

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 3**

**FUNGSI**



**Disusun Oleh :**

**Kanasya Abdi Aziz / 2311102140**

**S1-IF-11-05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom.,M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### A. Definisi Fungsi

I. Fungsi adalah kumpulan blok kode yang diberi nama untuk mempermudah penggunaannya. Dengan menerapkan fungsi dengan baik, kode menjadi lebih modular dan mengikuti prinsip DRY (Don't Repeat Yourself). Artinya, kita tidak perlu menulis ulang kode yang sama berkali-kali, cukup mendefinisikan fungsi tersebut sekali, lalu memanggilnya saat diperlukan.

Suatu subprogram dikatakan fungsi apabila:

1. Ada deklarasi tipe nilai yang dikembalikan.
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.

### B. Deklarasi Fungsi

	Notasi Algoritma
1	<code>function &lt;nama function&gt; (&lt;params&gt;) -&gt; &lt;type&gt;</code>
2	<code>kamus</code>
3	<code>  {deklarasi variabel lokal dari fungsi}</code>
4	<code>  ...</code>
5	<code>algoritma</code>
6	<code>  {badan algoritma fungsi}</code>
7	<code>  ...</code>
8	<code>  return &lt;value/variabel&gt;</code>
9	<code>endfunction</code>
	Notasi dalam bahasa Go
10	<code>func &lt;nama function&gt; (&lt;params&gt;) &lt;type&gt; {</code>
11	<code>  /* deklarasi variabel lokal dari fungsi */</code>
12	<code>  ...</code>
13	<code>  /* badan algoritma fungsi*/</code>
14	<code>  ...</code>
15	<code>  return &lt;value/variabel&gt;</code>
16	<code>}</code>
17	

Pada bagian deklarasi terlihat setelah parameter, terdapat tipe data 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	}

### C. Cara Pemanggilan Fungsi

Sama halnya dengan prosedur, pemanggilan fungsi cukup dilakukan dengan penulisan nama fungsi beserta argument yang diminta oleh parameter dari fungsi. Perbedaannya dengan prosedur adalah fungsi bisa di-assign ke suatu variable, menjadi bagian dari ekspresi, dan argument 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	v2 = volumeTabung(r,t) + volumeTabung(15,t) <span style="float: right;">// cara pemanggilan #2</span>
19	fmt.Println(volumeTabung(14,100)) <span style="float: right;">// cara pemanggilan #3</span>
20	}

## II. GUIDED

### 1. Guided 1

#### Soal Studi Case

Contoh program dengan function

#### Sourcecode

```
package main

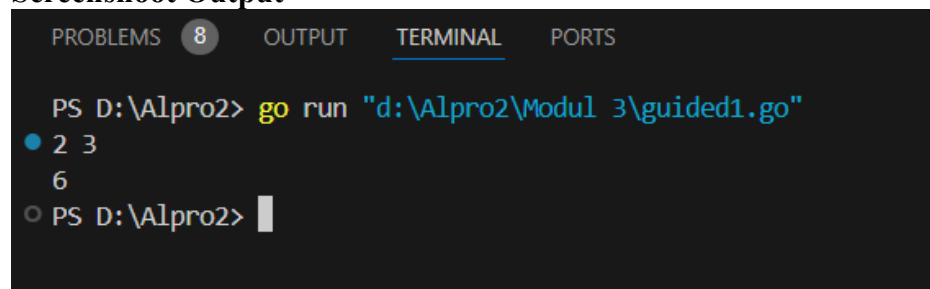
import "fmt"

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)
}
```

#### Screenshoot Output



```
PROBLEMS 8 OUTPUT TERMINAL PORTS

PS D:\Alpro2> go run "d:\Alpro2\Modul 3\guided1.go"
2 3
6
PS D:\Alpro2> 
```

#### Deskripsi Program

Program di atas merupakan program sederhana untuk menghitung permutasi  $P(n,r)$  dalam bahasa Go. Permutasi adalah cara untuk menyusun elemen dari sebuah himpunan dengan memperhatikan urutan.

