

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

FUNGSI



Oleh:

NAUFAL THORIQ MUZHAFAR

2311102078

IF-11-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

Fungsi adalah blok pernyataan yang dapat digunakan berulang kali dalam suatu program. Fungsi tidak akan dijalankan secara otomatis saat halaman dimuat. Fungsi akan dijalankan dengan memanggil fungsi tersebut.

Deklarasi Fungsi

Untuk membuat (sering disebut sebagai mendeklarasikan) suatu fungsi, lakukan hal berikut:

- Gunakan kata kunci **func**.
- Tentukan nama untuk fungsi tersebut, diikuti tanda kurung ().
- Tambahkan kode yang menentukan apa yang harus dilakukan fungsi tersebut, di dalam kurung kurawal {}.
- Mengembalikan suatu nilai dengan kunci **return**.

Jika Anda ingin fungsi tersebut mengembalikan suatu nilai, Anda perlu menentukan tipe data dari nilai yang dikembalikan (seperti int, string, dll), dan juga menggunakan kata kunci di dalam fungsi tersebut:

Sintaks

```
func NamaFungsi() {  
    // code yang akan dieksekusi  
  
    return nilai  
}
```

Aturan Penamaan untuk Fungsi

- Nama fungsi harus dimulai dengan huruf
- Nama fungsi hanya boleh berisi karakter alfanumerik dan garis bawah (A-z, 0-9, dan _)
- Nama fungsi peka huruf besar-kecil
- Nama fungsi tidak boleh berisi spasi
- Jika nama fungsi terdiri dari beberapa kata, teknik yang diperkenalkan untuk penamaan variabel multikata dapat digunakan

Parameter dan Argumen

Informasi dapat diteruskan ke fungsi sebagai parameter. Parameter bertindak sebagai variabel di dalam fungsi. Parameter dan tipenya ditentukan setelah nama fungsi, di dalam tanda kurung. Anda dapat menambahkan parameter sebanyak yang diinginkan, cukup pisahkan dengan koma:

Sintaks

```
func NamaFungsi(param1 type, param2 type, param3 type)
{
    // code yang akan dieksekusi

    return nilai
}
```

Memanggil Fungsi

Fungsi tidak langsung dieksekusi. Fungsi tersebut "disimpan untuk penggunaan nanti", dan akan dieksekusi saat dipanggil.

Dalam contoh di bawah ini, kita membuat fungsi bernama "penjumlahan()". Tanda kurung kurawal pembuka ({) menunjukkan awal kode fungsi, dan tanda kurung kurawal penutup (}) menunjukkan akhir fungsi. Kunci **return** pada akhir fungsi bertujuan untuk mengembalikan nilai pada tipe data. Fungsi tersebut akan menampilkan hasil dari pengembalian nilai fungsi penjumlahan. Untuk memanggil fungsi, cukup tulis namanya diikuti dua tanda kurung ():9

Contoh

```
package main
import ("fmt")

func penjumlahan(x int, y int) int {
    return x + y
}

func main() {
    fmt.Println(penjumlahan(1, 2))
}
```

Nilai Return Bernama

Dalam Go, Anda dapat memberi nama nilai kembalian suatu fungsi.

Contoh

Di sini, kita menamai nilai pengembalian sebagai **result**(bertipe int), dan mengembalikan nilai tersebut dengan pengembalian polos (artinya kita menggunakan pernyataan **return** tersebut tanpa menentukan nama variabel):

```
package main
import ("fmt")

func myFunction(x int, y int) (result int) {
    result = x + y
    return
}

func main() {
    fmt.Println(myFunction(1, 2))
}
```

Beberapa Nilai Pengembalian

Fungsi Go juga dapat mengembalikan beberapa nilai.

Contoh

Di sini, myFunction() mengembalikan satu integer (**result**) dan satu string (**txt1**):

```
package main
import ("fmt")

func myFunction(x int, y string) (result int, txt1
string) {
    result = x + x
    txt1 = y + " World!"
    return
}

func main() {
    fmt.Println(myFunction(5, "Hello"))
}
```

Jika kita (karena alasan tertentu) tidak ingin menggunakan beberapa nilai yang dikembalikan, kita dapat menambahkan garis bawah (_), untuk menghilangkan nilai ini.

