

**LAPORAN PRAKTIKUM  
ALGORITME DAN PEMEROGRAMAN**

**MODUL 3**

**FUNGSI**



Oleh:

ERVAN HAPIZ

2311102206

IF – 11- 02

**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.

	Notasi Algoritma
1	function <nama function> (<params>) -> <type>
2	kamus
3	{deklarasi
4	...                                      dari
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) -> 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) 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, 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) <span style="float: right;">{cara pemanggilan #1}</span>
9	v2 <- volumeTabung(r,t) + volumeTabung(15,t) <span style="float: right;">{cara pemanggilan #2}</span>
10	output(volumeTabung(14,100)) <span style="float: right;">{cara pemanggilan #3}</span>
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) <span style="float: right;">// cara pemanggilan #1</span>

18	<code>v2 = volumeTabung(r,t) + volumeTabung(15,t)</code>	<code>// cara pemanggilan #2</code>
19	<code>fmt.Println(volumeTabung(14,100))</code>	<code>// cara pemanggilan</code>
20	<code>}</code>	

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.

### 3.4 Contoh Program dengan Function

Berikut ini adalah contoh penulisan fungsi pada suatu program lengkap.

Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilai faktorial dan permutasi.

**Masukan** terdiri dari dua buah bilangan positif  $a$  dan  $b$ .

**Keluaran** berupa sebuah bilangan bulat yang menyatakan nilai  $a$  permutasi  $b$  apabila  $a \geq b$  atau  $b$  permutasi  $a$  untuk kemungkinan yang lain.

```

1  package main
2  import "fmt"
3  func main(){
4      var a,b int
5      fmt.Scan(&a, &b)
6      if a >= b {
7          fmt.Println(permutasi(a,b))
8      }else{
9          fmt.Println(permutasi(b,a))
10     }
11 }
12 func faktorial(n int) int{
13     var hasil int = 1
14     var i int
15     for i = 1; i <= n; i++ {
16         hasil = hasil * i
17     }
18     return hasil
19 }
20 func permutasi(n,r int) int {
21     return faktorial(n) / faktorial(n-r)
22 }
```

D:\DEV\DEMO>go build contoh.go

D:\DEV\DEMO>contoh.exe

2 5  
20

Pada contoh di atas fungsi faktorial dipanggil secara tidak langsung melalui fungsi permutasi, dan fungsi faktorial dan permutasi dipanggil sebagai ekspresi dari suatu statement.

# I. GUIDED

## 1. Guided 1

### Source Code

```
package main

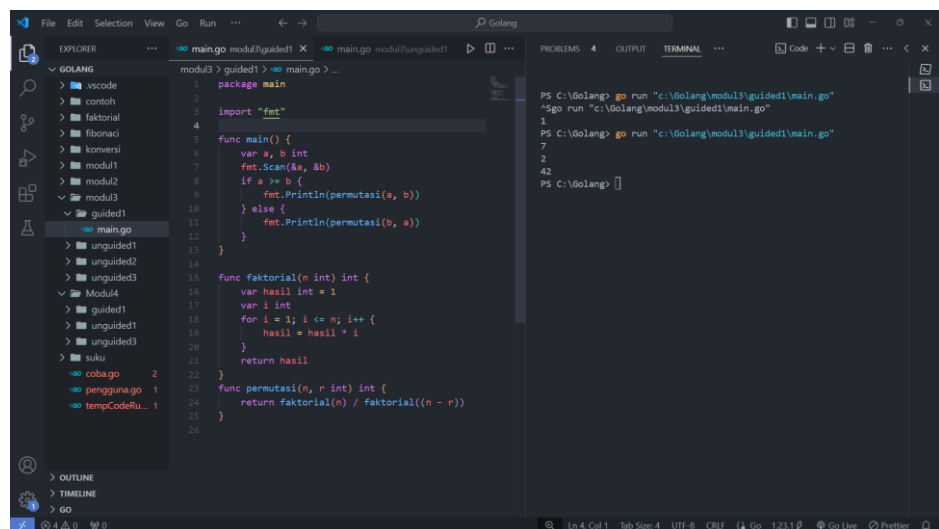
import "fmt"

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))
}
```