### Penjelasan Singkat:

#### 1. Input:

- Program meminta pengguna memasukkan dua bilangan bulat (a dan b).

#### 2. Proses:

- Program akan memeriksa apakah nilai  $a \geq b$ . Jika benar, akan dihitung permutasi dari  $P(a,b)$ .
- Jika  $a < b$ , permutasi dihitung sebagai  $P(b,a)$ .
- Permutasi dihitung menggunakan rumus  $P(n,r) = \frac{n!}{(n-r)!}$ .

#### 3. Fungsi Utama:

- **faktorial(n)**: Menghitung faktorial dari sebuah bilangan bulat n ( $n!$ ).
- **permutasi(n, r)**: Menghitung permutasi dari dua bilangan n dan r menggunakan fungsi faktorial.

#### 4. Output:

- Program akan mencetak hasil perhitungan permutasi berdasarkan input yang diberikan oleh pengguna.

### Contoh Penggunaan:

Jika pengguna memasukkan nilai  $a = 5$  dan  $b = 3$ , program akan menghitung  $P(5,3)$  dan mengembalikan hasil permutasinya.

### Alur Eksekusi:

1. Pengguna memasukkan dua nilai.
2. Program memutuskan mana nilai yang lebih besar antara a dan b.
3. Program menghitung dan menampilkan hasil permutasi dari nilai-nilai tersebut.

## 2. Guided 2 Soal Studi Case

Membuat program dengan bahasa go untuk mencari sebuah Luas dan Keliling Persegi.

#### Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung luas persegi panjang
func hitungLuas(panjang, lebar float64) float64 {
    return panjang * lebar
}

// Fungsi untuk menghitung keliling persegi panjang
func hitungKeliling(panjang, lebar float64) float64 {
    return 2 * (panjang + lebar)
}

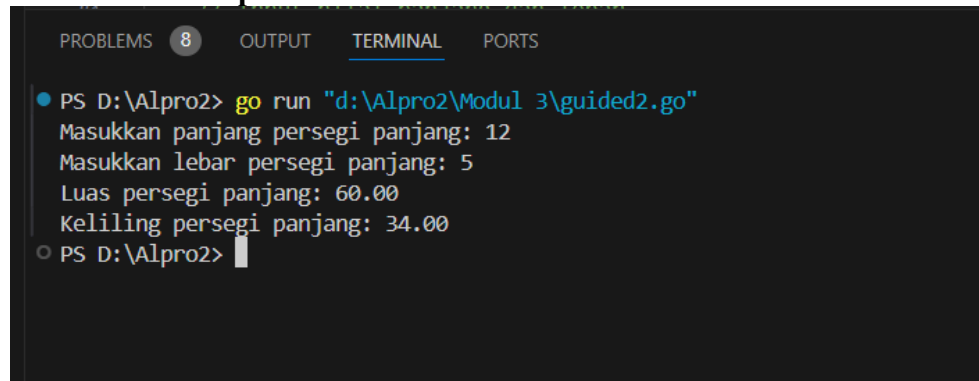
func main() {
    var panjang, lebar float64

    // Input nilai panjang dan lebar
    fmt.Print("Masukkan panjang persegi panjang: ")
    fmt.Scanln(&panjang)
    fmt.Print("Masukkan lebar persegi panjang: ")
    fmt.Scanln(&lebar)

    // Menghitung luas dan keliling
    luas := hitungLuas(panjang, lebar)
    keliling := hitungKeliling(panjang, lebar)

