, dan akhirnya mencetak hasil penjumlahan kedua bilangan tersebut.

2. Program ke 2

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 %v%% dan buat tubes=%t, mendapat indeks %s\n", nilai, pctHadir*100, adaTubes, indeks)

}
```

Output Program:

```
PS E:\Go\Tugasmodul_2> E::; cd 'E:\Go\Tugasmodul_2'; ${env:GOPATH}='C:
Nilai 80 dengan kehadiran 75% dan buat tubes=true, mendapat indeks A
PS E:\Go\Tugasmodul_2> |
```

Deskripsi Program: Kode Go (Golang) ini adalah sebuah program yang menentukan indeks nilai berdasarkan nilai ujian, persentase kehadiran, dan keberadaan tugas besar (tubes).

3. Program ke 3

Source Code:

```
package main

import "fmt"

func main() {

for i := 1; i <= 5; i++ {

fmt.Println("iterasi ke-", i)

}

}
```

Output Program:

```
iterasi ke- 1
iterasi ke- 2
iterasi ke- 3
iterasi ke- 4
iterasi ke- 5
```

Deskripsi Program: sebuah program yang melakukan iterasi perulangan sama ke 5

III. UNGUIDED

1. Program ke 1

Source Code:

```
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("%d adalah tahun kabisat.\n", tahun)

} else {

fmt.Printf("%d bukan tahun kabisat.\n", tahun)

}

}
```

Output Program:

```
Masukkan tahun: 2016
2016 adalah tahun kabisat.
```

Deskripsi Program: program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat (true) atau bukan (false)

2. Program ke 2

Source Code:

```
package main

import "fmt"

func main() {

    var k int

    var fK, hasil, term float64

    hasil = 1

    fmt.Print("Nilai K: ")

    fmt.Scan(&k)

    for i := 0; i <= k; i++ {

        term = (float64((4*i + 2) * (4*i + 2))) / ((4*i + 1) * (4*i + 3))

        hasil *= term

        if i == k {

            fK = term

        }

    }

    fmt.Printf("Nilai f(K) = %.10f\n", fK)

    fmt.Printf("Nilai hampiran \u221A2 = %.10f\n", hasil)

}
```

Output Program:

```
Nilai K: 10
Nilai f(K) = 1.0005672150
Nilai hampiran \u221A2 = 1.4062058441
```

Deskripsi Program: sebuah program yang menerima input sebuah bilangan sebagai K, kemudian menghitung dan menampilkan nilai $f(K)$ sesuai persamaan di atas.

3. Program ke 3

Source Code:

```
package main

import "fmt"

func main() {
    var Berat int

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

    fmt.Scan(&Berat)

    kg := Berat / 1000

    gr := Berat % 1000

    biayaKg := kg * 10000

    var biayaGr int

    if kg > 10 {

        biayaGr = 0

    } else if gr >= 500 {

        biayaGr = gr * 5

    } else {

        biayaGr = gr * 15

    }
}
```



```
totalBiaya := biayaKg + biayaGr
```

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

```
fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg, biayaGr)
```

```
fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
```

```
}
```

Output Program:

```
Berat Parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
```

Deskripsi Program: Program ini menghitung biaya pengiriman parcel berdasarkan beratnya dalam gram dengan ketentuan tertentu..

IV. KESIMPULAN

Modul 2 membahas dasar-dasar pemrograman dalam Go, termasuk struktur program, tipe data, operasi aritmatika, percabangan, perulangan, serta cara kompilasi dan eksekusi program.

Konsep-konsep ini adalah hal dasar dalam penulisan program yang lebih kompleks, seperti perhitungan biaya pengiriman, pengolahan data, dan analisis logika dalam pemrograman.

V. REFERENSI