

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 3**

**FUNGSI**



**Oleh:**

**TSAQIF KANZ AHMAD**

**2311102075**

**IF-11-02**

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## I. DASAR TEORI

### A. Definisi Fungsi

Fungsi adalah satu kesatuan rangkaian instruksi yang memberikan atau menghasilkan suatu nilai dan biasanya memetakan input ke suatu nilai yang lain. Oleh karena itu, fungsi selalu menghasilkan/mengembalikan nilai. Suatu subprogram dikatakan fungsi apabila:

1. Ada deklarasi tipe nilai yang dikembalikan, dan
2. Terdapat kata kunci return dalam badan subprogram. Maka fungsi digunakan jika suatu nilai biasanya diperlukan, seperti:

Assignment nilai ke suatu variabel

Bagian dari ekspresi

Bagian dari argumen suatu subprogram, dsb.

Karena itu selalu pilih nama fungsi yang menggambarkan nilai, seperti kata benda dan kata sifat. Contoh nama-nama fungsi: **median**, **rerata**, **nilaiTerbesar**, **ketemu**, **selesai**.

### B. Deklarasi Fungsi

Fungsi dideklarasikan menggunakan kata kunci func diikuti oleh nama fungsi, parameter (jika ada), dan tipe kembalian (jika ada).

```
func namaFungsi(parameter1 tipe1, parameter2 tipe2) tipeKembalian
{
    // Blok kode fungsi
}
```

- **NamaFungsi** : Nama fungsi yang digunakan untuk memanggilnya.
- **Parameter** : Variabel yang diterima fungsi sebagai input (opsional).
- **TipeKembalian** : Tipe data yang dikembalikan oleh fungsi (jika ada).
- **Blok kode** : Kode yang akan dieksekusi saat fungsi dipanggil.

### C. Fungsi Parameter dan Tipe Pengembalian

- **Parameter**: Fungsi dapat memiliki nol atau lebih parameter yang dipisahkan oleh koma. Setiap parameter memiliki nama dan tipe data yang ditentukan.

```
func cetakNama(nama string) {
    fmt.Println("Nama:", nama)
}
```

- **Tipe Pengembalian**: Fungsi dapat mengembalikan satu atau lebih nilai. Tipe pengembalian ditulis setelah daftar parameter.

```
func kaliDua(angka int) int {
    return angka * 2
}
```

## D. Fungsi Tanpa Parameter dan Tanpa Tipe Kembalian

Fungsi Tanpa Parameter dan Tanpa Tipe Kembalian adalah Fungsi tidak selalu memerlukan parameter atau tipe kembalian. Jika tidak ada, fungsi bisa dideklarasikan tanpa parameter atau tipe kembalian.

```
func cetakHello() {  
    fmt.Println("Hello, World!")  
}
```

## E. Fungsi dengan Multiple Return Values

Fungsi ini mengembalikan lebih dari satu nilai. Hal ini sering digunakan dalam error handling.

```
func bagi(a int, b int) (int, error) {  
    if b == 0 {  
        return 0, fmt.Errorf("pembagian dengan nol tidak diperbolehkan")  
    }  
    return a / b, nil  
}
```

## F. Fungsi Anonim (Anonymous Function)

Pada Fungsi ini bisa didefinisikan tanpa nama dan langsung disimpan ke dalam variabel atau digunakan sebagai parameter.

```
var cetak = func(pesan string) {  
    fmt.Println(pesan)  
}
```

## G. Fungsi Variadic (Variadic Function)

Fungsi variadic adalah fungsi yang bisa menerima jumlah parameter yang tidak terbatas. Parameter variadic menggunakan tanda ... sebelum tipe datanya.

```
func jumlahSemua(angka ...int) int {  
    total := 0  
    for _, n := range angka {  
        total += n  
    }  
    return total  
}
```

## H. First-Class Function

Pada fungsi ini dianggap sebagai first-class citizen, artinya fungsi bisa diperlakukan seperti nilai, yaitu bisa:

- Disimpan dalam variabel.
- Dikirim sebagai argumen ke fungsi lain.
- Dikembalikan dari fungsi lain.