### Screenshot



## Deskripsi

Program ini adalah program untuk mencari permutasi. Pada program terdapat tiga fungsi yaitu fungsi main (func main), fungsi faktorial (func factorial) dan fungsi permutasi (func permutasi). Func factorial digunakan untuk mencari nilai factorial dari bilangan yang di input, terdapat parameter n untuk manampung variable input. Kemudian dalam fungsi terdapat deklarasi variable hasil = 1 dan I dengan tipe data integer. Perulangan for dari i = 0 sampai n maka akan terjadi inisiasi hasil = hasil \* i, perkalian akan berlanjut hingga perulangan selesai setelah itu fungsi akan mengembalikan nilai variable hasil. Kemudian func permutasi yang digunakan untuk mengitung permutasi. Pada fungsi terdapat parameter n dan r untuk menerima nilai input. Kemudian dalam fungsi terdapat pengembalian nilai dengan memanggil func faktorial(n)/ func factorial(n-r). kemudian pada func main sebagai fungsi utama, terdapat deklarasi variable a dan b. kemudian program akan menerima input nilai a dan b. Percabangan if dengan kondisi  $a \geq b$  maka progam akan menampilkan hasil permutasi dari a permutasi b dengan pemanggilan fungsi permutasi(a, b) . Ketika kondisi if tidak tepenuhi maka program akan menampilkan hasil permutasi dari b permutasi a dengan pemanggilan fungsi permutasi( b, a)

## II. UNGUIDED

### 1. Unguided 1

#### Source Code

```
package main

import "fmt"

// Fungsi buat menghitung faktorial
func faktorial(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 permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

// Fungsi buat menghitung kombinasi
func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) *
faktorial(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 := permutasi(a, c)
        kombinasiAC := kombinasi(a, c)

        // Menghitung permutasi dan kombinasi b
dan d
        permutasiBD := permutasi(b, d)
        kombinasiBD := kombinasi(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

```

// Fungsi buat menghitung permutasi
func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

// Fungsi buat menghitung kombinasi
func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(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 := permutasi(a, c)
        kombinasiAC := kombinasi(a, c)

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

```

## Deskripsi

Program ini adalah program untuk mencari hasil dari permutasi dan kombinasi. Pada program terdapat func faktorial untuk mencari nilai faktorial dari bilangan n. Kemudian dalam fungsi terdapat deklarasi variable hasil = 1 dan i dengan tipe data integer. Perulangan for dari i = 0 sampai n maka akan terjadi inisiasi hasil = hasil \* i, perkalian akan berlanjut hingga perulangan selesai setelah itu fungsi akan mengembalikan nilai variable hasil. Kemudian func permutasi yang digunakan untuk menghitung permutasi. Pada fungsi terdapat parameter n dan r untuk menerima nilai input. Kemudian dalam fungsi terdapat pengembalian nilai dengan memanggil func faktorial(n)/ func faktorial(n-r). func kombinasi digunakan untuk mencari nilai kombinasi. Pada func kombinasi terdapat parameter n dan i dengan tipe data integer. Kemudian dalam fungsi terjadi pemngembalian nilai dengan memanggil fungsi faktorial(n) / faktorial(r) \* faktorial(n-r). kemudian pada func main sebagai fungsi utama, terdapat deklarasi variable a dan b. kemudian program akan menerima input nilai a dan b. Percabangan if dengan kondisi a >= c && b >= d maka program akan menampilkan permutasi

(a, c) , kombinasi (a,c) dan permutasi(b, d), kombinasi(b,d). kemudian Ketika kondisi if tidak terpenuhi mak program akan menampilkan "Syarat  $a \geq c$  dan  $b \geq d$  tidak terpenuhi."

## 2. Unguided 2

### Source Code

```
package main

import "fmt"

func HoFoG(n int) int {
    g := n - 2
    f := g * g
    h := f + 1
    return h
}

func GoHoF(n int) int {
    f := n * n
    h := f + 1
    g := h - 2
    return g
}

