

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL III**

**FUNGSI**



Oleh:

ALDEN AUDY AKBAR

2311102309

IF-11-07

**S1 TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

---

**2024**

## I. DASAR TEORI

### 3.1 Definisi Function

Fungsi merupakan 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, ...

### 3.2 Deklarasi Function

Deklarasi fungsi sama dengan prosedur yaitu berada pada blok yang terpisah dengan program utama.

INFORMATIKA

	Notasi Algoritma
1	function <nama function> (<params>)      <type>
2	kamus
3	{deklarasi variabel lokal dari fungsi}
4	
5	algoritma
6	{badan algoritma fungsi}
7	
8	return <value/variabel>
9	endfunction
	Notasi dalam bahasa Go

10	func <nama function> (<params>) <type> {
11	/* deklarasi variabel lokal dari fungsi */
12	/* badan algoritma fungsi*/
13	return <value/variabel>
14	
15	
16	
17	

## Modul

Pada bagian deklarasi terlihat setelah parameter terdapat tipe data dari nilai yang dikembalikan, sedangkan pada bagian badan fungsi terdapat return dari nilai yang dikembalikan.

Berikut adalah contoh fungsi untuk menghitung volume dari tabung apabila jari-jari alas dan tinggi tabung diketahui.

	Notasi Algoritma
1	function volumeTabung(jari_jari, tinggi integer)      real
2	kamus luasA1as, volume: real algoritma luasA1as
3	<- 3.14 * (jari_jari * jari_jari) volume <- luasA1as *
4	tinggi return volume endfunction
5	
6	
7	
8	
	Notasi dalam bahasa Go
10	func volumeTabung(jari_jari, tinggi int) float64 { var
11	luasA1as, volume float64 luasA1as = 3.14 *
12	float64(jari_jari * jari_jari) volume = luasA1as *
13	tinggi return volume
14	
15	

### 3.3 Cara Pemanggilan Function

Sama halnya dengan prosedury pemanggilan fungsi cukup dilakukan dengan penulisan nama fungsi beserta argumen yang diminta oleh parameter dari fungsi. Perbedaananya dengan R+ adalah fungsi bisa di-assign ke suatu variabel, menjadi bagian dari ekspresi, dan argumen dari suatu subprogram.

	Notasi Algoritma
--	------------------

2	program	
3	ContohProsedur kamus	
4	integer	
5	algoritma	
6		
7	t 10	
8	VI <- volumeTabung(r, t) v2 <- volumeTabung(r, t)	{cara pemanggilan #1}
9	+ volumeTabung(15, t) output (volumeTabung(14,	{cara pemanggilan #2}
10	100) ) endprogram	{cara pemanggilan #3}
11		
Notasi dalam bahasa Go		
12	func main() { var r, t	
13	int var VI ,v2	
14	float64	
15		
16	- 10	
17	VI = volumeTabung(r, t)	// cara pemanggilan #1
34 1		
18	v2 - volumeTabung(r, t) + volumeTabung(15, t) fmt .	// cara pemanggilan #2
19	Print In (volumeTabung(14, 100) )	// cara pemanggilan #3
20	}	

Pada contoh pemanggilan fungsi di atas terlihat tidak ada perbedaan pada saat pemanggilan fungsi pada pseudocode ataupun GoLang. Di sini terlihat fungsi bisa di-assign ke suatu variabel pada saat pemanggilan, bisa dioperasikan sesuai dengan tipe data yang dikembalikan, dan juga bisa langsung ditampilkan dengan perintah output ataupun print.

## **II. GUIDED**

Source Code + Screenshot hasil program beserta penjelasan

### **III. UNGUIDED**

1. 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?

```

package main

//2311102309
import (
    "fmt"
    "math/big"
)

func factorial(n int64) *big.Int {
    result := big.NewInt(1)
    for i := int64(2); i <= n; i++ {
        result.Mul(result, big.NewInt(i))
    }
    return result
}

func perm(n, r int64) *big.Int {
    if r > n {
        return big.NewInt(0)
    }
    return new(big.Int).Div(factorial(n), factorial(n-r))
}

func comb(n, r int64) *big.Int {
    if r > n {
        return big.NewInt(0)
    }

    return new(big.Int).Div(factorial(n), new(big.Int).Mul(factorial(r),
        factorial(n-r)))
}

func main() {
    var a, b, c, d int64
    fmt.Print("Masukkan empat angka :")
    fmt.Scan(&a, &b, &c, &d)
    P_ac := perm(a, c)
    C_ac := comb(a, c)
    P_bd := perm(b, d)
    C_bd := comb(b, d)
    fmt.Printf("Permutasi (%d, %d) = %d\n", a, c, P_ac)
    fmt.Printf("Kombinasi (%d, %d) = %d\n", a, c, C_ac)
    fmt.Printf("Permutasi (%d, %d) = %d\n", b, d, P_bd)
}

```

```
    fmt.Printf("Kombinasi (%d, %d) = %d\n", b, d, C_bd)
}
```

```
Masukkan empat angka :5 10 3 10
Permutasi (5, 3) = 60
Kombinasi (5, 3) = 10
Permutasi (10, 10) = 3628800
Kombinasi (10, 10) = 1
PS C:\Users\alden\OneDrive\Documents\GOLANG>
```