Contoh:

```
func eksekusi(f func(int) int, nilai int) int {  
    return f(nilai)  
}
```

```
}
```

## I. Penerima (Receiver) pada Metode

Fungsi juga bisa menjadi metode dalam Go dengan menggunakan penerima (*receiver*). Ini digunakan dalam konteks tipe data tertentu (misalnya, struct).

```
type Lingkaran struct {  
    JariJari float64  
}  
  
func (l Lingkaran) Luas() float64 {  
    return 3.14 * l.JariJari * l.JariJari  
}
```

## I. GUIDED

- 1) Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilai faktorial dan permutasi. **Masukan** terdiri dari dua buah bilangan positif a dan b. **Keluaran** berupa sebuah bilangan bulat yang menyatakan nilai a permutasi b apabila  $a \geq$  atau b permutasi a untuk kemungkinan yang lain.

### 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))
}
```

### SCREENSHOT OUTPUT :

```
PS C:\Users\ACER\Documents\golang> go run "c:\Users\ACER\Documents\golang\Modul 3 Alpro Laprak\Guided\No.1.go"
28 10
-932
PS C:\Users\ACER\Documents\golang> |
```

**PENJELASAN PROGRAM :**

*Program tersebut adalah program untuk menghitung permutasi dari dua bilangan bulat menggunakan faktorial. Pada program utama program meminta input dua bilangan a dan b, kemudian program akan membandingkan nilai dua bilangan a dan b. Jika  $a \geq b$ , maka menghitung permutasi a terhadap b ( $nPr$ ). Jika sebaliknya, permutasi dihitung dengan membalik nilai b dan a. Terdapat fungsi faktorial untuk menghitung faktorial dari bilangan n. Proses perhitungan dilakukan melalui loop yang mengalikan nilai hasil dari 1 hingga n. Kemudian ada fungsi permutasi untuk menghitung permutasi  $nPr$  menggunakan rumus  $nPr = \frac{n!}{(n-r)!}$  Dan memanggil faktorial(n) dan faktorial(n-r) untuk menghitung hasil permutasi.*

2). 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? Masukan terdiri dari empat buah bilangan asli a, b, c, dan d yang dipisahkan oleh spasi, dengan syarat  $a \leq c$  dan  $b \leq d$ . 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 r) dapat dihitung dengan menggunakan persamaan berikut!

$$P(n, r) = \frac{n!}{(n-r)!}, \text{ sedangkan } C(n, r) = \frac{n!}{r!(n-r)!}$$

**Contoh :**

No	Masukan	Keluaran	Penjelasan
1	5 10 3 10	60 10 3628800 1	$P(5, 3) = 5! / 2! = 120 / 2 = 60$ $C(5, 3) = 5! / (3! \times 2!) = 120 / 12 = 10$ $P(10, 10) = 10! / 0! = 3628800 / 1 = 3628800$ $C(10, 10) = 10! / (10! \times 0!) = 10! / 10! = 1$
2	8 0 2 0	56 28 1 1	

#### SOURCE CODE

```
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 (dengan 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.")
    }
}

```

## SCREENSHOT OUTPUT

```

PS C:\Users\ACER\Documents\golang> go run "c:\Users\ACER\Documents\golang\Modul 3 Alpro Laprak\Guided\No.2.go"
Masukkan bilangan a, b, c, d (dengan spasi):
5 9 4 7
Permutasi(a, c) dan Kombinasi(a, c): 120 5
Permutasi(b, d) dan Kombinasi(b, d): 181440 36
PS C:\Users\ACER\Documents\golang> go run "c:\Users\ACER\Documents\golang\Modul 3 Alpro Laprak\Guided\No.2.go"
Masukkan bilangan a, b, c, d (dengan spasi):
5 6 7
Syarat a >= c dan b >= d tidak terpenuhi.
PS C:\Users\ACER\Documents\golang>

```



## **PENJELASAN PROGRAM**

*Program tersebut adalah Program menghitung permutasi dan kombinasi untuk dua pasang bilangan (a, c) dan (b, d) jika syarat  $a \geq c$  dan  $b \geq d$  terpenuhi. Pada program tersebut terdapat fungsi **factorial** yang menghitung faktorial dari n. Jika n adalah 0 atau 1, hasilnya 1. Untuk nilai lebih dari 1, faktorial dihitung dengan perkalian bertahap. Kemudian ada fungsi Permutation yang menghitung permutasi menggunakan rumus  $nPr = n! / (n-r)!$  dengan bantuan fungsi faktorial. dan juga terdapat fungsi combination untuk menghitung kombinasi menggunakan rumus  $nCr = n! / (r! * (n-r)!)$  dengan fungsi faktorial. Pada fungsi utama program meminta input 4 bilangan a, b, c, d. Kemudian mengecek syarat  $a \geq c$  dan  $b \geq d$ . Jika syarat terpenuhi, program menghitung permutasi dan kombinasi untuk (a, c) dan (b, d) menggunakan fungsi yang sudah dibuat. Hasil perhitungan ditampilkan, jika tidak, program menampilkan pesan bahwa syarat tidak terpenuhi.*

## II. UNGUIDED

1). Lingkaran] Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx;cy) dengan radius r. Apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang (x, y) berdasarkan dua lingkaran tersebut.

**Masukan** terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat.

**Keluaran** berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 2", "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2".

**Contoh :**

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3	Titik di dalam lingkaran 2
	4 5 6 7 8	
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

### SOURCE CODE