    // Output hasil perhitungan
    fmt.Printf("Luas persegi panjang: %.2f\n", luas)
    fmt.Printf("Keliling persegi panjang: %.2f\n", keliling)
}
```

## Screenshoot Output

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', '8', 'OUTPUT', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active. The terminal shows the execution of a Go program. The prompt is 'PS D:\Alpro2>'. The user enters 'go run "d:\Alpro2\Modul 3\guided2.go"'. The program outputs: 'Masukkan panjang persegi panjang: 12', 'Masukkan lebar persegi panjang: 5', 'Luas persegi panjang: 60.00', and 'Keliling persegi panjang: 34.00'. The prompt returns to 'PS D:\Alpro2>'.

### Deskripsi Program

Program di atas merupakan program sederhana dalam bahasa Go untuk menghitung **luas** dan **keliling persegi panjang** berdasarkan panjang dan lebar yang diinputkan oleh pengguna.

### Penjelasan Singkat:

#### 1. Input:

- Pengguna diminta memasukkan nilai **panjang** dan **lebar** dari persegi panjang.

#### 2. Proses:

- Program menghitung luas dengan rumus:  
$$\text{Luas} = \text{panjang} \times \text{lebar}$$
- Program menghitung keliling dengan rumus:  
$$\text{Keliling} = 2 \times (\text{panjang} + \text{lebar})$$

#### 3. Fungsi:

- **hitungLuas(panjang, lebar):** Menghitung luas persegi panjang.
- **hitungKeliling(panjang, lebar):** Menghitung keliling persegi panjang.

#### 4. Output:

- Program akan mencetak hasil perhitungan **luas** dan **keliling** dengan format desimal dua angka di belakang koma.

### Contoh Penggunaan:

Jika pengguna memasukkan panjang 5.0 dan lebar 3.0, program akan menghitung luas dan keliling persegi panjang lalu menampilkan hasilnya.

### III. UNGUIDED

#### 1. Unguided 1

##### Soal Studi Case

Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalian membantu Jonas? (tidak tentunya ya :p) Masukan terdiri dari empat buah bilangan asli  $a$ ,  $b$ ,  $c$ , dan  $d$  yang dipisahkan oleh spasi, dengan syarat  $a \geq c$  dan  $b \geq d$ . Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi  $a$  terhadap  $c$ , sedangkan baris kedua adalah hasil permutasi dan kombinasi  $b$  terhadap  $d$ .

Catatan: permutasi (P) dan kombinasi (C) dari  $n$  terhadap  $r$  ( $n \geq r$ ) dapat dihitung dengan menggunakan persamaan berikut!

$$P(n, r) = \frac{n!}{(n-r)!}, \text{ sedangkan } C(n, r) = \frac{n!}{r!(n-r)!}$$

##### Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung faktorial
func factorial(n int) int {
    if n == 0 {
        return 1
    }
    result := 1
    for i := 1; i <= n; i++ {
        result *= i
    }
    return result
}

// Fungsi untuk menghitung permutasi
func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

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



```

func main() {
    // Input empat bilangan: a, b, c, d
    var a, b, c, d int
    fmt.Print("Masukkan 4 bilangan: ")
    fmt.Scan(&a, &b, &c, &d)

    // Baris pertama: permutasi dan kombinasi a
    terhadap c
    p1 := permutation(a, c)
    c1 := combination(a, c)
    fmt.Printf("%d %d\n", p1, c1)

    // Baris kedua: permutasi dan kombinasi b
    terhadap d
    p2 := permutation(b, d)
    c2 := combination(b, d)
    fmt.Printf("%d %d\n", p2, c2)
}

```

### Screenshoot Output

