

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 3  
FUNGSI**



**Disusun Oleh:**

**Boutefhika Nuha Ziyadatul Khair/2311102316**

**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

Dalam konsep pemrograman, fungsi adalah sekumpulan blok kode yang dibungkus dengan nama tertentu. Penerapan fungsi yang tepat akan menjadikan kode lebih modular dan juga dry (singkatan dari don't repeat yourself) yang artinya kita tidak perlu menuliskan banyak kode untuk kegunaan yang sama berulang kali. Cukup deklarasikan sekali saja blok kode sebagai suatu fungsi, lalu panggil sesuai kebutuhan.

### B. Penerapan fungsi

Sebenarnya kita sudah mengimplementasikan fungsi pada banyak praktek sebelumnya, yaitu fungsi `main()`. Fungsi `main()` sendiri merupakan fungsi utama pada program Go, yang akan dieksekusi ketika program dijalankan.

Selain fungsi `main()`, kita juga bisa membuat fungsi lainnya. Dan caranya cukup mudah, yaitu dengan menuliskan keyword `func` kemudian diikuti nama fungsi, lalu kurung `()` (yang bisa diisi parameter), dan diakhiri dengan kurung kurawal untuk membungkus blok kode.

Data yang digunakan sebagai value parameter saat pemanggilan fungsi biasa disebut dengan argument parameter (atau argument).

Parameter merupakan variabel yang menempel di fungsi yang nilainya ditentukan saat pemanggilan fungsi tersebut. Parameter sifatnya opsional, suatu fungsi bisa tidak memiliki parameter, atau bisa saja memiliki satu atau banyak parameter (tergantung kebutuhan).

Agar lebih jelas, silakan lihat dan praktekan kode contoh implementasi fungsi berikut ini:

```
package main

import "fmt"
import "strings"

func main() {
    var names = []string{"John", "Wick"}
    printMessage("halo", names)
}

func printMessage(message string, arr []string) {
    var nameString = strings.Join(arr, " ")
}
```

```
    fmt.Println(message, nameString)
}
```

Code di atas, sebuah fungsi baru dibuat dengan nama `printMessage()` memiliki 2 buah parameter yaitu string `message` dan slice string `arr`.

Fungsi tersebut dipanggil dalam `main()`, dalam pemanggilannya disisipkan dua buah argument parameter.

1. Argument parameter pertama adalah string "halo" yang ditampung parameter `message`
2. Argument parameter ke-2 adalah slice string `names` yang nilainya ditampung oleh parameter `arr`

Di dalam `printMessage()`, nilai `arr` yang merupakan slice string digabungkan menjadi sebuah string dengan pembatas adalah karakter **spasi**. Penggabungan slice dapat dilakukan dengan memanfaatkan fungsi `strings.Join()` (berada di dalam package `strings`).

### C. Fungsi Dengan Return Value / Nilai Balik

Fungsi memiliki attribute **return value** atau nilai balik. Fungsi yang memiliki return value, saat deklarasinya harus ditentukan terlebih dahulu tipe data dari nilai baliknya.

Fungsi yang tidak mengembalikan nilai apapun (contohnya seperti fungsi `main()` dan `printMessage()`) biasa disebut dengan **void function**.

Selain sebagai penanda nilai balik, keyword `return` juga bisa dimanfaatkan untuk menghentikan proses dalam blok fungsi di mana ia ditulis. Contohnya bisa dilihat pada kode berikut.

```
package main

import "fmt"

func main() {
    divideNumber(10, 2)
    divideNumber(4, 0)
    divideNumber(8, -4)
}

func divideNumber(m, n int) {
    if n == 0 {
        fmt.Printf("invalid divider. %d cannot divided by %d\n", m, n)
    }
}
```

```
        return
    }

    var res = m / n
    fmt.Printf("%d / %d = %d\n", m, n, res)
}
```

Fungsi `divideNumber()` dirancang tidak memiliki nilai balik. Fungsi ini dibuat untuk membungkus proses pembagian 2 bilangan, lalu menampilkan hasilnya.

Di dalamnya terdapat proses validasi nilai variabel pembagi, jika nilainya adalah 0, maka akan ditampilkan pesan bahwa pembagian tidak bisa dilakukan, lalu proses dihentikan pada saat itu juga (dengan memanfaatkan keyword `return`). Jika nilai pembagi valid, maka proses pembagian diteruskan.

