

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III  
FUNGSI**



**Disusun Oleh :**

**Ghilbran Alfaries Pryma / 2311102267**

**IF 11 05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom., M.Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

**Fungsi dalam Golang** adalah sebuah blok instruksi yang memiliki tujuan tertentu, seperti menghitung nilai, memproses data, atau melakukan tindakan tertentu. Fungsi ini dibuat agar program lebih terorganisir dan lebih mudah dipelihara. Dengan menggunakan fungsi, kita dapat mengelompokkan kode yang serupa dalam satu tempat, lalu memanggil fungsi tersebut kapanpun diperlukan, tanpa harus mengulang penulisan kode yang sama berulang kali.

### Konsep Dasar Fungsi

Dalam Golang, fungsi terdiri dari beberapa elemen penting:

- **Nama fungsi:** Ini adalah label atau identitas yang digunakan untuk memanggil fungsi.
- **Parameter:** Fungsi bisa menerima input berupa parameter. Misalnya, jika fungsi dibuat untuk menghitung jumlah dua angka, maka kedua angka tersebut bisa disertakan sebagai parameter.
- **Nilai Kembali:** Setelah fungsi menyelesaikan tugasnya, ia bisa mengembalikan hasilnya kepada program. Misalnya, setelah menghitung dua angka, fungsi bisa mengembalikan hasil penjumlahannya.
- **Blok Instruksi:** Ini adalah kumpulan perintah atau logika yang didefinisikan di dalam fungsi untuk menyelesaikan tugas tertentu.

### Jenis-Jenis Fungsi dalam Golang

- **Fungsi Tanpa Parameter dan Nilai Kembali:** Fungsi ini tidak membutuhkan input dan tidak mengembalikan hasil apapun. Biasanya digunakan untuk menjalankan suatu tindakan sederhana yang tidak melibatkan perhitungan atau input dari luar.
- **Fungsi dengan Parameter:** Fungsi ini menerima satu atau lebih parameter dari luar. Parameter adalah nilai yang diberikan ke fungsi agar dapat memproses data yang spesifik. Misalnya, jika fungsi ingin menjumlahkan dua angka, angka-angka tersebut akan diterima sebagai parameter.

- **Fungsi dengan Nilai Kembali:** Setelah fungsi menyelesaikan tugasnya, ia dapat mengembalikan satu atau lebih hasil kepada bagian lain dari program. Misalnya, setelah fungsi menghitung hasil penjumlahan, hasil tersebut akan dikembalikan agar bisa digunakan di bagian lain dari program.
- **Fungsi dengan Multiple Return Values:** Salah satu fitur unik Golang adalah kemampuannya untuk mengembalikan lebih dari satu nilai dari sebuah fungsi. Misalnya, sebuah fungsi dapat mengembalikan hasil penjumlahan dan hasil perkalian dua angka sekaligus.
- **Variadic Functions:** Golang mendukung fungsi yang bisa menerima sejumlah parameter yang tidak terbatas dengan tipe data yang sama. Misalnya, kita bisa membuat fungsi untuk menjumlahkan semua angka yang kita berikan, tidak peduli berapa banyaknya.

### **Fungsi Sebagai Parameter**

Fungsi juga bisa diteruskan sebagai argumen ke fungsi lain. Artinya, kita bisa membuat sebuah fungsi yang menerima fungsi lain sebagai input dan menggunakannya di dalam fungsi tersebut. Ini memberikan fleksibilitas dalam cara fungsi digunakan.

- **Fungsi Anonim (Anonymous Functions)**  
Fungsi anonim adalah fungsi tanpa nama. Fungsi ini biasanya digunakan saat kita hanya membutuhkan fungsi sekali dalam sebuah bagian program, tanpa perlu mendefinisikan nama khusus untuk fungsi tersebut.
- **Fungsi Penutup (Closure)**  
Closure adalah fungsi yang dapat "mengingat" variabel dari ruang lingkup luar tempat fungsi itu dibuat. Fungsi ini biasanya digunakan dalam konteks yang lebih kompleks, di mana kita membutuhkan fungsi untuk menyimpan status atau nilai dari luar ruang lingkungannya.