```

PROBLEMS 8 OUTPUT TERMINAL PORTS

PS D:\Alpro2> go run "d:\Alpro2\Modul 3\unguided1.go"
● Masukkan 4 bilangan bulat: 1 2 3 4
  1 0
  2 0
○ PS D:\Alpro2>

```

### Deskripsi Program

Program di atas merupakan program dalam bahasa Go yang menghitung **permutasi** dan **kombinasi** dari dua pasang bilangan yang diinput oleh pengguna. Permutasi dan kombinasi dihitung menggunakan rumus yang melibatkan faktorial.

### Penjelasan Singkat:

#### 1. Input:

- Pengguna diminta memasukkan empat bilangan bulat: a, b, c, dan d.

## 2. Proses:

- Program menghitung **permutasi**  $P(n,r)=\frac{n!}{(n-r)!}$  dan **kombinasi**  $C(n,r)=\frac{n!}{r!(n-r)!}$  untuk:
  - a terhadap c.
  - b terhadap d.
- Hasil permutasi dan kombinasi untuk masing-masing pasangan dihitung menggunakan fungsi permutation dan combination.

## 3. Fungsi:

- **factorial(n)**: Menghitung faktorial dari bilangan n.
- **permutation(n, r)**: Menghitung permutasi  $P(n,r)$ .
- **combination(n, r)**: Menghitung kombinasi  $C(n,r)$ .

## 4. Output:

- Program menampilkan dua baris hasil:
  - Baris pertama: Hasil permutasi dan kombinasi dari a terhadap c.
  - Baris kedua: Hasil permutasi dan kombinasi dari b terhadap d.

## Contoh Penggunaan:

Jika pengguna memasukkan angka 5, 3, 2, 1, program akan menghitung permutasi dan kombinasi:

- $P(5,2)$  dan  $C(5,2)$ .
- $P(3,1)$  dan  $C(3,1)$ .

## 2. Unguided 2

### Soal Studi Case

Diberikan tiga buah fungsi matematika yaitu  $f(x) = x^2$ ,  $g(x) = x - 2$  dan  $h(x) = x + 1$ . Fungsi komposisi  $(f \circ g \circ h)(x)$  artinya adalah  $f(g(h(x)))$ . Tuliskan  $f(x)$ ,  $g(x)$  dan  $h(x)$  dalam bentuk function. Masukan terdiri dari sebuah bilangan bulat a, b dan c yang dipisahkan oleh spasi. Keluaran terdiri dari tiga baris. Baris pertama adalah  $(f \circ g \circ h)(a)$ , baris kedua  $(g \circ h)(b)$ , dan baris ketiga adalah  $(h \circ f)(c)$ !

No	Masukan	Keluaran	Penjelasan
1	7 2 10	36 3 65	(fogog)(7) = 36 (gohof)(2) = 3 (hofog)(10) = 65
2	5 5 5	16 24 10	(fogog)(5) = 16 (gohof)(5) = 24 (hofog)(5) = 10
3	3 8 4	4 63 5	(fogog)(5) = 4 (gohof)(5) = 63 (hofog)(5) = 5

### Sourcecode

```
package main

import (
    "fmt"
)

// Definisi fungsi f(x) = x^2
func f(x int) int {
    return x * x
}

// Definisi fungsi g(x) = x - 2
func g(x int) int {
    return x - 2
}

// Definisi fungsi h(x) = x + 1
func h(x int) int {
    return x + 1
}

// Komposisi f(g(h(x))) -> (fogoh)(x)
func fogoh(x int) int {
    return f(g(h(x)))
}

// Komposisi g(h(f(x))) -> (gohof)(x)
func gohof(x int) int {
    return g(h(f(x)))
}

// Komposisi h(f(g(x))) -> (hofog)(x)
func hofog(x int) int {
```

```

        return h(f(g(x)))
    }

    func main() {
        // Input tiga bilangan: a, b, c
        var a, b, c int
        fmt.Print("Masukkan 3 bilangan (pisahkan
dengan spasi): ")
        fmt.Scan(&a, &b, &c)

        // Output hasil dari komposisi fungsi
        fmt.Println(fogoh(a)) // Hasil (fogoh) (a)
        fmt.Println(gohof(b)) // Hasil (gohof) (b)
        fmt.Println(hofog(c)) // Hasil (hofog) (c)
    }

```

### Screenshot Output

