

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III  
FUNGSI**



**Disusun Oleh :**

**Aji Noto Sutrisno (2311102262)**

**IF 11 05**

**Dosen Pengampu :**

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

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

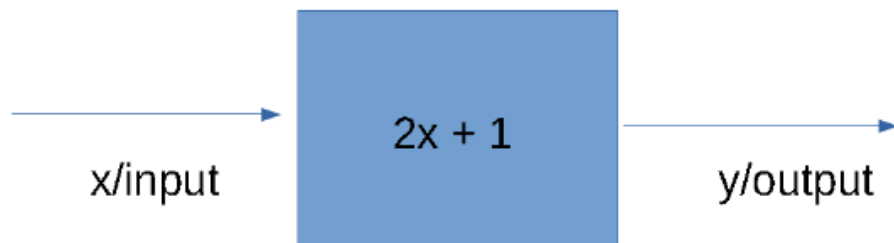
**2024**

## I. DASAR TEORI

fungsi adalah sekumpulan blok kode yang dibungkus dengan nama tertentu. fungsi itu terdiri dari tiga komponen :

- Input , x.
- Output, y.
- Rumus / kode dalam body fungsinya.

Fungsi dapat digambarkan seperti dibawah



Penjelasan gambar :

- x/Input misalkan tipenya Integer bisa diisi 1,2,3,4,5, dll.
- y/Output misalkan tipenya Integer menjadi 3,5,7,9,11.
- Kotak biru adalah body dari fungsi.

Dapat dicontohkan pada program ini

```
package main

import (
    "fmt"
)

func main() {

    panjang := 10
    lebar := 20
```

```

        luas:= panjang * lebar

        fmt.Printf("Panjang = %d, Lebar =%d maka luas adalah
%d\n",panjang, lebar, luas)

        panjang = 100
        lebar = 200

        luas = panjang * lebar

        fmt.Printf("Panjang = %d, Lebar =%d maka luas adalah
%d\n",panjang, lebar, luas)

        panjang = 1000
        lebar = 2000

        luas= panjang * lebar

        fmt.Printf("Panjang = %d, Lebar =%d maka luas adalah
%d\n",panjang, lebar, luas)

```

Program dapat disederhanakan dengan fungsi

```

package main

import (
    "fmt"
)

func main() {

    luas(10, 20)
    luas(100, 200)
    luas(1000, 2000)
}

func luas(panjang int32, lebar int32) {

```

```
        luas := panjang * lebar
        fmt.Printf("Panjang = %d, Lebar =%d maka luas adalah
%d\n", panjang, lebar, luas)

}
```

## Output

```
Panjang = 10, Lebar =20 maka luas adalah 200
Panjang = 100, Lebar =200 maka luas adalah 20000
Panjang = 1000, Lebar =2000 maka luas adalah 2000000
```

Kedua Program diatas memiliki banyak perbedaan yang signifikan.

Maka fungsi itu gunanya :

- Membuat program mudah dibaca atau readability.
- Membuat program gampang dirubah atau maintainability.
- Membuat program mudah ditest atau testability.

### A. Function Signature atau Deklarasi Fungsi.

Sekarang kita bahas komponen-komponen dan aturan membuat fungsi di-Golang. Fungsi terdiri dari beberapa komponen yaitu

#### Nama Fungsi

- Nama fungsi tidak boleh diawali dengan angka.
- Nama fungsi tidak boleh mengandung spasi.
- Jika nama fungsi diawali dengan huruf besar maka fungsi bisa diakses dari package lain.
- Jika nama fungsi diawali dengan huruf kecil maka hanya bisa diakses dalam satu package.
- nama fungsi bersifat case sensitive, jadi luas() sama Luas() berbeda.
- konvensi penamaan fungsi adalah Camel Case seperti : luasSegitiga, printHello, dll.

