

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL III
PENGENALAN FUNGSI**



Oleh:

Destia Ananda Putra

2311102176

IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

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 variable
- 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**

Deklarasi Function

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

	Notasi Algoritma
1	function function> (<params>) -> <type>
2	kamus
3	{deklarasi
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	...
13	/* badan algoritma fungsi*/
14	...
15	return <value/variabel>
16	
17	}

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

Cara Pemangggllan Function

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

Notasi Algoritma	
1	program ContohProsedur
2	kamus
3	r,t : integer
4	v1,v2 : real
5	algoritma
6	r <- 5;
7	t <- 10
8	v1 <- volumeTabung(r,t) {cara pemanggilan #1}
9	v2 <- volumeTabung(r,t) + volumeTabung(15,t) {cara pemanggilan #2}
10	output(volumeTabung(14,100)) {cara pemanggilan #3}
11	endprogram
Notasi dalam bahasa Go	
12	func main() {
13	var r,t int
14	var v1,v2 float64
15	r = 5
16	t = 10
17	v1 = volumeTabung(r,t) // cara pemanggilan #1
18	v2 = volumeTabung(r,t) + volumeTabung(15,t) // cara pemanggilan #2
19	fmt.Println(volumeTabung(14,100)) // cara pemanggilan #3
20	}

II. GUIDED

1. Guided

Source Code

```
package main

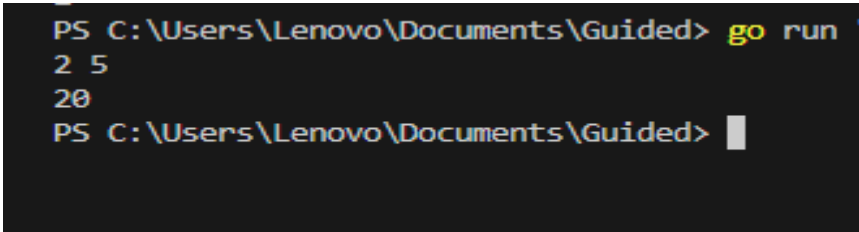
import "fmt"

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))
}
```

Output



```
PS C:\Users\Lenovo\Documents\Guided> go run
2 5
20
PS C:\Users\Lenovo\Documents\Guided>
```

Penjelasan :

Pada Output dimulai dengan membaca dua input angka dari pengguna, a dan b. Kemudian, program memeriksa apakah a lebih besar atau sama dengan b. Jika ya, maka program menghitung permutasi $P(a, b)$, jika tidak, dihitung permutasi $P(b, a)$.

dihitung menggunakan fungsi $\text{permutasi}(n, r)$, yang memanfaatkan fungsi $\text{faktorial}(n)$ untuk menghitung nilai faktorial dari n. faktorial digunakan untuk menyusun elemen secara berurutan. Program akhirnya mengeluarkan hasil perhitungan tersebut.

2. Guided 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.")
    }
}

```

Output

```

PS C:\Users\Lenovo\Documents\Guided> go run "c:
Masukkan bilangan a, b, c, d (dengan spasi):
7 6 5 4
Permutasi(a, c) dan Kombinasi(a, c): 2520 21
Permutasi(b, d) dan Kombinasi(b, d): 360 15
PS C:\Users\Lenovo\Documents\Guided> go run "c:
Masukkan bilangan a, b, c, d (dengan spasi):
1 2 3 4
Syarat a >= c dan b >= d tidak terpenuhi.

```

Penjelasan :

Program ini dengan menerima input empat bilangan: a, b, c, dan d. Setelah itu, program memeriksa apakah dua syarat terpenuhi, yaitu $a \geq c$ dan $b \geq d$. Jika syarat tersebut benar, program akan menghitung nilai permutasi dan kombinasi untuk pasangan (a, c) dan (b, d).

Hasil dari kedua perhitungan tersebut kemudian ditampilkan sebagai output. Namun, jika salah satu atau kedua syarat tidak terpenuhi, program akan menampilkan pesan bahwa syarat tersebut tidak dipenuhi, tanpa melakukan perhitungan apa pun.

III. UNGUIDED

1. 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.

```
package main

// Destia Ananda Putra
// 23111021761

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 n int
    fmt.Print("Masukkan jumlah lingkaran: ")
    fmt.Scan(&n)

    // Menggunakan slice untuk menyimpan data lingkaran
    circles := make([]struct {
        cx, cy, r float64
    }, n)
```



```

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

u   // Input koordinat titik
    var x, y float64
l   fmt.Print("Masukkan koordinat titik (x, y): ")
i   fmt.Scan(&x, &y)

n   // Mengecek apakah titik berada di dalam masing-masing
    lingkaran
g   dalam_Lingkaran := make([]bool, n)
k   for i := 0; i < n; i++ {
a       dalam_Lingkaran[i] = di_dalam(circles[i].cx,
r       circles[i].cy, circles[i].r, x, y)
    }

a   // Menampilkan hasil
n   result := ""
    for i := 0; i < n; i++ {
        if dalam_Lingkaran[i] {
            result += fmt.Sprintf("Titik di dalam lingkaran
%d ", i+1)
        }
    }

    if result == "" {
        fmt.Println("Titik di luar semua lingkaran")
    } else {
        fmt.Println(result)
    }
}

```

```

PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Guided\ed1.go"
Masukkan jumlah lingkaran: 2
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
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Guided\ed1.go"
Masukkan jumlah lingkaran: 2
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

```

Penjelasan :

Program ini akan memberitahukan pengguna tentang posisi titik yang dimasukkan dibandingkan dengan lingkaran-lingkaran yang telah didefinisikan. Jika titik berada di dalam satu atau lebih lingkaran, program akan mencetak pesan yang menunjukkan nomor lingkaran mana saja yang mengandung titik tersebut. Misalnya, jika titik berada di dalam lingkaran pertama dan ketiga, outputnya akan menyebutkan "Titik di dalam lingkaran 1" dan "Titik di dalam lingkaran 3". Jika titik tidak berada di dalam lingkaran mana pun, program akan mencetak pesan "Titik di luar semua lingkaran".