Penjelasan:

Program meminta pengguna untuk memasukkan empat bilangan bulat. Empat bilangan ini akan digunakan sebagai input untuk perhitungan permutasi dan kombinasi. Program memiliki fungsi khusus untuk menghitung faktorial dari suatu bilangan. Faktorial dari sebuah bilangan adalah hasil perkalian bilangan itu dengan semua bilangan bulat positif yang lebih kecil darinya. Fungsi ini sangat penting karena rumus permutasi dan kombinasi melibatkan faktorial. Menggunakan rumus permutasi, program menghitung banyaknya cara menyusun objek-objek dalam suatu urutan tertentu.

Rumus permutasi:  $P(n, r) = n! / (n-r)!$

---

$P(n, r)$  adalah jumlah permutasi



n adalah jumlah total objek

r adalah jumlah objek yang dipilih

Hasil perhitungan permutasi kemudian disimpan. Menggunakan rumus kombinasi, program menghitung banyaknya cara memilih objek-objek tanpa memperhatikan urutannya.

Rumus kombinasi:  $C(n, r) = n! / (r! * (n-r)!)$

$C(n, r)$  adalah jumlah kombinasi

n dan r memiliki arti yang sama seperti pada permutasi

Hasil perhitungan kombinasi kemudian disimpan. Setelah semua perhitungan selesai, program akan menampilkan hasil perhitungan permutasi dan kombinasi untuk setiap pasangan bilangan yang telah dimasukkan oleh pengguna.

```
program Permutasi_Kombinasi_Dengan_Validasi
```

```
kamus
```

```
  a, b, c, d: integer
```

```
algoritma
```

```
  Tulis("Masukkan empat angka (dipisahkan spasi): ")
```

```
  Baca(a, b, c, d)
```

```
  jika r > n untuk semua (r, n) dari pasangan [(a, c), (b, d)] lakukan
```

```
    Tulis("Input tidak valid. r tidak boleh lebih besar dari n.")
```

```
    hentikan program
```

```
  akhir jika
```

```
  permutasiAC := Permutasi(a, c)
```

```
  kombinasiAC := Kombinasi(a, c)
```

```

permutasiBD := Permutasi(b, d)
kombinasiBD := Kombinasi(b, d)

Tulis("Permutasi dan Kombinasi:")
Tulis(" Permutasi (" , a, ", ", c, ") = ", permutasiAC)
Tulis(" Kombinasi (" , a, ", ", c, ") = ", kombinasiAC)
Tulis("") // Baris kosong untuk pemisah
Tulis(" Permutasi (" , b, ", ", d, ") = ", permutasiBD)
Tulis(" Kombinasi (" , b, ", ", d, ") = ", kombinasiBD)

```

akhirprogram

fungsi Permutasi(n, r: integer): big.Int

```

    jika r > n maka
        kembalikan big.NewInt(0)
    akhir jika
    kembalikan Faktorial(n) / Faktorial(n - r)

```

akhirfungsi

fungsi Kombinasi(n, r: integer): big.Int

```

    jika r > n maka
        kembalikan big.NewInt(0)
    akhir jika
    return Faktorial(n) / (Faktorial(r) * Faktorial(n - r))

```

akhirfungsi

fungsi Faktorial(n: integer): big.Int

```

    hasil := big.NewInt(1)
    untuk i := 2 hingga n lakukan
        hasil := hasil * big.NewInt(i)
    akhir untuk
    kembalikan hasil

```

akhirfungsi

2. Diberikan tiga buah fungsi matematika yaitu  $f(x) = x^2$ ,  $g(x) = x - 2$  dan  $h(x) = x + 1$ . Fungsi komposisi (fogoh) (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.

```
package main

//2311102309
import "fmt"

func f(x int) int {
    return x * x
}
func g(x int) int {
    return x - 2
}
func h(x int) int {
    return x + 1
}
func main() {
    var a, b, c int
    fmt.Print("Masukkan nilai : ")
    fmt.Scan(&a, &b, &c)

    fogoh := f(g(h(a)))
    gohof := g(h(f(b)))
    hofog := h(f(g(c)))

    fmt.Println(fogoh)
    fmt.Println(gohof)
    fmt.Println(hofog)
}
```

```
Masukkan nilai : 7 2 10
36
3
65
PS C:\Users\alden\OneDrive\Documents\GOLANG> █
```

Penjelasan: Deklarasikan variabel a, b, dan c untuk menyimpan nilai input.

Minta pengguna untuk memasukkan nilai untuk a, b, dan c.

Hitung fogoh =  $f(g(h(a)))$  dengan memanggil fungsi f, g, dan h secara berurutan.

Hitung gohof =  $g(h(f(b)))$  dengan memanggil fungsi g, h, dan f secara berurutan.

Hitung hofog =  $h(f(g(c)))$  dengan memanggil fungsi h, f, dan g secara berurutan.