### Input Parameter

- Jumlah input parameter bisa nol atau lebih.
- tipe input parameter bisa apa saja

### Output Parameter

- Jumlah output parameter bisa nol atau lebih .
- tipe output parameter bisa apa saja.
- Berikut beberapa contoh program deklarasi fungsi :

1. Fungsi dengan jumlah input dan output parameter nol, berikut contoh programnya

```
package main
import (
    "fmt"
)
func main() {
    printHelloWorld()
}
func printHelloWorld() {
    fmt.Println("Hello World")
}
```

2. Fungsi dengan jumlah input parameter satu dan output parameter nol, print “hello world” sebanyak input parameter

```
package main
import "fmt"
func main() {
    printHelloWorld(10)
}
func printHelloWorld(n int) {
    for i := 0; i < n; i++ {
        fmt.Println("Hello World")
    }
}
```

```
}
```

3. Fungsi dengan satu input parameter dan satu output parameter. Contoh program hitung total jumlah anggota Slice,

```
package main

import (
    "fmt"
)

func main() {

    n := []int{1, 2, 3, 4, 5}

    fmt.Printf("Jumlah Slice n = %d", hitungSlice(n))

}

func hitungSlice(n []int) int {

    sum := 0

    for i := 0; i < len(n); i++ {

        sum = sum + n[i]

    }

    return sum

}
```

## B. Blank Identifier

Perhatikan contoh program 8 diatas baris 21

```
luas, err = hitungLuas(panjang, lebar)
```

Misal kita tidak perlu nilai err maka programnya bisa dibuat dengan blank identifier “\_” seperti dibawah.

```
luas, _ = hitungLuas(panjang, lebar)
```

Fungsi dengan input parameter lebih dari satu dan tipe datanya sama bisa digabungkan

```
package main
import (
    "fmt"
    "errors"
)

func main() {
    panjang := 10
    lebar := 10

    luas, err := hitungLuas(panjang, lebar)

    fmt.Printf("Luas = %d\n", luas)
    fmt.Println( err)

    panjang = -10
    lebar = 10

    luas, err = hitungLuas(panjang, lebar)

    fmt.Printf("Luas = %d\n", luas)
    fmt.Println( err)
```

```

}

func hitungLuas(panjang , lebar int) (int, error) {

    if panjang < 0 || lebar < 0 {
return 0, errors.New("Panjang atau lebar tidak boleh
lebih kecil dari 0")

    }

    return panjang * lebar, nil

}

```

### C. Fungsi Dengan Named Return Values

DiGolang output parameter bisa dibuat seperti input parameter, yang dideklarasikan bukan hanya tipe datanya saja tetapi nama variabel. Berikut adalah contoh program tentang named return values atau output parameter dideklarasikan seperti input parameter.

```

package main

import (
    "errors"
    "fmt"
)

func main() {
    panjang := 10

    lebar := 10

    luas, err := hitungLuas(panjang, lebar)

```



```

        fmt.Printf("Luas = %d\n", luas)
        fmt.Println(err)
    }

    func hitungLuas(panjang, lebar int) (luas int, err
    error) {

        if panjang < 0 || lebar < 0 {

            luas = 0
            err = errors.New("Panjang atau lebar tidak boleh
            lebih kecil dari 0")

        }

        luas = panjang * lebar
        err = nil
        return

    }

```

Output:

```

Luas = 100
<nil>

```

#### D. Scope Variabel Fungsi

“Pada fungsi pada umumnya variabel fungsi sifatnya lokal hanya bisa diakses didalam badan fungsi itu sendiri itu sendiri”

Contohnya program diatas tidak bisa dibuat seperti kode dibawah.

```

package main

import (

```

```

    "errors"
    "fmt"
)

func main() {

    panjang := 10
    lebar := 10

    luas, err := hitungLuas()

    fmt.Printf("Luas = %d\n", luas)
    fmt.Println(err)

}

func hitungLuas() (luas int, err error) {

    if panjang < 0 || lebar < 0 {

        luas = 0
        err = errors.New("Panjang atau lebar tidak
boleh lebih kecil dari 0")
    }

    luas = panjang * lebar
    err = nil

    return

}

```

Program diatas ketika di-compile akan error karena fungsi hitungLuas mengakses variabel panjang dan lebar yang milik fungsi main.

## E. Variadic function

Fungsi variadic adalah fungsi yang mempunyai input parameter jumlahnya tidak tetap, contohnya yang terkenal sering dipakai dicontoh program blog ini adalah

```
fmt.Printf
fmt.Printf("%f",7.6)
fmt.Printf("%f %d",7.6,1)
fmt.Printf("%d %d % d",7,1,2)
```

fungsi Printf inputnya jumlahnya beda-beda jumlahnya. Berikut adalah contoh program

```
package main

import (
    "fmt"
)

func main() {

    fmt.Printf("Jumlah Slice n = %d",
hitungSlice(1,2,3,4,5))

    fmt.Printf("Jumlah Slice n = %d",
hitungSlice(1,2,3,4,5,6))

}

func hitungSlice(n ...int) int {

    sum := 0

    for _, v := range n {
        sum = sum + v
    }

}
```

```
        return sum
    }
```

Output :

```
Jumlah Slice n = 15
```

## II. GUIDED

### 1. Soal Studi Case

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.