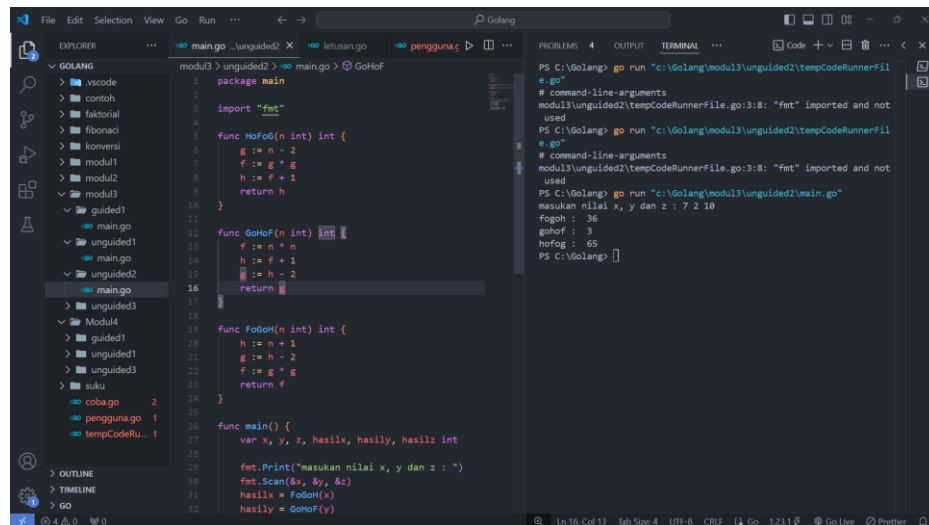
func FoGoH(n int) int {
    h := n + 1
    g := h - 2
    f := g * g
    return f
}

func main() {
    var x, y, z, hasilx, hasily, hasilz int

    fmt.Print("masukan nilai x, y dan z : ")
    fmt.Scan(&x, &y, &z)
    hasilx = FoGoH(x)
    hasily = GoHoF(y)
    hasilz = HoFoG(z)

    fmt.Println("fogoh : ", hasilx)
    fmt.Println("gohof : ", hasily)
    fmt.Println("hofog : ", hasilz)
}
```

### Screenshot



## Deskripsi

Program ini adalah program untuk mencari nilai komposisi. Dalam program terdapat fungsi `func HoFoG` untuk mencari nilai dari komposisi `hofog(x)` dengan parameter `n` tipe data integer. Dalam fungsi terdapat variable  $g = n - 2$ , kemudian variable  $f = g * g$ , dan variable  $h = f + 1$ . Kemudian memabalikan nilai `h`. kemudian `func GoHoF` dengan parameter `n` yang digunakan untuk mencari nilai komposisi dari `gohof(x)`. Dalam fungsi terdapat  $f = n * n$ , variable  $h = f + 1$ , dan variable  $g = h - 2$ . Kemudian terdapat pengembalian nilai dari fungsi `g`. `func FoGoH` dengan parameter `n` yang digunakan untuk mencari komposisi `fogoh(X)`. dalam fungsi terdapat  $h = n + 1$ ,  $g = h - 2$ , dan  $f = g * g$ . kemudian terdapat pembalikan nilai dari fungsi `f`. dalam `func main` sebagai fungsi utama terdapat deklarasi variable `x`, `y`, `z`, `hasilx`, `hasily`, `hasilz`. Kemudian program akan menerima input untuk nilai variable `x`, `y`, dan `z`. kemudian variable `hasilx = fungsi FoGoH(x)`, variable `hasily = GoHoF(y)`, dan `hasilz = HoFoG(x)`. kemudian program akan menampilkan hasil dari variable `hasilx`, `hasily`, dan `hasilz`

## 3. Unguided 3 Source Code

```
package main

import (
    "fmt"
    "math"
)

func hitungjarak(x, y, cx, cy float64) float64 {
```

```

        hasil := math.Sqrt(math.Pow(x-cx, 2) +
math.Pow(y-cy, 2))
        return hasil
    }

    func posisi(x, y, cx, cy, r float64) bool {
        hasil := hitungjarak(x, y, cx, cy) <= r
        return hasil
    }