Tampilkan hasil: Cetak nilai fogoh, gohof, dan hofog ke layar.

f(x): Fungsi kuadrat, mengembalikan nilai  $x^2$ .

g(x): Fungsi linear, mengembalikan nilai  $x - 2$ .

h(x): Fungsi linear, mengembalikan nilai  $x + 1$ .

Jika pengguna memasukkan nilai  $a = 3$ ,  $b = 4$ , dan  $c = 5$ , maka program akan menghitung:

$$\text{fogoh} = f(g(h(3))) = f(g(4)) = f(2) = 4$$

$$\text{gohof} = g(h(f(4))) = g(h(16)) = g(17) = 15$$

$$\text{hofog} = h(f(g(5))) = h(f(3)) = h(9) = 10$$

Kemudian, program akan mencetak hasil perhitungan tersebut ke layar.

3. Lingkaran] Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx;cy) dengan radius. 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.

```
package main

//2311102309
import (
    "fmt"
    "math"
)

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

func posisiTitik(cx1, cy1, r1, cx2, cy2, r2, x, y float64) string {
    d1 := hitungJarak(x, y, cx1, cy1)
    d2 := hitungJarak(x, y, cx2, cy2)
    if d1 < r1 && d2 < r2 {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if d1 < r1 {
        return "Titik di dalam lingkaran 1"
    } else if d2 < r2 {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, x, y float64
    fmt.Print("Masukkan koordinat titik pusat lingkaran 1 : ")
    fmt.Scan(&cx1, &cy1)

    fmt.Print("Masukkan radius lingkaran 1: ")
    fmt.Scan(&r1)
```

```

fmt.Print("Masukkan koordinat titik pusat lingkaran 2 : ")
fmt.Scan(&cx2, &cy2)

fmt.Print("Masukkan radius lingkaran 2: ")
fmt.Scan(&r2)

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

hasil := posisiTitik(cx1, cy1, r1, cx2, cy2, r2, x, y)
fmt.Println(hasil)
}

```

```

Masukkan koordinat titik pusat lingkaran 1 : 1 1 5
Masukkan radius lingkaran 1: Masukkan koordinat titik pusat lingkaran 2 : 8 8 4
Masukkan radius lingkaran 2: Masukkan koordinat titik : 2 2
Titik di dalam lingkaran 1
PS C:\Users\alden\OneDrive\Documents\GOLANG>

```

Penjelasan: Program mendeklarasikan beberapa variabel bertipe float64 untuk menyimpan koordinat titik pusat, jari-jari kedua lingkaran, dan koordinat titik yang ingin diperiksa.

Program meminta pengguna untuk memasukkan nilai-nilai numerik untuk variabel-variabel tersebut melalui input

Fungsi hitungJarak digunakan untuk menghitung jarak antara dua titik dalam koordinat kartesian. Rumus yang digunakan adalah rumus jarak Euclidean. Jarak ini kemudian disimpan dalam variabel d1 dan d2 yang mewakili jarak titik yang diberikan ke titik pusat lingkaran 1 dan 2.

Fungsi posisiTitik menggunakan hasil perhitungan jarak untuk menentukan posisi titik tersebut.

Jika jarak titik ke kedua titik pusat lebih kecil dari jari-jari masing-masing lingkaran, maka titik tersebut berada di dalam kedua lingkaran.

Jika jarak titik ke salah satu titik pusat lebih kecil dari jari-jari lingkaran tersebut, maka titik tersebut berada di dalam lingkaran yang bersangkutan.

Jika jarak titik ke kedua titik pusat lebih besar dari jari-jari masing-masing lingkaran, maka titik tersebut berada di luar kedua lingkaran.

Program mencetak hasil berupa string yang menjelaskan posisi titik tersebut relatif terhadap kedua lingkaran.

```
program_Hitung_Jarak_Dan_Posisi_Titik

kamus
  a, b, c, d, x, y, cx, cy, r: bilangan real
  jarak_ab_cd, dalam_lingkaran: boolean

algoritma
  a <- 1.0
  b <- 2.0
  c <- 3.0
  d <- 4.0
  cx <- 0.0
  cy <- 0.0
  r <- 5.0

  jarak_ab_cd <- HitungJarak(a, b, c, d)
  Tulis("Jarak antara titik (" , a , "," , b , ") dan (" , c , "," , d , ") adalah:",
  jarak_ab_cd)

  dalam_lingkaran <- DidalamLingkaran(a, b, cx, cy, r)
  jika dalam_lingkaran maka
    Tulis("Titik (" , a , "," , b , ") berada di dalam lingkaran")
  lain
    Tulis("Titik (" , a , "," , b , ") berada di luar lingkaran")
  akhir jika

akhirprogram

fungsi HitungJarak(a, b, c, d: bilangan real): bilangan real
  kembalikan akarKuadrat((a-c)^2 + (b-d)^2)
```

akhirfungsi

fungsi DidalamLingkaran(x, y, cx, cy, r: bilangan real): boolean

  jika HitungJarak(x, y, cx, cy) <= r maka

    kembalikan benar

  lain

    kembalikan salah

  akhir jika

akhirfungsi