## II. GUIDED I

### Source Code

```
package main
import "fmt"
func main (){
    var a, b int
    fmt.Scan(&a, &b)
    if a >= b{
        fmt.Println(permutasi(a,b))
    }else{
        fmt.Println(permutasi(b,a))
    }
}
func faktorial (n int) int{
    var hasil int = 1
    var i int
    for i = 1; i <= n; i++){
        hasil = hasil * i
    }
    return hasil
}
func permutasi (n,r int) int{
    return faktorial(n) / faktorial(n-r)
}
```

### Screenshoot Output

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 3\Guided 1> go run
5 10
30240
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 3\Guided 1> go run
4 6
360
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 3\Guided 1> █
```

### Deskripsi program

#### 1. Fungsi main():

- Meminta dua bilangan dari pengguna (a dan b).
- Jika b lebih besar dari a, nilainya ditukar agar a selalu lebih besar atau sama dengan b.
- Hasil dari fungsi permutasi(a, b) dicetak.

#### 2. Fungsi faktorial(n):

- Menghitung faktorial dari bilangan n menggunakan perulangan.

3. **Fungsi permutasi(n, r):**
- Menghitung nilai permutasi

## GUIDED II

### Source Code

```
package main

import (
    "fmt"
)

func main() {
    var sisi int
    fmt.Print("Masukkan panjang sisi persegi: ")

    // Mencoba untuk membaca input dari pengguna
    _, err := fmt.Scan(&sisi)

    // Memeriksa apakah terjadi kesalahan saat membaca input
    if err != nil || sisi <= 0 {
        fmt.Println("Input tidak valid! Pastikan Anda memasukkan angka positif.")
        return
    }

    luas := luasPersegi(sisi)
    keliling := kelilingPersegi(sisi)

    fmt.Printf("Luas persegi: %d\n", luas)
    fmt.Printf("Keliling persegi: %d\n", keliling)
}

// Fungsi untuk menghitung luas persegi
func luasPersegi(sisi int) int {
    return sisi * sisi
}

// Fungsi untuk menghitung keliling persegi
func kelilingPersegi(sisi int) int {
    return 4 * sisi
}
```

## Screenshoot Output

```
Masukkan panjang sisi persegi: 15
Luas persegi: 225
Keliling persegi: 60
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 3\Guided 2>
```

## Deskripsi Program

### 1. Fungsi main():

- Program meminta pengguna untuk memasukkan panjang sisi persegi menggunakan fungsi `fmt.Print`.
- Input dari pengguna dibaca menggunakan `fmt.Scan` dan disimpan dalam variabel `sisi`. Selain itu, variabel `err` digunakan untuk mendeteksi jika ada kesalahan dalam proses input.
- Jika terjadi kesalahan dalam membaca input atau jika pengguna memasukkan nilai sisi yang kurang dari atau sama dengan nol (angka negatif atau nol tidak valid untuk panjang sisi persegi), maka program akan mencetak pesan kesalahan: "Input tidak valid! Pastikan Anda memasukkan angka positif." dan keluar dari program.
- Jika input valid, program akan menghitung luas dan keliling persegi menggunakan fungsi `luasPersegi()` dan `kelilingPersegi()`.
- Hasil perhitungan untuk luas dan keliling dicetak ke layar.

### 2. Fungsi `luasPersegi(sisi int) int`:

- Fungsi ini menerima parameter berupa panjang sisi (`sisi`).
- Fungsi mengembalikan hasil dari perhitungan luas persegi, yaitu  $sisi * sisi$ .

### 3. Fungsi `kelilingPersegi(sisi int) int`:

- Fungsi ini menerima parameter panjang sisi (`sisi`).
- Fungsi mengembalikan hasil dari perhitungan keliling persegi, yaitu  $4 * sisi$ .

### III. UNGUIDED

#### 1. Soal Studi Case

Minggu Ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalian membantu Jonas? (tidak tentunya ya :p)

**Masukan** terdiri dari empat buah bilangan asli a, b, c, dan d yang dipisahkan oleh spasi, dengan syarat  $a \geq c$  dan  $b \geq d$ .

Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi a terhadap c, sedangkan baris kedua adalah hasil permutasi dan kombinasi b terhadap d.

**Catatan:** permutasi (P) dan kombinasi (C) dari n terhadap r ( $n \geq r$ ) dapat dihitung dengan menggunakan persamaan berikut!

#### Source Code

```
package main

import "fmt"

// Fungsi utama
func main() {
    var a, b, c, d int

    // Input nilai
    fmt.Print("input nilai: ")
    fmt.Scan(&a, &b, &c, &d)

    // Pengecekan kondisi apakah a >= c dan b >= d
    if a >= c && b >= d {
        // Menghitung dan menampilkan permutasi dan kombinasi untuk a
        // dan c
        fmt.Println("Permutasi:", permutasi(a, c))
        fmt.Println("Kombinasi:", kombinasi(a, c))