#### Sourcecode

```
package main

import "fmt"

func main() {
    var a, b int
    fmt.Scan(&a, &b)
    if a >= b {
        fmt.Print(permutasi(a, b))
    } else {
        fmt.Print(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 D:\Pratikum Alpro 2\Laprak 3\Guided> go run "d:\Pratikum Alpro 2\Laprak 3\Guided\Guided1.go"
5 10
30240
PS D:\Pratikum Alpro 2\Laprak 3\Guided> go run "d:\Pratikum Alpro 2\Laprak 3\Guided\Guided1.go"
2 3
6
PS D:\Pratikum Alpro 2\Laprak 3\Guided> █
```

## Deskripsi

Program diatas dibuat untuk menghitung sebuah permutasi dengan cara menghitung faktorial(n) / faktorial (n-r). Program ini meminta *user* untuk menginputkan a, dan b, yang dimana inputan tadi akan diproses oleh program untuk menjalankan fungsi dari func permutasi(n, r int) didalam fungsi tersebut masih memiliki fungsi lain yaitu fungsi func faktorial(n int) fungsi ini digunakan untuk mencari factorial dari inputan *user*: Berikut algoritma dari program ini :

- *User* akan menginputkan 2 bilangan bulat yaitu a, b
- Kemudian inputan tersebut memasuki kondisi jika  $a \geq b$ , maka lakukan  $P(a,b)$  jika tidak memenuhi maka  $P(b,a)$
- Program akan menjalankan func faktorial(n int), untuk mencari factorial dari inputan user, kemudian program akan menjalankan func permutasi(n, r int)
- Setelah itu output akan tampil

## 2. Soal Studi Case

### Sourcecode

```
package main

import "fmt"

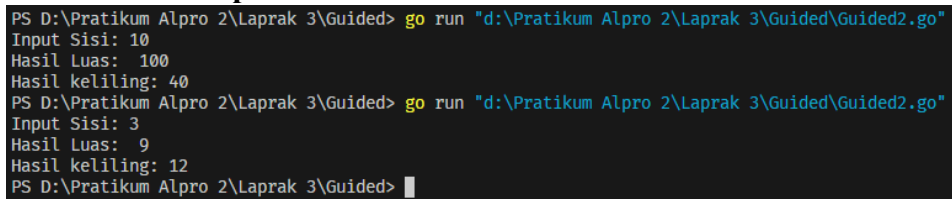
func main() {
    var s int
    fmt.Print("Input Sisi: ")
    fmt.Scan(&s)

    luas := luasPersegi(s)
    kel := kelilingPersegi(s)

    fmt.Println("Hasil Luas: ", luas)
    fmt.Print("Hasil keliling: ", kel)
```

```
}  
func luasPersegi(s int) int {  
    return s * s  
}  
  
func kelilingPersegi(s int) int {  
    return 4 * s  
}
```

## Screenshoot Output



```
PS D:\Pratikum Alpro 2\Laprak 3\Guided> go run "d:\Pratikum Alpro 2\Laprak 3\Guided\Guided2.go"  
Input Sisi: 10  
Hasil Luas: 100  
Hasil keliling: 40  
PS D:\Pratikum Alpro 2\Laprak 3\Guided> go run "d:\Pratikum Alpro 2\Laprak 3\Guided\Guided2.go"  
Input Sisi: 3  
Hasil Luas: 9  
Hasil keliling: 12  
PS D:\Pratikum Alpro 2\Laprak 3\Guided> |
```

## Deskripsi Program

Program ini dibuat untuk *user* menyelesaikan perhitungan sederhana yaitu mencari luas persegi dan keliling persegi dengan menggunakan sebuah program. Program ini memiliki beberapa fungsi yaitu `func luasPersegi(s int)` & `func kelilingPersegi(s int)` yang dimana kedua fungsi tersebut berguna untuk perhitungan sederhana. Program ini memiliki algoritma seperti berikut :

