

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 3**

**FUNGSI**



Oleh:

**FAHRUR RIZQI**

2311102059

S1IF-11-02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO 2024**

## I. DASAR TEORI

### 1. Definisi Fungsi

Fungsi adalah satu kesatuan rangkaian instruksi yang memberikan atau menghasilkan suatu nilai, biasanya memetakan input ke suatu nilai yang lain. Dalam konteks pemrograman, fungsi berfungsi untuk mengorganisir kode, memudahkan pemeliharaan, dan meningkatkan keterbacaan. Fungsi dapat digunakan dalam berbagai cara, seperti:

- Menugaskan nilai ke suatu variabel
- Menjadi bagian dari ekspresi
- Menjadi argumen dari suatu subprogram.

### 2. Deklarasi Fungsi

Deklarasi fungsi mirip dengan prosedur, dan biasanya terletak pada blok yang terpisah dari program utama. Notasi algoritma untuk mendeklarasikan fungsi adalah sebagai berikut:

Dalam bahasa Go, notasi tersebut ditulis sebagai:

```
func <nama function> (<params>) <type> {  
    // deklarasi variabel lokal dari fungsi  
    // badan algoritma fungsi  
    return <value/variabel>  
}
```

Pada bagian deklarasi, tipe data dari nilai yang dikembalikan harus dinyatakan, dan dalam badan fungsi, nilai tersebut harus dikembalikan menggunakan kata kunci return.

### 3. Cara Pemanggilan Fungsi

Pemanggilan fungsi dilakukan dengan menuliskan nama fungsi beserta argumen yang diminta oleh parameter dari fungsi. Fungsi dapat di-assign ke suatu variabel, menjadi bagian dari ekspresi, dan juga dapat langsung ditampilkan dengan perintah output atau print. Contoh pemanggilan fungsi dalam pseudocode dan GoLang adalah sebagai berikut:

```
v1 = volumeTabung(r, t) // cara pemanggilan #1  
v2 = volumeTabung(r, t) + volumeTabung(15, t) // cara pemanggilan #2  
fmt.Println(volumeTabung(14, 18)) // cara pemanggilan #3
```

Di sini, terlihat bahwa fungsi dapat digunakan dalam berbagai konteks, baik sebagai nilai yang diassign ke variabel maupun sebagai bagian dari ekspresi.

#### 4. Contoh Program dengan Fungsi

Salah satu contoh penggunaan fungsi adalah untuk menghitung nilai faktorial dan permutasi. Berikut adalah contoh penulisan fungsi dalam program:

```
func faktorial(n int) int {  
    var hasil int = 1  
    for i := 1; i <= n; i++ {  
        hasil = hasil * i  
    }  
    return hasil  
}  
  
func permutasi(n, r int) int {  
    return faktorial(n) / faktorial(n - r)  
}
```

Dalam contoh di atas, fungsi faktorial dipanggil secara tidak langsung melalui fungsi permutasi, menunjukkan bagaimana fungsi dapat saling berinteraksi.

#### Kesimpulan

Fungsi adalah elemen penting dalam pemrograman yang memungkinkan pengorganisasian kode dan pengulangan logika. Dengan memahami cara mendeklarasikan, memanggil, dan menggunakan fungsi, programmer dapat menulis kode yang lebih efisien dan mudah dipelihara.

## II. GUIDED

### 1. 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))

}
```

## Output

```
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go run
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3\Fahrur Rizqi_2311102059\tempCodeRunnerFil
4
8
1680
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> |
```

## Deskripsi Program

Program di atas menghitung permutasi dari dua angka input `a` dan `b`. Program membaca dua angka, membandingkan nilai keduanya, lalu menghitung permutasi dengan rumus:

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

Fungsi `faktorial(n)` menghitung faktorial dari `n`, sedangkan fungsi `permutasi(n, r)` menggunakan faktorial untuk menghitung permutasi. Hasil permutasi ditampilkan berdasarkan nilai terbesar dari dua input.

## Guided 2

```
package main

import "fmt"