        // Menghitung dan menampilkan permutasi dan kombinasi untuk b
        // dan d
        fmt.Println("Permutasi:", permutasi(b, d))
        fmt.Println("Kombinasi:", kombinasi(b, d))
    }
}
```

```

    } else {
        fmt.Println("tidak memenuhi kondisi")
    }
}

// Fungsi untuk menghitung faktorial
func faktorial(n int) int {
    hasil := 1
    for i := 1; i <= n; i++ {
        hasil *= i
    }
    return hasil
}

// Fungsi untuk menghitung permutasi
// rumus:  $P(n, r) = n! / (n - r)!$ 
func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

// Fungsi untuk menghitung kombinasi
// rumus:  $C(n, r) = n! / (r! * (n - r)!)$ 
func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}

```

### Screenshoot Output\

```

input nilai: 5 10 3 10
Permutasi: 60
Kombinasi: 10
Permutasi: 3628800
Kombinasi: 1

```

### Deskripsi Program

Program ini menghitung permutasi dan kombinasi dari dua pasang bilangan `(a, c)` dan `(b, d)` setelah memeriksa apakah `a >= c` dan `b >= d`. Jika kondisi terpenuhi, program menghitung permutasi dan kombinasi menggunakan rumus  $P(n, r)$  untuk permutasi dan  $C(n, r)$  untuk kombinasi. Fungsi `faktorial()` digunakan



untuk membantu perhitungan ini. Jika kondisi tidak terpenuhi, program menampilkan pesan "tidak memenuhi kondisi".

## 2. Soal Studi Case

Diberikan tiga buah fungsi matematika yaitu  $f(x) = x^2$ ,  $g(x) = x - 2$  dan  $h(x) = x + 1$ . Fungsi komposisi  $(f \circ g \circ h)(x)$  artinya adalah  $f(g(h(x)))$ . Tuliskan  $f(x)$ ,  $g(x)$  dan  $h(x)$  dalam bentuk function.

Masukan terdiri dari sebuah bilangan bulat  $a$ ,  $b$  dan  $c$  yang dipisahkan oleh spasi.

Keluaran terdiri dari tiga baris. Baris pertama adalah  $(f \circ g \circ h)(a)$ , baris kedua  $(g \circ h \circ f)(b)$ , dan baris ketiga adalah  $(h \circ f \circ g)(c)$ !

### Source Code

```
package main

import (
    "fmt"
)

// Fungsi f(x) = x^2
func f(x int) int {
    return x * x
}

// Fungsi g(x) = x - 2
func g(x int) int {
    return x - 2
}

// Fungsi h(x) = x + 1
func h(x int) int {
    return x + 1
}

// Fungsi komposisi fogoh(x) = f(g(h(x)))
func fogoh(x int) int {
    return f(g(h(x)))
}
```

```
// Fungsi komposisi gohof(x) = g(h(f(x)))
func gohof(x int) int {
    return g(h(f(x)))
}

// Fungsi komposisi hofog(x) = h(f(g(x)))
func hofog(x int) int {
    return h(f(g(x)))
}

func main() {
    // Membaca input dari user
    var a, b, c int
    fmt.Scan(&a, &b, &c)

    // Menghitung dan mencetak hasil komposisi fungsi
    fmt.Println(fogoh(a)) // Baris pertama (fogoh)(a)
    fmt.Println(gohof(b)) // Baris kedua (gohof)(b)
    fmt.Println(hofog(c)) // Baris ketiga (hofog)(c)
}
```

### Screenshoot Output

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 3\Unguided 2> go ru
7 2 10
36
3
65
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 3\Unguided 2> █
```

### Deskripsikan Program

Program ini menghitung komposisi dari tiga fungsi:  $f(x) = x^2$ ,  $g(x) = x - 2$ , dan  $h(x) = x + 1$ . Setelah pengguna memasukkan tiga nilai 'a', 'b', dan 'c', program menghitung tiga komposisi fungsi:  $f(g(h(a)))$ ,  $g(h(f(b)))$ , dan  $h(f(g(c)))$ . Hasil dari setiap komposisi kemudian ditampilkan. Sebagai contoh, untuk input 'a = 7', 'b = 2', dan 'c = 10', program menghasilkan 36, 3, dan 65.