- *User* menginputkan sisi (s)
- Kemudian program akan menjalankan fungsi yang ada seperti `func luasPersegi(s int) (s * s)` & `func kelilingPersegi(s int) (4 * s)`
- Kemudian program akan menampilkan hasil perhitungan dari kedua fungsi tersebut

### III. 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  $a \leq c$  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"

func main() {
    var a, b, c, d int
    fmt.Println("Input nilai a, b, c, d : ")
    fmt.Scan(&a, &b, &c, &d)

    if a >= c && b >= d {
        fmt.Println("Hasil Permutasi : ", permutasi(a, c))
        fmt.Println("Hasil Kombinasi : ", kombinasi(a, c))
        fmt.Println("Hasil Permutasi : ", permutasi(b, d))
        fmt.Println("Hasil Kombinasi : ", kombinasi(b, d))
    } else {
        fmt.Print("Kondisi Tidak Terpenuhi")
    }
}

func faktorial(n int) int {
    var hasil int = 1
    var i int
```

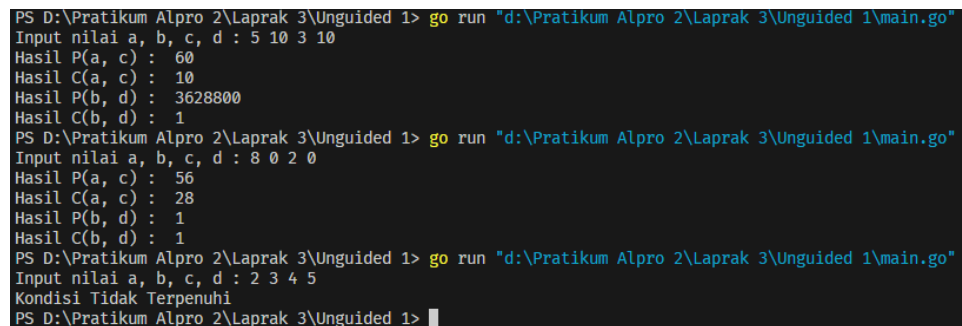
```

        for i = 1; i <= n; i++ {
            hasil = hasil * i
        }
        return hasil
    }

    func permutasi(n, r int) int {
        return faktorial(n) / faktorial(n-r)
    }
    func kombinasi(n, r int) int {
        return faktorial(n) / (faktorial(r) * faktorial(n-
r))
    }
}

```

## Screenshoot Output



```

PS D:\Pratikum Alpro 2\Laprak 3\Unguided 1> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 1\main.go"
Input nilai a, b, c, d : 5 10 3 10
Hasil P(a, c) : 60
Hasil C(a, c) : 10
Hasil P(b, d) : 3628800
Hasil C(b, d) : 1
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 1> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 1\main.go"
Input nilai a, b, c, d : 8 0 2 0
Hasil P(a, c) : 56
Hasil C(a, c) : 28
Hasil P(b, d) : 1
Hasil C(b, d) : 1
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 1> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 1\main.go"
Input nilai a, b, c, d : 2 3 4 5
Kondisi Tidak Terpenuhi
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 1>

```

## Deskripsi Program

Program ini merupakan program lanjutan dari Guided1 yang dimana guided1 mencari hasil permutasi dari dua bilangan, sedangkan program ini mencari permutasi dua bilangan dan mencari kombinasi dari dua bilangan yaitu P(a,c), C(a,c), P(b, d), dan C(b,d). Program ini memiliki beberapa fungsi yaitu func faktorial(n int), func permutasi(n, r int) dan func kombinasi(n, r int) ketiga fungsi ini masing masing memiliki kegunaan yaitu mencari faktorial dari nilai input, kemudian hasil faktorial tersebut diproses lagi dengan menggunakan kedua fungsi yang lain, yang satu mencari sebuah permutasi yang kedua mencari sebuah kombinasi dari hasil input *user* berikut algoritma dari program diatas :

- *User* menginput (a, b, c, d) type data integer
- Cek kondisi Jika  $a \geq c$  &  $b \geq d$ , maka lakukan permutasi func permutasi(n, r int) dan kombinasi func kombinasi(n, r int) seperti P(a, c) C(a, c) dan P(b, d) C(b, d).



- Lalu print hasil permutasi dan kombinasi.
- Apabila tidak memenuhi kondisi, program akan langsung menampilkan output "Kondisi Tidak Terpenuhi"

## 2. Soal Studi Case

Diberikan tiga buah fungsi matematika yaitu  $f(x) = x^2$ ,  $g(x) = x - 2$  dan  $h(x) = x + 1$ .

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  $(g \circ h \circ f)(b)$ , dan baris ketiga adalah  $(h \circ f \circ g)(c)$ !

### Sourcecode