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

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

```

}

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

func main() {
    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)

    if a >= c && b >= d {
        permutasiAC := permutation(a, c)
        kombinasiAC := combination(a, c)

        permutasiBD := permutation(b, d)
        kombinasiBD := combination(b, d)

        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.")
    }
}

```

## Output

```

PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go run
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3\Fahrur Rizqi_2311102059\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 C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> █

```

## Deskripsi Program

Pengguna memasukkan 4 bilangan a, b, c, d. Jika syarat  $a \geq c$  dan  $b \geq d$  terpenuhi, program akan menghitung permutasi dan kombinasi dari kedua pasangan angka. Jika tidak, akan ditampilkan pesan bahwa syarat tidak terpenuhi. Fungsi factorial digunakan untuk menghitung faktorial dari sebuah bilangan n. Faktorial adalah perkalian dari semua bilangan bulat positif kurang dari atau sama dengan n. Fungsi permutation digunakan untuk menghitung **permutasi** menggunakan rumus:  $nPr = n! / (n-r)!$ . Permutasi menghitung urutan dari objek yang dipilih dari total objek yang ada. Dan fungsi combination digunakan untuk menghitung kombinasi menggunakan rumus:  $nCr = n! / (r!(n-r)!)$ . Kombinasi menghitung cara memilih objek tanpa memperhatikan urutan.

## III. UNGUIDED

### Unguided

```
package main

import (
    "fmt"
    "math" )

type Circle struct {
    cx, cy, r float64
}

func isPointInside(c Circle, x, y float64) bool {
    distance := math.Sqrt(math.Pow(x-c.cx, 2) + math.Pow(y-
c.cy, 2))
    return distance <= c.r
}

func determinePosition(c1, c2 Circle, x, y float64) string {
    in1 := isPointInside(c1, x, y)
    in2 := isPointInside(c2, x, y)
```

```

        if in1 && in2 {
            return "Titik di dalam lingkaran 1 dan 2"
        } else if in1 {
            return "Titik di dalam lingkaran 1"
        } else if in2 {
            return "Titik di dalam lingkaran 2"
        }
        return "Titik di luar lingkaran 1 dan 2"
    }

func main() {
    var c1, c2 Circle
    var x, y float64

    fmt.Println("Masukkan koordinat pusat dan radius lingkaran
1 (cx cy r):")
    fmt.Scan(&c1.cx, &c1.cy, &c1.r)

    fmt.Println("Masukkan koordinat pusat dan radius lingkaran
2 (cx cy r):")
    fmt.Scan(&c2.cx, &c2.cy, &c2.r)

    fmt.Println("Masukkan koordinat titik yang akan dicek (x
y):")
    fmt.Scan(&x, &y)

    result := determinePosition(c1, c2, x, y)
    fmt.Println(result)
}

```



## Output

```
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3\Fahrur Rizqi_2311102059\unguided1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r):
1 1 5
Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r):
8 8 4
Masukkan koordinat titik yang akan dicek (x y):
2 2
Titik di dalam lingkaran 1
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3\Fahrur Rizqi_2311102059\unguided1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r):
1 2 3
Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r):
4 5 6
Masukkan koordinat titik yang akan dicek (x y):
7 8
Titik di dalam lingkaran 2
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> █
```

## Deskripsi Program

Pengguna diminta untuk memasukkan data dua lingkaran berupa koordinat pusat (cx, cy) dan jari-jari (r), serta memasukkan koordinat sebuah titik (x, y). Fungsi jarak digunakan untuk menghitung jarak antara dua titik dalam bidang kartesian menggunakan rumus jarak Euclidean. Fungsi `di_dalam` mengecek apakah jarak dari pusat lingkaran ke titik yang diberikan kurang dari atau sama dengan jari-jari lingkaran. Jika ya, maka titik tersebut berada di dalam lingkaran. Program akan memeriksa apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, dan hasilnya dicetak. Jika titik tidak berada di dalam kedua lingkaran, program menampilkan pesan bahwa titik berada di luar lingkaran.