

**LAPORAN PRAKTIKUM  
ALGORITMA PEMOGRAMAN 2**

**MODUL 3**

**MATERI**

**FUNGSI**



Oleh:

**FEBRIAN FALIH ALWAFI**

**2311102181**

**S1F-11-02**

**S1 TEKNIK INFORMATIKA  
UNIVERSITAS TELKOM PURWOKERTO**

**2024**

## I. DASAR TEORI

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.

### ➤ Deklarasi Function

Deklarasi fungsi sama dengan prosedur, yaitu berada pada blok yang terpisah dengan program utama. 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 r) -> real
2	kamus
3	luasAlas, volume: real
4	algoritma
5	luasAlas <- 3.14 * (jari_jari jari_j
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
13	volume = luasAlas * tinggi
14	return volume
15	}

➤ **Cara Pemanggilan Function**

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	5
16	10
17	v1   volumeTabung(r,t) <span style="float:right">// cara pemanggilan #1</span>

## II. GUIDED

### 1. Source Code :

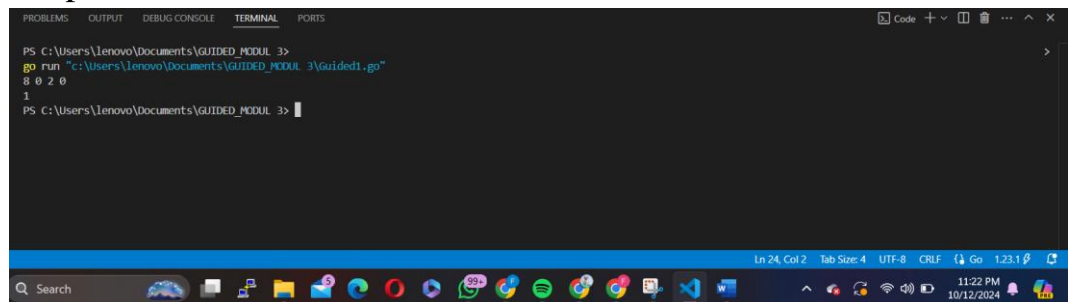
```
package main
import "fmt"

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))
}
```

### Output :



```
PS C:\Users\lenovo\Documents\GUIDED_MODULE_3>
go run "c:\Users\lenovo\Documents\GUIDED_MODULE_3\guided1.go"
8 2
1
PS C:\Users\lenovo\Documents\GUIDED_MODULE_3>
```

Deskripsi :

Program di atas bertujuan untuk menghitung **permutasi** dari dua bilangan yang dimasukkan oleh pengguna. Program ini dimulai dengan meminta dua bilangan bulat (a dan b) dari pengguna. Kemudian, program membandingkan kedua bilangan tersebut, memastikan bahwa bilangan yang lebih besar digunakan sebagai nilai n dan yang lebih kecil sebagai nilai r dalam perhitungan permutasi. Setelah itu, program menghitung permutasi dengan memanggil fungsi permutasi, yang menggunakan faktorial untuk menghitung hasil, dan mencetak hasil tersebut ke layar.

## 2. Source Code :

```
package main

import "fmt"

func factorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
    return result
}

func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}
```

```

}

func main() {

    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)

    if a >= c && b >= d {

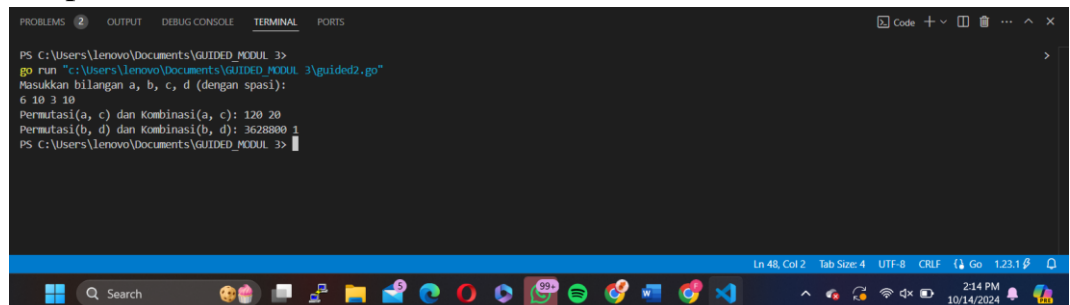
        permutasiAC := permutation(a, c)
        kombinasiAC := combination(a, c)

        permutasiBD := permutation(b, d)
        kombinasiBD := combination(b, d)

        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_MODULE_3>
go run "c:\Users\lenovo\Documents\GUIDED_MODULE_3\guided2.go"
Masukkan bilangan a, b, c, d (dengan spasi):
6 10 3 10
Permutasi(a, c) dan Kombinasi(a, c): 120 20
Permutasi(b, d) dan Kombinasi(b, d): 3628800 1
PS C:\Users\lenovo\Documents\GUIDED_MODULE_3>

```

Deskripsi :

Program ini di gunakan untuk menghitung nilai permutasi dan kombinasi berdasarkan empat bilangan bulat yang dimasukkan oleh pengguna, yaitu a, b, c, dan d.

Permutasi menghitung berapa banyak cara menyusun  $r$  objek dari  $n$  objek yang berbeda, sementara kombinasi menghitung berapa banyak cara memilih  $r$  objek dari  $n$  objek tanpa memperhatikan urutan.

