

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

FUNGSI



**Universitas
Telkom**

Oleh:

HENDWI SAPUTRA

2311102218

IF-11-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY 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 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**, ...

3.2 Deklarasi Function

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

	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	...
13	/* badan algoritma fungsi*/
14	...
15	return <value/variabel>
16	}
17	

Pada bagian deklarasi terlihat setelah parameter terdapat tipe data dan 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
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) float64 {
11	var luasAlas,volume float64
12	luasAlas = 3.14 * float64(jari_jari * jari_jari)
13	volume = luasAlas * tinggi
14	return volume
15	}

3.3 Cara Pemanggilan Function

Sama halnya dengan prosedur cukup dilakukan dengan penulisan nama fungsi beserta argumen yang diminta oleh parameter dari fungsi. Perbedaannya dengan R+ 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	}

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

GUIDED I

Source Code

```
package main

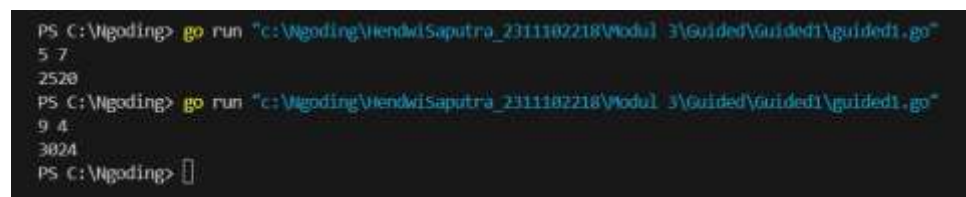
import "fmt"

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



```
PS C:\Vgoding> go run "c:\Vgoding\ViendwiSaputra_2311102218\Modul 3\Guided\Guided1\guided1.go"
5 7
2520
PS C:\Vgoding> go run "c:\Vgoding\ViendwiSaputra_2311102218\Modul 3\Guided\Guided1\guided1.go"
9 4
3024
PS C:\Vgoding> 
```

Deskripsi:

Program ini adalah aplikasi sederhana yang dapat menghitung permutasi dua bilangan. Pengguna diminta untuk memasukkan dua bilangan bulat, a dan b. Program kemudian menentukan bilangan yang lebih besar antara keduanya dan menghitung permutasi dengan memanggil fungsi permutasi. Fungsi faktorial digunakan untuk menghitung faktorial dari suatu bilangan,

dan hasilnya digunakan dalam fungsi permutasi untuk menghitung permutasi nPr , yaitu faktorial n dibagi dengan faktorial $n-r$. Hasil perhitungan permutasi ini kemudian dicetak ke layar.

GUIDED II

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)

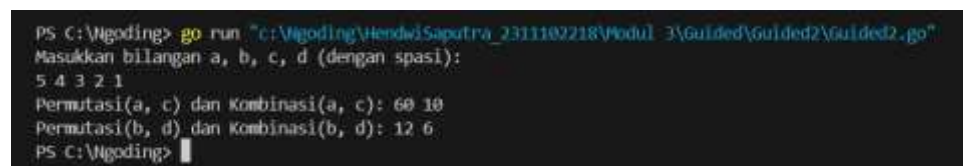
        // Menghitung permutasi dan kombinasi b dan d
        permutasiBD := permutation(b, d)
        kombinasiBD := combination(b, d)
    }
}
```

```

        // 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



```

PS C:\Vgoding> go run "c:\Vgoding\HendriSaputra_231102218\Modul 3\Guided\Guided2\Guided2.go"
Masukkan bilangan a, b, c, d (dengan spasi):
5 4 3 2 1
Permutasi(a, c) dan Kombinasi(a, c): 60 10
Permutasi(b, d) dan Kombinasi(b, d): 12 6
PS C:\Vgoding>

```

Deskripsi:

Program ini adalah program yang menghitung permutasi dan kombinasi dari dua pasang bilangan yang dimasukkan oleh pengguna. Pertama, pengguna diminta untuk memasukkan empat bilangan bulat: a, b, c, dan d. Program kemudian memeriksa apakah nilai a lebih besar atau sama dengan c, dan apakah b lebih besar atau sama dengan d. Jika kedua syarat tersebut terpenuhi, program menghitung permutasi dan kombinasi untuk pasangan (a, c) dan (b, d) menggunakan fungsi permutation dan combination yang bergantung pada hasil perhitungan faktorial dari fungsi factorial. Hasil perhitungan tersebut kemudian ditampilkan. Jika syarat tidak terpenuhi, program akan memberikan pesan bahwa syarat tidak terpenuhi.

III. UNGUIDED

UNGUIDED I

Source Code

```
package main

import (
    "fmt"
    "math"
)

func jarak(a, b, c, d float64) float64 {
    return math.Sqrt(math.Pow(a-c, 2) + math.Pow(b-d, 2))
}

func diDalam(cx, cy, r, x, y float64) bool {
    return jarak(cx, cy, x, y) <= r
}

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, x, y float64

    fmt.Print("Lingkaran 1: ")
    fmt.Scan(&cx1, &cy1, &r1)

    fmt.Print("Lingkaran 2: ")
    fmt.Scan(&cx2, &cy2, &r2)

    fmt.Print("Titik sembarang: ")
    fmt.Scan(&x, &y)

    insideLingkaran1 := diDalam(cx1, cy1, r1, x, y)
    insideLingkaran2 := diDalam(cx2, cy2, r2, x, y)

    if insideLingkaran1 && insideLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if insideLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if insideLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```


Screenshot

```
PS C:\Ngoding> go run "c:\Ngoding\HendwiSaputra_2311102218\Modul_3\Unguided\Unguided3\Ungu
guided3.go"
Lingkaran 1: 1 1 5
Titik sembarang: 2 2
Titik di dalam lingkaran 1
PS C:\Ngoding> go run "c:\Ngoding\HendwiSaputra_2311102218\Modul_3\Unguided\Unguided3\Un
guided3.go"
Lingkaran 1: 1 2 3
Titik sembarang: 7 8
Titik di dalam lingkaran 2
PS C:\Ngoding> go run "c:\Ngoding\HendwiSaputra_2311102218\Modul_3\Unguided\Unguided3\Un
guided3.go"
Lingkaran 1: 5 10 15
Lingkaran 2: -15 4 20
Titik sembarang: 0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Ngoding> go run "c:\Ngoding\HendwiSaputra_2311102218\Modul_3\Unguided\Unguided3\Ungu
guided3.go"
Lingkaran 1: 1 1 5
Lingkaran 2: 8 8 4
Titik sembarang: 15 20
Titik di luar lingkaran 1 dan 2
PS C:\Ngoding>
```

Deskripsi:

Program ini merupakan program untuk menentukan apakah sebuah titik berada di dalam satu atau dua lingkaran, atau di luar keduanya. Program dimulai dengan mendefinisikan dua fungsi: jarak, yang menghitung jarak antara dua titik menggunakan rumus Euclidean, dan diDalam, yang memeriksa apakah suatu titik berada di dalam lingkaran berdasarkan jarak dari pusat lingkaran ke titik tersebut. Pengguna diminta untuk memasukkan koordinat pusat dan jari-jari dua lingkaran, serta koordinat titik sembarang. Program kemudian menghitung apakah titik tersebut berada di dalam salah satu atau kedua lingkaran dengan memanggil fungsi-fungsi yang telah didefinisikan. Berdasarkan hasil perhitungan, program mencetak apakah titik tersebut berada di dalam lingkaran pertama, lingkaran kedua, kedua lingkaran, atau di luar keduanya.