### 3. Soal Studi Case

Buatlah program Pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebut hingga salah satu kantong terpal berisi 9 kg atau lebih.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

### Source Code

```
package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung jarak antara dua titik (a, b) dan (c, d)
func jarak(a, b, c, d float64) float64 {
    return math.Sqrt((a-c)*(a-c) + (b-d)*(b-d))
}

// Fungsi untuk mengecek apakah titik (x, y) berada di dalam lingkaran
// dengan pusat (cx, cy) dan radius r
func didalam(cx, cy, r, x, y float64) bool {
    return jarak(cx, cy, x, y) <= r
}

func main() {
    var cx1, cy1, r1 float64 // Lingkaran 1
    var cx2, cy2, r2 float64 // Lingkaran 2
    var x, y float64         // Titik sembarang

    // Input lingkaran 1: pusat dan radius
    fmt.Println("Masukkan koordinat pusat lingkaran 1 (cx1 cy1) dan radius r1:")
    fmt.Scan(&cx1, &cy1, &r1)

    // Input lingkaran 2: pusat dan radius
    fmt.Println("Masukkan koordinat pusat lingkaran 2 (cx2 cy2) dan radius r2:")
    fmt.Scan(&cx2, &cy2, &r2)

    // Input titik sembarang
    fmt.Println("Masukkan koordinat titik sembarang (x y):")
    fmt.Scan(&x, &y)

    // Mengecek apakah titik berada di dalam lingkaran 1 atau 2
```

```

dalamLingkaran1 := didalam(cx1, cy1, r1, x, y)
dalamLingkaran2 := didalam(cx2, cy2, r2, x, y)

// Menentukan posisi titik berdasarkan hasil pengecekan
if dalamLingkaran1 && dalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if dalamLingkaran1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if dalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

## Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 3\Unguided 3> go run "
Masukkan koordinat pusat lingkaran 1 (cx1 cy1) dan radius r1:
1 1 5
Masukkan koordinat pusat lingkaran 2 (cx2 cy2) dan radius r2:
8 8 4
Masukkan koordinat titik sembarang (x y):
2 2
Titik di dalam lingkaran 1
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 3\Unguided 3>

```

## Deskripsi Program

Program ini mengecek apakah sebuah titik berada di dalam dua lingkaran berdasarkan pusat dan radiusnya.

Deskripsi Program:

- Fungsi `jarak(a, b, c, d float64)`: Menghitung jarak antara dua titik (a, b) dan (c, d) menggunakan rumus Euclidean.
- Fungsi `didalam(cx, cy, r, x, y float64)`: Memeriksa apakah titik (x, y) berada di dalam lingkaran dengan pusat (cx, cy) dan radius r.
- Fungsi `main()`:

- Meminta pengguna memasukkan pusat dan radius untuk dua lingkaran.
- Meminta pengguna untuk memasukkan koordinat titik sembarang  $(x, y)$ .
- Mengecek posisi titik terhadap kedua lingkaran dan mencetak hasilnya.

## DAFTAR PUSTAKA

- **Donovan, A. A., & Kernighan, B. W. (2015).** *The Go Programming Language*. Addison-Wesley Professional.

Buku ini merupakan sumber yang komprehensif tentang bahasa pemrograman Go, termasuk konsep dasar seperti fungsi, tipe data, dan cara penggunaan yang efektif.

- **Balbaert, I. (2016).** *The Way to Go: A Thorough Introduction to the Go Programming Language*. iUniverse.

Buku ini menyediakan pengantar lengkap untuk pemrograman dengan Go, membahas berbagai fitur bahasa, termasuk fungsi, kontrol alur, dan pengorganisasian kode.

- **Cox, R. (2016).** *Effective Go*. Retrieved from [https://golang.org/doc/effective\\_go.html](https://golang.org/doc/effective_go.html)

Dokumen ini memberikan panduan praktis dan tips terbaik untuk menulis kode Go yang baik, termasuk penggunaan fungsi dan pemrograman yang efisien.

- **Go Documentation (n.d.).** *Functions in Go*. Retrieved from <https://golang.org/doc/>

Dokumentasi resmi Go memberikan penjelasan mendetail tentang fungsi, cara mendefinisikan, dan contoh penggunaan, serta berbagai konsep terkait.

- **McGrath, C. (2018).** *Go Programming Blueprints: A Practical Guide to Building Real-World, Production-Ready Solutions in Go*. Packt Publishing.

Buku ini berfokus pada penerapan praktis dari Go, termasuk struktur fungsi dalam aplikasi nyata dan bagaimana mengorganisasi kode untuk pengembangan yang efisien.

- **Laforenza, A. (2021).** *Hands-On Go Programming: Develop a Strong Understanding of the Go Programming Language by Building Real-World Applications*. Packt Publishing.

Buku ini menawarkan pendekatan praktis dalam mempelajari Go, dengan fokus pada pembuatan aplikasi nyata, termasuk pemahaman mendalam tentang fungsi.