```
package main

import "fmt"

func main() {
    var a, b, c int
    fmt.Print("Input (a) (b) (c) : ")
    fmt.Scanln(&a, &b, &c)

    fmt.Printf("(Fogoh) (%d) = %d\n", a, fgh(a))
    fmt.Printf("(Gohof) (%d) = %d\n", b, ghf(b))
    fmt.Printf("(Hofog) (%d) = %d\n", c, hfg(c))
}

func f(x int) int {
    return x * x
}

func g(x int) int {
    return x - 2
}

func h(x int) int {
    return x + 1
}

func fgh(x int) int {
    //f(g(hx))
    return f(g(h(x)))
}
```

```

    }

    func ghf(x int) int {
        //g(h(fx))
        return g(h(f(x)))
    }

    func hfg(x int) int {
        //h(f(gx))
        return h(f(g(x)))
    }
}

```

## Screenshot Output

```

PS D:\Pratikum Alpro 2\Laprak 3\Unguided 2> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 2\main.go"
Input (a)(b)(c) : 7 2 10
(Fogoh)(7) = 36
(Gohof)(2) = 3
(Hofog)(10) = 65
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 2> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 2\main.go"
Input (a)(b)(c) : 5 5 5
(Fogoh)(5) = 16
(Gohof)(5) = 24
(Hofog)(5) = 10
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 2> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 2\main.go"
Input (a)(b)(c) : 3 8 4
(Fogoh)(3) = 4
(Gohof)(8) = 63
(Hofog)(4) = 5
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 2>

```

## Deskripsi Program

Program diatas dibuat untuk mencari hasil perhitungan fungsi dengan menggunakan elemen dasar atau fungsi dasar yang berupa  $f(x) = x * x$ ;  $g(x) = x - 2$ ;  $h(x) = x + 1$ , yang kemudian dilakukan perkalian pada masing masing fungsi, seperti  $(fogoh)(x)$  artinya adalah  $f(g(h(x)))$ ,  $(gohof)(x)$  adalah  $g(h(f(x)))$ , dan  $(hofog)(x)$  maka  $h(f(g(x)))$ . Ketiganya melibatkan perkalian antar fungsi yang ada. Jadi pada program diatas memiliki tiga fungsi yaitu `func fgh(x int)` ; `func ghf(x int)` dan `func hfg(x int)` ketiga fungsi diatas mencari hasil perkalian fungsi dari inputan *user*. Program ini memiliki algoritma sebagai berikut :

- *User* menginput 3 buah nilai yang masing masing akan dijalankan oleh fungsi (a, b, c)
- Input (a), akan menjalankan fungsi, `func fgh(x int)`
- Input (b), akan menjalankan fungsi, `func ghf(x int)`
- Input (c), akan menjalankan fungsi, `func hfg(x int)`
- Setelah itu program akan menampilkan hasil dari perkalian fungsi tersebut

### 3. Soal Studi Case

[Lingkaran] Suatu lingkaran didefinisikan dengan koordinat titik pusat (ex, cy) dengan radius 7. 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".

Fungsi untuk menghitung Jarak titik (a, b) dan (c, d) dimana rumus Jarak adalah:

$$jarak = \sqrt{(a - c)^2 + (b - d)^2}$$

dan juga fungsi untuk menentukan posisi sebuah titik sembarang berada di dalam suatu lingkaran atau tidak.