```

PS D:\Alpro2> go run "d:\Alpro2\Modul 3\unguided2.go"
• Masukkan 3 bilangan (pisahkan dengan spasi): 3 6 7
4
35
26
○ PS D:\Alpro2>

```

### Deskripsi Program

Program di atas adalah program dalam bahasa Go yang melakukan komposisi fungsi menggunakan tiga fungsi dasar. Program menerima tiga bilangan sebagai input dan menghitung hasil dari berbagai komposisi fungsi untuk setiap bilangan.

### Penjelasan Singkat:

#### 1. Fungsi Dasar:

- **f(x)**: Menghitung kuadrat dari nilai x (yaitu  $x^2$ ).
- **g(x)**: Mengurangi nilai x dengan 2 (yaitu  $x-2x$ ).
- **h(x)**: Menambahkan 1 pada nilai x (yaitu  $x+1$ ).

#### 2. Komposisi Fungsi:

- **fogoh(x)**: Menghitung komposisi  $(f \circ g \circ h)(x)$ , yaitu menghitung fungsi h(x), kemudian g(h(x)), lalu f(g(h(x))).
- **gohof(x)**: Menghitung komposisi  $(g \circ h \circ f)(x)$ , yaitu menghitung fungsi f(x), kemudian h(f(x)), lalu g(h(f(x))).

- **hofog(x)**: Menghitung komposisi  $(h \circ f \circ g)(x)$ , yaitu menghitung fungsi  $g(x)$ , kemudian  $f(g(x))$ , lalu  $h(f(g(x)))$ .

### 3. Proses:

- Pengguna diminta memasukkan tiga bilangan bulat: a, b, dan c.
- Program kemudian menghitung dan menampilkan hasil dari tiga komposisi fungsi:
  - $(f \circ g \circ h)(a)$
  - $(g \circ h \circ f)(b)$
  - $(h \circ f \circ g)(c)$

### 4. Output:

- Hasil perhitungan setiap komposisi fungsi untuk masing-masing bilangan input akan ditampilkan di layar.

### Contoh Penggunaan:

Jika pengguna memasukkan 3 4 5 sebagai input, program akan menghitung:

- $(f \circ g \circ h)(3)$
- $(g \circ h \circ f)(4)$
- $(h \circ f \circ g)(5)$

## 3. Unguided 3

### Soal Studi Case

[Lingkaran] Suatu lingkaran 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. Masukan terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu  $x$  dan  $y$  dari semua titik dan juga radius direpresentasikan dengan bilangan bulat. Keluaran berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 2", "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2".

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3	Titik di dalam lingkaran 2

	4 5 6 7 8	
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung jarak antara dua titik
(a, b) dan (c, d)
func jarak(a, b, c, d float64) float64 {
    return math.Sqrt(math.Pow(a-c, 2) +
math.Pow(b-d, 2))
}

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

// Fungsi untuk mengecek posisi titik terhadap
dua lingkaran
func checkPosition(cx1, cy1, r1, cx2, cy2, r2,
x, y float64) string {
    inCircle1 := dalam(cx1, cy1, r1, x, y)
    inCircle2 := dalam(cx2, cy2, r2, x, y)

    if inCircle1 && inCircle2 {
        return "Titik berada di dalam lingkaran
1 dan 2"
    } else if inCircle1 {
        return "Titik berada di dalam lingkaran
1"
    } else if inCircle2 {
```

```

        return "Titik berada di dalam lingkaran
2"
    } else {
        return "Titik berada di luar lingkaran 1
dan 2"
    }
}

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

    // Masukkan data sesuai dengan format yang
diberikan
    fmt.Println("Masukkan koordinat pusat dan
radius lingkaran 1:")
    fmt.Scan(&cx1, &cy1, &r1)

    fmt.Println("Masukkan koordinat pusat dan
radius lingkaran 2:")
    fmt.Scan(&cx2, &cy2, &r2)

    fmt.Println("Masukkan koordinat titik
sembarang:")
    fmt.Scan(&x, &y)

    // Cek posisi titik sembarang
    result := checkPosition(cx1, cy1, r1, cx2,
cy2, r2, x, y)
    fmt.Println(result)
}