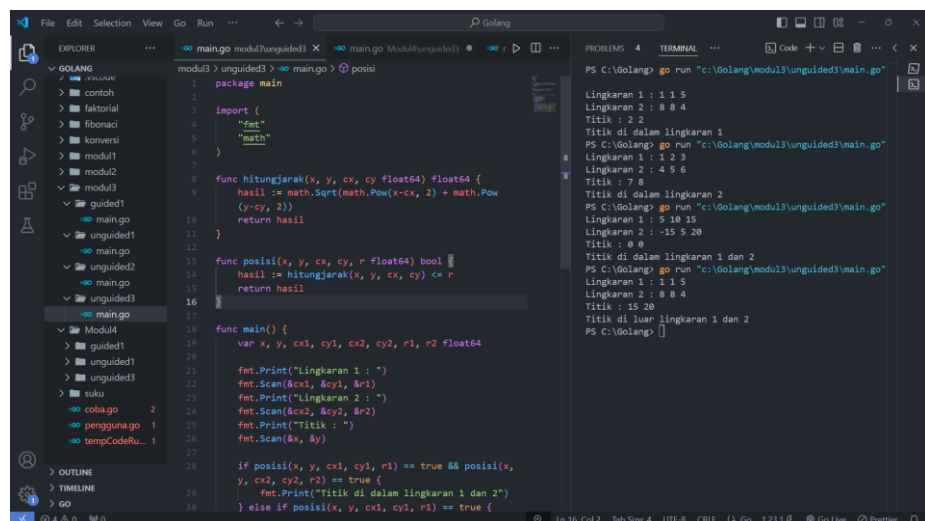
    func main() {
        var x, y, cx1, cy1, cx2, cy2, r1, r2 float64

        fmt.Print("Lingkaran 1 : ")
        fmt.Scan(&cx1, &cy1, &r1)
        fmt.Print("Lingkaran 2 : ")
        fmt.Scan(&cx2, &cy2, &r2)
        fmt.Print("Titik : ")
        fmt.Scan(&x, &y)

        if posisi(x, y, cx1, cy1, r1) == true &&
posisi(x, y, cx2, cy2, r2) == true {
            fmt.Print("Titik di dalam lingkaran 1 dan
2")
        } else if posisi(x, y, cx1, cy1, r1) == true
{
            fmt.Print("Titik di dalam lingkaran 1")
        } else if posisi(x, y, cx2, cy2, r2) == true
{
            fmt.Print("Titik di dalam lingkaran 2")
        } else {
            fmt.Print("Titik di luar lingkaran 1 dan
2")
        }
    }
}

```

## Screenshot



### **Deskripsi**

Program ini adalah program untuk menentukan apakah suatu titik berada dalam lingkaran atau diluar. Pada program terdapat func hitungjarak dengan parameter x, y, cx, cy dengan tipe data float. Dalam fungsi terdapat variabel  $hasil = \text{math.sqrt}(\text{math.pow}(x - xc, 2) + \text{math.pow}(y - yc))$ . Math.sqrt digunakan untuk menghitung nilai akar dan math.pow digunakan untuk mencari nilai kuadrat. Kemudian fungsi membalikan nilai dari variabel hasil. Func posisi dengan parameter x, y, cx, cy, r dengan integer dan fungsi tipe boolean. Dalam fungsi terdapat variabel hasil = memanggil fungsi hitungjarak (x, y, cx, cy ) lebih kecil atau sama dengan variabel r. Func main terdapat deklarasi variabel x, y, cx1, cy1, cx2, cy2, r1, r2 dengan tipe data float. Kemudian program akan menerima input untuk lingkaran 1 diisi variabel cx1, cy1, r1 kemudian lingkaran 2 di isi variabel cx2, cy2, r2 kemudian untuk titik x, y. Kemudian terdapat percabangan if dengan kondisi  $\text{posisi}(x, y, cx1, cy1, r1) == \text{true} \ \&\& \ \text{posisi}(x, y, cx2, cy2, r2) == \text{true}$ . Ketika kondisi terpenuhi maka program akan menampilkan "Titik di dalam lingkaran 1 dan 2", kondisi lainnya  $\text{posisi}(x, y, cx1, cy1, r1) == \text{true}$ , ketika kondisi terpenuhi maka program akan menampilkan "Titik di dalam lingkaran 1", kondisi lainnya  $\text{posisi}(x, y, cx2, cy2, r2) == \text{true}$ , kondisi terpenuhi maka program menampilkan "Titik di dalam lingkaran 2", dan ketika semua kondisi tidak terpenuhi maka program akan menampilkan "Titik di luar lingkaran 1 dan 2".