```
function jarak(a,b,c,d real) -> real
{Mengembalikan jarak antara titik (a,b) dan titik (c,d)}

function didalam (cx,cy, r, x, y real) -> boolean
{Mengembalikan true apabila titik (x,y) berada di dalam
lingkaran yang memiliki titik pusat (cx,cy) dan radius r}
```

Catatan: Lihat paket math dalam lampiran untuk menggunakan fungsi math.Sqrt() untuk menghitung akar kuadrat.

#### Sourcecode

```
package main

import (
    "fmt"
```

```

    "math"
)

func main() {
    var cx, cxx, cy, cyy, r, rr, x, y float64

    fmt.Println("Lingkaran 1")
    fmt.Print("(cx) (cy) (r) : ")
    fmt.Scanln(&cx, &cy, &r)

    fmt.Println("Lingkaran 2")
    fmt.Print("(cx) (cy) (r) : ")
    fmt.Scanln(&cxx, &cyy, &rr)

    fmt.Print("Input koordinat sembarang (x) (y) : ")
    fmt.Scanln(&x, &y)

    //mari kita cek posisi Lingkaran1 dang Lingkaran2
    posisi1 := inercircle(cx, cy, x, y, r)
    posisi2 := inercircle(cxx, cyy, x, y, rr)

    if posisi1 && posisi2 {
        fmt.Println("Titik di dalam lingkaran 1 dan
2")
    } else if posisi1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if posisi2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

func jarak(a, b, c, d float64) float64 {
    //pengecekan Jarak
    kiri := (a - c) * (a - c)
    kanan := (b - d) * (b - d)
    return math.Sqrt(kiri + kanan)
    //math.Sqrt digunakan untuk pemanggilan akar
}

func didalam(cx, cy, x, y, r float64) bool {
    //pengecekan apakah x, y berada dalam lingkaran
    return jarak(cx, cy, x, y) <= r
}

```

## Screenshoot Output

```
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 3> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 3\main.go"
Lingkaran 1
Koordinat (x)(y)(radius) : 1 1 5
Lingkaran 2
Koordinat (x)(y)(radius) : 8 8 4
Input koordinat sembarang (x)(y) : 2 2
Titik di dalam lingkaran 1
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 3> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 3\main.go"
Lingkaran 1
Koordinat (x)(y)(radius) : 1 2 3
Lingkaran 2
Koordinat (x)(y)(radius) : 5 10 15
Input koordinat sembarang (x)(y) : 7 8
Titik di dalam lingkaran 2
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 3> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 3\main.go"
Lingkaran 1
Koordinat (x)(y)(radius) : 5 10 15
Lingkaran 2
Koordinat (x)(y)(radius) : -15 4 20
Input koordinat sembarang (x)(y) : 0 0
Titik di dalam lingkaran 1 dan 2
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 3> go run "d:\Pratikum Alpro 2\Laprak 3\Unguided 3\main.go"
Lingkaran 1
Koordinat (x)(y)(radius) : 1 1 5
Lingkaran 2
Koordinat (x)(y)(radius) : 8 8 4
Input koordinat sembarang (x)(y) : 15 20
Titik di luar lingkaran 1 dan 2
PS D:\Pratikum Alpro 2\Laprak 3\Unguided 3> █
```

## Deskripsi Program

Program ini dibuat untuk menentukan apakah titik koordinat berada pada lingkaran 1 atau lingkaran 2, atau di tengah keduanya. Program ini memiliki beberapa fungsi yaitu `func jarak(a, b, c, d float64)` dan `func didalam(cx, cy, x, y, r float64)`. Program ini mempunyai algoritma sebagai berikut :

- *User* menginputkan  $(cx, cy, r)$  dan  $(x, y)$  untuk lingkaran 1 dan lingkaran 2
- Kemudian program akan mengecek apakah koordinat masih didalam radius lingkaran " $(cx, cy, x, y) \leq r$ "
- Setelah true, maka program akan menjalankan fungsi `jarak(a, b, c, d float64)` untuk mencari jarak setiap lingkaran.
- Kemudian program akan menjalankan hasilnya dengan percabangan yang dimana jika `lingkaran1 && lingkaran2`, maka ("Titik di dalam lingkaran 1 dan 2"), jika `lingkaran1` ("Titik di dalam lingkaran 1"), jika `lingkaran2` ("Titik di dalam lingkaran 2") jika tidak ketiganya maka ("Titik di luar lingkaran 1 dan 2")

#### **IV. DAFTAR PUSTAKA**

**I. Satriawan**, "Golang Bagian 15," *Medium*, 23 Pebruari 2021,  
<https://medium.com/clean-code-62/golang-bagian-15-d96017950b22>. diakses  
pada 13 Oktober 2024.