```

### Screenshot Output

```

PROBLEMS 8 OUTPUT TERMINAL PORTS
● PS D:\Alpro2> go run "d:\Alpro2\Modul 3\unguided3.go"
Masukkan koordinat pusat dan radius lingkaran 1:
2 4 3
Masukkan koordinat pusat dan radius lingkaran 2:
2 1 2
Masukkan koordinat titik sembarang:
5 4 7
Titik berada di dalam lingkaran 1
○ PS D:\Alpro2>

```

## Deskripsi Program

Program di atas adalah program dalam bahasa Go yang digunakan untuk menentukan posisi sebuah titik terhadap dua lingkaran. Program mengecek apakah titik tersebut berada di dalam salah satu, kedua, atau di luar kedua lingkaran.

## Penjelasan Singkat:

### 1. Fungsi Utama:

- **jarak(a, b, c, d)**: Menghitung jarak Euclidean antara dua titik (a,b) dan (c,d) menggunakan rumus:  $\text{jarak} = \sqrt{(a - c)^2 + (b - d)^2}$
- **dalam(cx, cy, r, x, y)**: Mengecek apakah titik (x,y) berada di dalam lingkaran yang memiliki pusat di (cx,cy) dan jari-jari rrr. Titik dianggap berada di dalam lingkaran jika jaraknya dari pusat lingkaran lebih kecil atau sama dengan radius.
- **checkPosition(cx1, cy1, r1, cx2, cy2, r2, x, y)**: Mengecek posisi titik (x,y) terhadap dua lingkaran yang memiliki pusat di (cx1,cy1) dengan jari-jari r1 dan (cx2,cy2) dengan jari-jari r2. Fungsi mengembalikan salah satu dari empat hasil:
  - Titik berada di dalam lingkaran 1 dan 2.
  - Titik berada di dalam lingkaran 1.
  - Titik berada di dalam lingkaran 2.
  - Titik berada di luar kedua lingkaran.

### 2. Proses:

- Pengguna diminta memasukkan koordinat pusat dan radius untuk dua lingkaran serta koordinat titik sembarang.
- Program kemudian menghitung jarak titik dari pusat setiap lingkaran untuk mengecek apakah titik tersebut berada di dalam lingkaran atau tidak.

### 3. Output:

- Program akan mencetak hasil pengecekan posisi titik sembarang terhadap dua lingkaran, apakah di dalam salah satu, kedua, atau di luar kedua lingkaran.

## Contoh Penggunaan:

Jika pengguna memasukkan data sebagai berikut:

- Pusat lingkaran 1: (0,0), radius: 5
- Pusat lingkaran 2: (3,3), radius: 4
- Titik sembarang: (1,1)



#### **IV. KESIMPULAN**

Setelah mempelajari fungsi dalam bahasa Go, dapat disimpulkan bahwa fungsi berperan penting dalam membuat kode lebih modular dan mendukung prinsip DRY (Don't Repeat Yourself), sehingga mengurangi pengulangan kode yang sama. Fungsi di Go dapat mengembalikan nilai dengan cara mendeklarasikan tipe nilai yang akan dikembalikan dan menggunakan kata kunci `return` untuk mengirimkannya kembali. Selain itu, fungsi juga dapat memiliki parameter, yang memungkinkan kita untuk memasukkan data ke dalam fungsi dan memprosesnya. Pemanggilan fungsi dilakukan dengan memberikan argumen yang sesuai dengan jumlah dan tipe parameter yang telah didefinisikan. Pada contoh yang dipelajari, penggunaan fungsi built-in seperti `strings.Join()` menunjukkan bagaimana fungsi membantu memanipulasi data dengan lebih efisien. Secara keseluruhan, penerapan fungsi dalam Go sangat penting untuk menjaga kode tetap terorganisir, bersih, dan efisien.

#### **V. REFERENSI**

- [1] Modul 3 Praktikum Algoritma 2
- [2] Novalagung. "Golang Tipe Data." Dasar Pemrograman Golang. Accessed October 6, 2024.  
<https://dasarpemrogramangolang.novalagung.com/A-tipe-data.html>