Contoh

Di sini, kita ingin menghilangkan nilai yang dikembalikan pertama (result- yang disimpan dalam variabel a):

```
package main
import ("fmt")

func myFunction(x int, y string) (result int, txt1
string) {
    result = x + x
    txt1 = y + " World!"
    return
}

func main() {
    _, b := myFunction(5, "Hello")
    fmt.Println(b)
}
```

II. GUIDED

GUIDED 1

Source Code

```
package main

import "fmt"

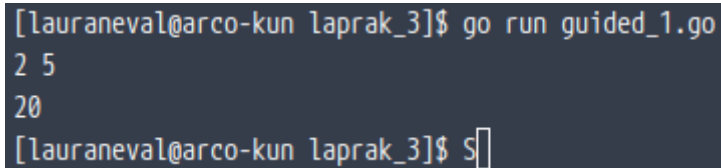
func main() {
    var a, b int
    fmt.Scan(&a, &b)
    if a >= b {
        fmt.Println(permutasi(a, b))
    } else {
        fmt.Println(permutasi(b, a))
    }
}

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



```
[lauraneval@arco-kun laprak_3]$ go run guided_1.go  
2 5  
20  
[lauraneval@arco-kun laprak_3]$ S
```

Penjelasan Program

Jika pengguna memasukkan dua angka a dan b, program akan menentukan permutasi dengan memanggil fungsi permutasi. Jika $a \geq b$, maka akan menghitung permutasi dari a dan b (nPr). Jika tidak, akan menghitung permutasi dari b dan a. Fungsi factorial digunakan untuk menghitung faktorial dari bilangan n. Faktorial adalah hasil perkalian dari semua bilangan bulat positif yang kurang dari atau sama dengan n. Sedangkan fungsi permutasi digunakan untuk menghitung nilai permutasi menggunakan rumus $nPr = n! / (n-r)!$, di mana n adalah jumlah total item, dan r adalah jumlah item yang dipilih.

GUIDED 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? (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!

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

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.")

    }

}

```

Screenshoot Output

```

[lauraneval@arco-kun laprak_3]$ go run guided_2.go
Masukkan bilangan a, b, c, d (dengan spasi):
5 10 3 10
Permutasi(a, c) dan Kombinasi(a, c): 60 10
Permutasi(b, d) dan Kombinasi(b, d): 3628800 1
[lauraneval@arco-kun laprak_3]$ S

```

Penjelasan 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 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	Titik di dalam lingkaran 1

	8 8 4 2 2	
2	1 2 3 4 5 6 7 8	Titik di dalam lingkaran 2
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

Fungsi untuk menghitung jarak titik (a, b) dan (c, d) dimana rumus jarak adalah:

$$\text{jarak} = \sqrt{(a - c)^2 + (b - d)^2}$$

dan juga fungsi untuk menentukan posisi sebuah titik sembarang berada di dalam suatu lingkaran atau tidak.

```
function jarak(a,b,c,d : real) -> real
{Mengembalikan jarak antara titik (a,b) dan titik (c,d)}

function didalam(cx,cy,r,x,y : real) -> boolean
{Mengembalikan true apabila titik (x,y) berada di dalam lingkaran yang
memiliki titik pusat (cx,cy) dan radius r}
```

Catatan: Lihat paket **math** dalam lampiran untuk menggunakan fungsi **math.Sqrt()** untuk menghitung akar kuadrat.

Source Code

```
package main

import (
    "fmt"
    "math"
)

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

func di_dalam(cx, cy, r, x, y float64) bool {
    return jarak(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.Print("Masukkan koordinat titik (x, y): ")

fmt.Scan(&x, &y)


dalam_Lingkaran := [2]bool{}

for i := 0; i < 2; i++ {

    dalam_Lingkaran[i] =
di_dalam(circles[i].cx, circles[i].cy, circles[i].r,
x, y)

}


result := ""

for i := 0; i < 2; i++ {

    if dalam_Lingkaran[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

```
[lauraneval@arco-kun laprak_3]$ go run unguided_1.go
Masukkan data lingkaran:
Lingkaran 1 (cx, cy, r): 5 10 15
Lingkaran 2 (cx, cy, r): -15 4 20
Masukkan koordinat titik (x, y): 0 0
Titik di dalam lingkaran 1 Titik di dalam lingkaran 2
[lauraneval@arco-kun laprak_3]$ go run unguided_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): 2 2
Titik di dalam lingkaran 1
[lauraneval@arco-kun laprak_3]$ SS
```

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.