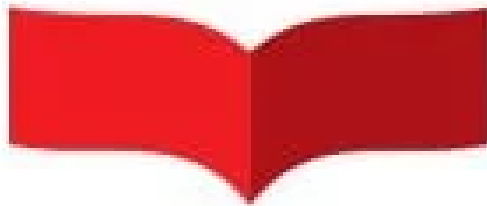


**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMOGRAMAN 2**

**MODUL 3**

**FUNGSI**



Oleh:

TRI PANJI UTOMO

2311102213

IF – 11- 02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Fungsi adalah satu kesatuan rangkaian instruksi yang memberikan atau menghasilkan suatu nilai. Fungsi memetakan input (argumen) ke suatu nilai lain (hasil). Dengan kata lain, fungsi menerima data, mengolahnya, dan mengembalikan hasil. Fungsi sangat penting dalam pemrograman karena membantu modularisasi kode, membuatnya lebih terorganisir, dan memungkinkan penggunaan ulang kode. Suatu subprogram dikatakan fungsi apabila:

1. Ada deklarasi tipe nilai yang dikembalikan.
2. Terdapat kata kunci return dalam badan subprogram.

### Deklarasi Fungsi

	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	}

Dibagian deklarasi setelah parameter terdapat tipe data dari nilai yang dikembalikan, dan pada bagian fungsi terdapat return dari nilai yang dikembalikan.

### Pemanggilan Fungsi

Fungsi bisa di-assign ke suatu variable pada saat pemanggilan, bisa dioperasikan sesuai dengan tipe data yang dikembalikan, dan juga bisa langsung ditampilkan dengan perintah print.

Notasi dalam bahasa Go
<pre>func main() {     var r,t int     var v1,v2 float64     r = 5     t = 10     v1 = volumeTabung(r,t)                                     // cara pemanggilan #1 </pre>
<pre> v2 = volumeTabung(r,t) + volumeTabung(15,t)    // cara pemanggilan #2 fmt.Println(volumeTabung(14,100))              // cara pemanggilan #3 } </pre>

## II. GUIDED

### 1. Guided 1

#### Source Code

```
package main

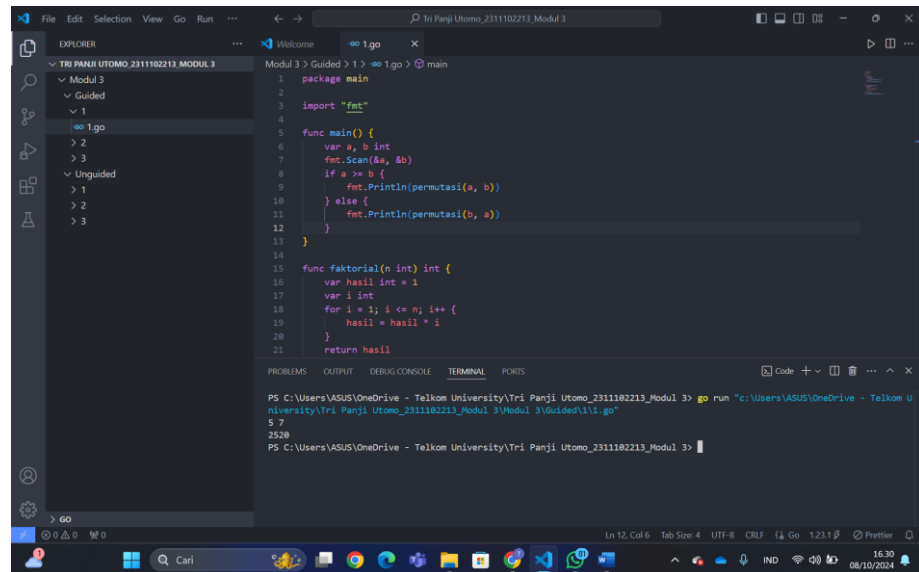
import "fmt"

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 dirancang untuk menghitung permutasi dari dua bilangan bulat yang diinput oleh user dengan memeriksa apakah a lebih besar atau sama dengan b. Jika benar, program akan menghitung permutasi dengan a sebagai n dan b sebagai r. Jika tidak, program akan menghitung permutasi dengan b sebagai n dan a sebagai r.

## 2. Guided 2

### 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 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**

