

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 3  
FUNCTION**



Oleh:

**ANANDA BASKORO PUTRA**

2311102187

IF 11 02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO 2024**

# I. DASAR TEORI

## 1. Definisi Function

Fungsi (function) adalah blok kode mandiri yang dirancang untuk menjalankan tugas tertentu. Fungsi menerima input (parameter), melakukan operasi berdasarkan input tersebut, dan dapat mengembalikan hasil berupa output. Fungsi digunakan untuk mengorganisasi program agar lebih modular, memudahkan pemeliharaan, serta menghindari pengulangan kode.

## 2. Deklarasi Function

Deklarasi fungsi dilakukan dengan menentukan nama fungsi, parameter, tipe data parameter, dan tipe data hasil yang akan dikembalikan (jika ada). Contoh deklarasi fungsi di Go:

```
func namaFungsi(parameter1 tipeData1, parameter2 tipeData2) tipeDataHasil {  
  
    // blok kode  
  
    return hasil  
  
}
```

## 3. Pemanggilan Function

Pemanggilan fungsi dilakukan dengan menggunakan nama fungsi diikuti dengan argumen yang sesuai dengan parameter yang telah didefinisikan dalam deklarasi. Contoh pemanggilan fungsi:

```
hasil := namaFungsi(argumen1, argumen2)
```

## I. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

### Guided 1

```
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:\Bedor\KULIAH\laprak modul 3> go run "d:\Bedor\KULIAH\laprak modul 3\Guided\guided1.go"
7
9
181440
PS D:\Bedor\KULIAH\laprak modul 3> █
```

## Deskripsi Program

Program diatas berfungsi untuk menghitung permutasi dari dua bilangan, di mana permutasi adalah cara menghitung kemungkinan urutan elemen dari kumpulan tanpa pengulangan. Fungsi permutasi(n, r) menghitung jumlah cara memilih dan mengatur r elemen dari n elemen berbeda. Program ini menerima dua bilangan a dan b, kemudian menghitung permutasi berdasarkan nilai yang lebih besar sebagai n (total elemen) dan nilai yang lebih kecil sebagai r (elemen yang diambil).

## Guided 2

```
package main

import "fmt"

// Fungsi buat menghitung faktorial
func factorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
    return result
}

// Fungsi buat menghitung permutasi
```

```

func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

// Fungsi buat menghitung kombinasi

func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func main() {
    // Input 4 bilangan
    var a, b, c, d int

    fmt.Println("Masukkan bilangan a, b, c, d (engan spasi): ")

    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    // Cek syarat a >= c dan b >= d
    if a >= c && b >= d {
        // Menghitung permutasi dan kombinasi a dan c
        permutasiAC := permutation(a, c)
        kombinasiAC := combination(a, c)

        // Menghitung permutasi dan kombinasi b dan d
        permutasiBD := permutation(b, d)
        kombinasiBD := combination(b, d)

        // Output hasil
        fmt.Println("Permutasi(a, c) dan Kombinasi(a, c):", permutasiAC,
kombinasiAC)
        fmt.Println("Permutasi(b, d) dan Kombinasi(b, d):", permutasiBD,
kombinasiBD)
    } else {
        fmt.Println("Syarat a >= c dan b >= d tidak terpenuhi.")
    }
}

```

```
}
```

## Screenshoot Output

```
PS D:\Bedor\KULIAH\laprak modul 3> go run "d:\Bedor\KULIAH\laprak modul 3\Guided\guided2.go"
Masukkan bilangan a, b, c, d (dengan spasi):
a b c d
Permutasi(a, c) dan Kombinasi(a, c): 1 1
Permutasi(b, d) dan Kombinasi(b, d): 1 1
PS D:\Bedor\KULIAH\laprak modul 3> |
```

## Deskripsi Program

Program diatas berfungsi untuk menghitung permutasi dan kombinasi dari dua pasang bilangan berdasarkan syarat tertentu. Permutasi menghitung urutan yang mungkin dari elemen yang dipilih, sedangkan kombinasi menghitung cara memilih elemen tanpa memperhatikan urutan. Program menerima empat bilangan sebagai input, kemudian menghitung permutasi dan kombinasi untuk pasangan (a, c) dan (b, d), tetapi hanya jika syarat  $a \geq c$  dan  $b \geq d$  terpenuhi. Jika syarat tersebut tidak dipenuhi, program tidak akan melakukan perhitungan dan menampilkan pesan kesalahan.