#### D. Fungsi Tambahan

- Penggunaan Fungsi `math.Pow()`  
Fungsi `math.Pow()` digunakan untuk operasi pangkat nilai. `math.Pow(2, 3)` berarti 2 pangkat 3, hasilnya 8. Fungsi ini berada dalam package `math`.
- Penggunaan Konstanta `math.Pi`  
`math.Pi` adalah konstanta bawaan package `math` yang merepresentasikan Pi atau  $22/7$

## II. GUIDED

1. 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

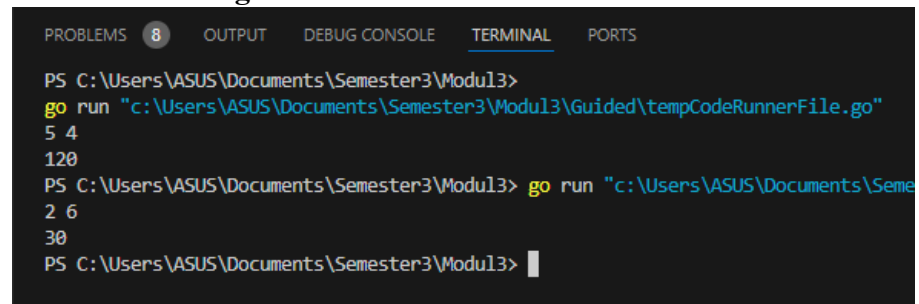
import "fmt"

func main() {
    var a, b int
    fmt.Scan(&a, &b)
    if a >= b {
        fmt.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 Program



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS\Documents\Semester3\Modul3>
go run "c:\Users\ASUS\Documents\Semester3\Modul3\Guided\tempCodeRunnerFile.go"
5 4
120
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\ASUS\Documents\Semester3\Modul3\Guided\tempCodeRunnerFile.go"
2 6
30
PS C:\Users\ASUS\Documents\Semester3\Modul3>
```

### Deskripsi Program

Program ini berfungsi untuk menghitung permutasi dari dua angka yang dimasukkan oleh pengguna. Pertama, dalam fungsi `main`, program meminta pengguna memasukkan dua nilai bilangan bulat `a` dan `b`.

menggunakan `fmt.Scan`. Kemudian, program memeriksa apakah `a` lebih besar atau sama dengan `b`. Jika benar, maka fungsi `permutasi(a, b)` dipanggil untuk menghitung permutasi dari `a` dan `b`, dan jika tidak, `permutasi(b, a)` yang dipanggil. Fungsi `permutasi` memanggil fungsi `faktorial` untuk menghitung faktorial dari `n` dan `n-r`, kemudian membagi hasil faktorial tersebut. Faktorial dihitung menggunakan perulangan dari 1 hingga `n`. Akhirnya, hasil perhitungan permutasi akan dicetak sebagai output program.

2. Program beserta fungsi yang digunakan untuk menghitung luas dan keliling persegi.

Masukan terdiri dari sisi persegi.

Keluaran berupa hasil luas dan keliling persegi.

#### Sourcecode

```
package main

import "fmt"

// Fungsi untuk menghitung luas persegi
func hitungLuas(sisi int) int {
    return sisi * sisi
}

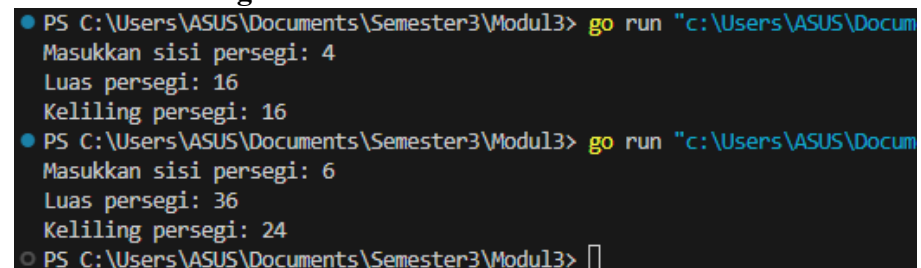
// Fungsi untuk menghitung keliling persegi
func hitungKeliling(sisi int) int {
    return 4 * sisi
}

func main() {
    var sisi int
    fmt.Print("Masukkan sisi persegi: ")
    fmt.Scan(&sisi)

    luas := hitungLuas(sisi)
    keliling := hitungKeliling(sisi)

    fmt.Println("Luas persegi:", luas)
    fmt.Println("Keliling persegi:", keliling)
}
```

#### Screenshoot Program