### III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

#### 1. Source Code :

```
// Febrian Falih Alwafi
// 2311102181
//S1F-11-02

package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung jarak antara dua titik (a, b) dan (c, d)
func hitungJarak(x1, y1, x2, y2 float64) float64 {
    return math.Hypot(x1-x2, y1-y2) // Menggunakan fungsi
    math.Hypot untuk menghitung jarak Euclidean
}

// Fungsi untuk memeriksa apakah titik (x, y) berada di dalam
lingkaran dengan pusat (cx, cy) dan radius r
func diDalamLingkaran(cx, cy, r, x, y float64) bool {
    return hitungJarak(cx, cy, x, y) <= r
}

func main() {
    // Array untuk menyimpan data dua lingkaran
    var lingkaran [2]struct {
        cx, cy, r float64
    }

    fmt.Println("Masukkan data dua lingkaran (koordinat pusat
    dan radius):")
```



```

for i := 0; i < 2; i++ {
    fmt.Printf("Lingkaran %d - masukkan (cx, cy, r): ", i+1)
    fmt.Scan(&lingkaran[i].cx, &lingkaran[i].cy,
&lingkaran[i].r)
}

// Meminta input koordinat titik
var x, y float64
fmt.Print("Masukkan koordinat titik (x, y): ")
fmt.Scan(&x, &y)

// Memeriksa apakah titik berada di dalam salah satu atau
kedua lingkaran
var beradaDiLingkaran [2]bool
for i := 0; i < 2; i++ {
    beradaDiLingkaran[i] =
diDalamLingkaran(lingkaran[i].cx, lingkaran[i].cy,
lingkaran[i].r, x, y)
}

// Menampilkan hasil
switch {
case beradaDiLingkaran[0] && beradaDiLingkaran[1]:
    fmt.Println("Titik berada di dalam kedua lingkaran.")
case beradaDiLingkaran[0]:
    fmt.Println("Titik berada di dalam lingkaran 1.")
case beradaDiLingkaran[1]:
    fmt.Println("Titik berada di dalam lingkaran 2.")
default:
    fmt.Println("Titik berada di luar kedua lingkaran.")
}
}

```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\Documents\Unguided_Modul 3> go run "c:\Users\lenovo\Documents\Unguided_Modul 3\Unguided1.go"
Masukkan data dua lingkaran (koordinat pusat dan radius):
Lingkaran 1 - masukkan (cx, cy, r): 1 3 5
Lingkaran 2 - masukkan (cx, cy, r): 7 7 4
Masukkan koordinat titik (x, y): 2 2
Titik berada di dalam lingkaran 1.
PS C:\Users\lenovo\Documents\Unguided_Modul 3>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\Documents\Unguided_Modul 3> go run "c:\Users\lenovo\Documents\Unguided_Modul 3\Unguided1.go"
Masukkan data dua lingkaran (koordinat pusat dan radius):
Lingkaran 1 - masukkan (cx, cy, r): 2 3 4
Lingkaran 2 - masukkan (cx, cy, r): 5 6 7
Masukkan koordinat titik (x, y): 8 9
Titik berada di dalam lingkaran 2.
PS C:\Users\lenovo\Documents\Unguided_Modul 3>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\Documents\Unguided_Modul 3> go run "c:\Users\lenovo\Documents\Unguided_Modul 3\Unguided1.go"
Masukkan data dua lingkaran (koordinat pusat dan radius):
Lingkaran 1 - masukkan (cx, cy, r): 6 12 18
Lingkaran 2 - masukkan (cx, cy, r): -18 6 24
Masukkan koordinat titik (x, y): 0 0
Titik berada di dalam lingkaran 1 dan 2.
PS C:\Users\lenovo\Documents\Unguided_Modul 3>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\Documents\Unguided_Modul 3> go run "c:\Users\lenovo\Documents\Unguided_Modul 3\Unguided1.go"
Masukkan data dua lingkaran (koordinat pusat dan radius):
Lingkaran 1 - masukkan (cx, cy, r): 1 1 5
Lingkaran 2 - masukkan (cx, cy, r): 8 8 4
Masukkan koordinat titik (x, y): 12 17
Titik berada di luar lingkaran 1 dan 2.
PS C:\Users\lenovo\Documents\Unguided_Modul 3>
```

Deskripsi :

Program ini bertujuan untuk menentukan posisi suatu titik relatif terhadap dua lingkaran. Program ini memiliki dua fungsi utama: yang pertama, `hitungJarak`, digunakan untuk menghitung jarak antara dua titik dalam ruang dua dimensi. Fungsi kedua, `diDalamLingkaran`,

memeriksa apakah titik tertentu berada di dalam lingkaran yang memiliki pusat dan radius tertentu.

Dalam bagian utama program, pengguna diminta untuk memasukkan data dua lingkaran, termasuk pusat dan radius masing-masing.

Selanjutnya, pengguna juga diminta untuk memasukkan koordinat titik yang ingin diuji. Program kemudian memeriksa apakah titik tersebut berada di dalam salah satu dari kedua lingkaran, menyimpan hasil pemeriksaan ini dalam bentuk boolean, yaitu true jika titik berada di dalam lingkaran dan false jika tidak.