```
package main

import (
    "fmt"
    "math"
)

func distance(a, b, c, d float64) float64 {
    return math.Sqrt((a-c)*(a-c) + (b-d)*(b-d))
}

func isInside(cx, cy, r, x, y float64) bool {
    return distance(cx, cy, x, y) <= r
}
```

```

func main() {
    var circles [2]struct {
        cx, cy, r float64
    }

    fmt.Println("Masukkan data lingkaran:")
    for i := 0; i < 2; i++ {
        fmt.Printf("Lingkaran %d (cx, cy, r): ", i+1)
        fmt.Scan(&circles[i].cx, &circles[i].cy, &circles[i].r)
    }

    var x, y float64
    fmt.Println("Masukkan koordinat titik (x, y):")
    fmt.Scan(&x, &y)

    inCircles := [2]bool{}
    for i := 0; i < 2; i++ {
        inCircles[i] = isInside(circles[i].cx, circles[i].cy,
circles[i].r, x, y)
    }

    result := ""
    for i := 0; i < 2; i++ {
        if inCircles[i] {
            result += fmt.Sprintf("Titik di dalam lingkaran %d ",
i+1)
        }
    }
    if result == "" {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    } else {
        fmt.Println(result)
    }
}

```

## SCREENSHOT OUTPUT

```

PS C:\Users\ACER\Documents\golang> go run "c:\Users\ACER\Documents\golang\Modul 3 Alpro Laprak\Unguided\No.1.go"
Masukkan data lingkaran:
Lingkaran 1 (cx, cy, r): 1 2 3
Lingkaran 2 (cx, cy, r): 4 5 6
Masukkan koordinat titik (x, y):
7 8
Titik di dalam lingkaran 2
PS C:\Users\ACER\Documents\golang> go run "c:\Users\ACER\Documents\golang\Modul 3 Alpro Laprak\Unguided\No.1.go"
Masukkan data lingkaran:
Lingkaran 1 (cx, cy, r): 1 1 5
Lingkaran 2 (cx, cy, r): 8 8 4
Masukkan koordinat titik (x, y):
15 20
Titik di luar lingkaran 1 dan 2
PS C:\Users\ACER\Documents\golang>

```

## **PENJELASAN PROGRAM**

*Program tersebut adalah program untuk mengecek apakah sebuah titik berada di dalam salah satu atau kedua lingkaran. Didalam program tersebut terdapat Fungsi distance untuk menghitung jarak antara dua titik  $(a, b)$  dan  $(c, d)$  menggunakan rumus jarak Euclidean, kemudian ada fungsi *IsInside* untuk mengecek apakah titik  $(x, y)$  berada di dalam lingkaran dengan pusat  $(cx, cy)$  dan jari-jari  $r$ . Jika jarak dari pusat ke titik tersebut kurang dari atau sama dengan jari-jari lingkaran, maka fungsi mengembalikan *true* (titik berada di dalam lingkaran). Pada fungsi utama program meminta input untuk dua lingkaran  $(cx, cy, r)$  untuk masing-masing lingkaran. Kemudian, program meminta input koordinat titik  $(x, y)$ . Program mengecek apakah titik tersebut berada di dalam salah satu atau kedua lingkaran dengan memanggil fungsi *isInside*. Jika titik berada di dalam lingkaran, program akan mencetak bahwa titik berada di dalam lingkaran yang bersangkutan. Jika tidak berada di salah satu lingkaran, akan mencetak bahwa titik berada di luar kedua lingkaran.*