## II. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

### Unguided 2b\_1

```
package main

import (
    "fmt"
    "math"
)

func jarak(x1, y1, x2, y2 float64) float64 {
    return math.Sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2))
}

func diDalamLingkaran(cx, cy, r, x, y float64) bool {
    return jarak(cx, cy, x, y) <= r
}

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, x, y float64

    fmt.Println("Masukkan koordinat pusat dan jari-jari lingkaran 1:")
    fmt.Scan(&cx1, &cy1, &r1)

    fmt.Println("Masukkan koordinat pusat dan jari-jari lingkaran 2:")
    fmt.Scan(&cx2, &cy2, &r2)

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

    diLingkaran1 := diDalamLingkaran(cx1, cy1, r1, x, y)
    diLingkaran2 := diDalamLingkaran(cx2, cy2, r2, x, y)

    if diLingkaran1 && diLingkaran2 {
        fmt.Println("Titik berada di dalam lingkaran 1 dan 2")
    }
}
```

```

    } else if diLingkaran1 {

        fmt.Println("Titik berada di dalam lingkaran 1")

    } else if diLingkaran2 {

        fmt.Println("Titik berada di dalam lingkaran 2")

    } else {

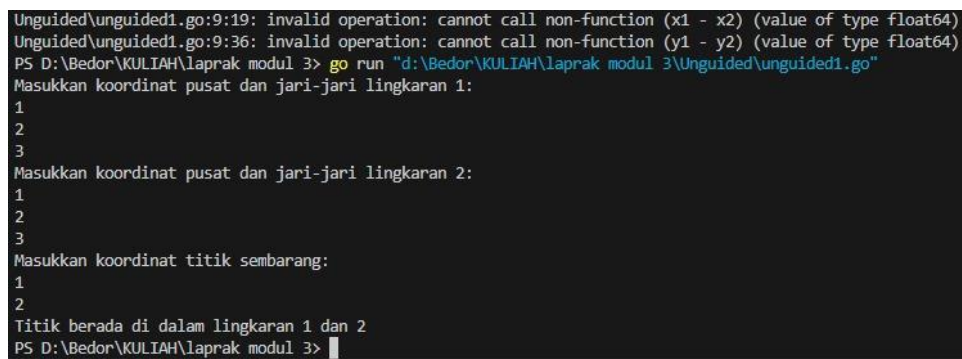
        fmt.Println("Titik berada di luar lingkaran 1 dan 2")

    }

}

```

## Screenshoot Output



```

Unguided\unguided1.go:9:19: invalid operation: cannot call non-function (x1 - x2) (value of type float64)
Unguided\unguided1.go:9:36: invalid operation: cannot call non-function (y1 - y2) (value of type float64)
PS D:\Bedor\KULIAH\laprak modul 3> go run "d:\Bedor\KULIAH\laprak modul 3\Unguided\unguided1.go"
Masukkan koordinat pusat dan jari-jari lingkaran 1:
1
2
3
Masukkan koordinat pusat dan jari-jari lingkaran 2:
1
2
3
Masukkan koordinat titik sembarang:
1
2
Titik berada di dalam lingkaran 1 dan 2
PS D:\Bedor\KULIAH\laprak modul 3>

```

## Deskripsi Program

Program diatas berfungsi untuk menentukan apakah suatu titik berada di dalam dua lingkaran yang berbeda. Pertama, pengguna diminta untuk memasukkan koordinat pusat dan jari-jari dari dua lingkaran (lingkaran 1 dan lingkaran 2), serta koordinat titik sembarang. Fungsi jarak digunakan untuk menghitung jarak antara pusat lingkaran dan titik yang diberikan menggunakan rumus jarak Euclidean. Fungsi `diDalamLingkaran` memeriksa apakah jarak tersebut kurang dari atau sama dengan jari-jari lingkaran. Di dalam fungsi main, hasil pemeriksaan ditampilkan, memberitahukan pengguna apakah titik tersebut berada di dalam lingkaran 1, lingkaran 2, keduanya, atau di luar keduanya.