```
27 func main() {
28     fmt.Println("Masukkan bilangan a, b, c, d (dengan spasi): ")
29     fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)
30
31     // Cek syarat a >= c dan b >= d
32     if a >= c && b >= d {
33         // Menghitung permutasi dan kombinasi a dan c
34         permutasiAC := permutation(a, c)
35         kombinasiAC := combination(a, c)
36
37         // Menghitung permutasi dan kombinasi b dan d
38         permutasiBD := permutation(b, d)
39         kombinasiBD := combination(b, d)
40
41         // Output hasil
42         fmt.Println("Permutasi(a, c) dan Kombinasi(a, c):", permutasiAC, kombinasiAC)
43         fmt.Println("Permutasi(b, d) dan Kombinasi(b, d):", permutasiBD, kombinasiBD)
44     } else {
45         fmt.Println("Syarat a >= c dan b >= d tidak terpenuhi.")
46     }
47 }
48 }
```

Terminal Output:

```
niversity\Tri Panji Utomo_2311182213_Modul 3\Modul 3\Guided\2\2.go
Masukkan bilangan a, b, c, d (dengan spasi):
4 3 2 1
Permutasi(a, c) dan Kombinasi(a, c): 12 6
Permutasi(b, d) dan Kombinasi(b, d): 3 3
PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3\Modul 3\Guided\2\2.go
Masukkan bilangan a, b, c, d (dengan spasi):
1 2 3
Syarat a >= c dan b >= d tidak terpenuhi.
PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3>
```

## Deskripsi

Program bertujuan untuk menghitung permutasi dan kombinasi dari dua pasang bilangan yang diinput oleh user. Program memeriksa apakah  $a \geq c$  dan  $b \geq d$ . Jika kondisi ini terpenuhi, maka program akan melanjutkan untuk menghitung permutasi dan kombinasi. Jika kondisi tidak terpenuhi, program akan mencetak pesan bahwa syarat tidak terpenuhi.

### III. UNGUIDED

#### 1. Unguided 1 Source Code

```
package main

import (
    "fmt"
    "math"
)

func hitungJarak(x1213, y1213, x2213, y2 float64) float64 {
    return math.Sqrt(math.Pow(x2213-x1213, 2) +
math.Pow(y2-y1213, 2))
}

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

func main() {
    var x1213, y1213, r1, x2213, y2, r2, xt213, yt213
float64

    fmt.Print("lingkaran 1 : ")
    fmt.Scanln(&x1213, &y1213, &r1)

    fmt.Print("lingkaran 2 : ")
    fmt.Scanln(&x2213, &y2, &r2)

    fmt.Print("titik sembarang : ")
    fmt.Scanln(&xt213, &yt213)

    diLingkaran1 := diDalamLingkaran(x1213, y1213, r1,
xt213, yt213)
    diLingkaran2 := diDalamLingkaran(x2213, y2, r2, xt213,
yt213)

    if diLingkaran1 && diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    }
}
```

```

    } else if diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

## Screenshot

The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code in the editor defines a function `diLingkaran` that checks if a point `(x, y)` is inside a circle with center `(cx, cy)` and radius `r`. The terminal shows the execution of the program for three different points, with output indicating whether the point is inside or outside the circles.

```

16 func main() {
28     diLingkaran := diLingkaran(x2213, y2213, r2, x213, y213)
29     diLingkaran2 := diLingkaran(x2213, y2, r2, x213, y213)
30
31     if diLingkaran1 && diLingkaran2 {
32         fmt.Println("Titik di dalam lingkaran 1 dan 2")
33     }
34 }

```

```

182213_Modul 3> go run "c:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3\Modul 3\Unguided\3\3.go"
lingkaran 1 : 1 1 5
lingkaran 2 : 8 8 4
titik sembarang : 2 2
Titik di dalam lingkaran 1
PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3> go run "c:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3\Modul 3\Unguided\3\3.go"
lingkaran 1 : 1 1 5
lingkaran 2 : 4 5 6
titik sembarang : 7 8
Titik di dalam lingkaran 2
PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3> go run "c:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3\Modul 3\Unguided\3\3.go"
lingkaran 1 : 5 10 15
lingkaran 2 : -15 4 20
titik sembarang : 0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3> go run "c:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3\Modul 3\Unguided\3\3.go"
lingkaran 1 : 1 1 5
lingkaran 2 : 8 8 4
titik sembarang : 15 20
Titik di luar lingkaran 1 dan 2
PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 3>

```

## Deskripsi

Program ini membantu menentukan posisi sebuah titik relatif terhadap dua lingkaran dengan memeriksa apakah titik tersebut berada di dalam salah satu atau kedua lingkaran.