```
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\ASUS\Documents\Semester3\Modul3\main.go"
Masukkan sisi persegi: 4
Luas persegi: 16
Keliling persegi: 16
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\ASUS\Documents\Semester3\Modul3\main.go"
Masukkan sisi persegi: 6
Luas persegi: 36
Keliling persegi: 24
PS C:\Users\ASUS\Documents\Semester3\Modul3> 
```

### Deskripsi Program

Program di atas bertujuan untuk menghitung luas dan keliling persegi berdasarkan panjang sisi yang diinput oleh pengguna. Pertama, program mendeklarasikan dua fungsi: **'hitungLuas'** untuk menghitung luas persegi dengan rumus  $\text{sisi} \times \text{sisi}$ , dan **'hitungKeliling'** untuk menghitung keliling persegi dengan rumus  $4 \times \text{sisi}$ . Dalam fungsi **'main'**, program meminta pengguna memasukkan nilai panjang sisi persegi dengan fungsi **'fmt.Scan'**. Setelah nilai sisi diperoleh, fungsi **'hitungLuas'** dan **'hitungKeliling'** dipanggil untuk menghitung luas dan keliling berdasarkan nilai sisi tersebut. Hasil perhitungan luas dan keliling kemudian ditampilkan kepada pengguna menggunakan fungsi **'fmt.Println'**. Output program berupa nilai luas dan keliling persegi yang telah dihitung dari input yang diberikan.

### III. UNGUIDED

1. 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"
)

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

// permutation menghitung P(n, r) = n! / (n - r)!
func permutation(n, r int) int {
    if r > n {
        return 0
    }
    return factorial(n) / factorial(n-r)
}

// combination menghitung C(n, r) = n! / (r! * (n - r)!)
```



```

func combination(n, r int) int {
    if r > n {
        return 0
    }
    return factorial(n) / (factorial(r) *
factorial(n-r))
}

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

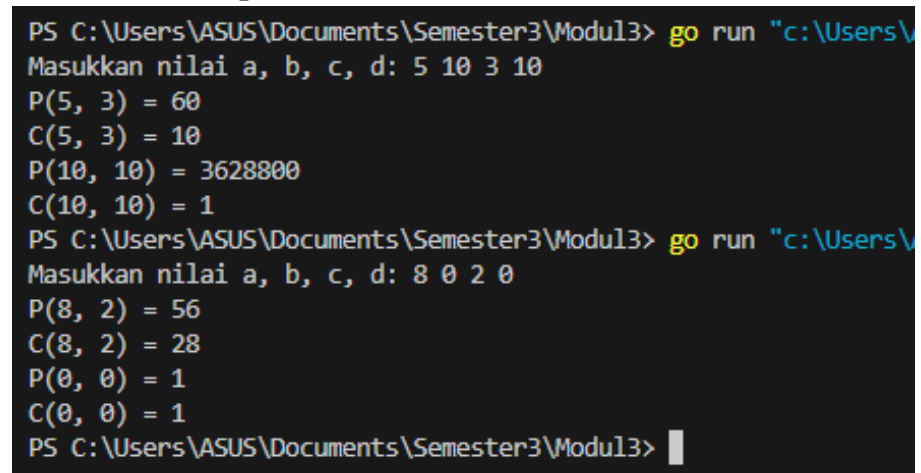
    // Memvalidasi kondisi: a >= c dan b >= d
    if a < c || b < d {
        fmt.Println("Syarat tidak terpenuhi: a <= c
dan b <= d.")
        return
    }

    // Menghitung hasil
    p1 := permutation(a, c)
    c1 := combination(a, c)
    p2 := permutation(b, d)
    c2 := combination(b, d)

    fmt.Printf("P(%d, %d) = %d\n", a, c, p1)
    fmt.Printf("C(%d, %d) = %d\n", a, c, c1)
    fmt.Printf("P(%d, %d) = %d\n", b, d, p2)
    fmt.Printf("C(%d, %d) = %d\n", d, b, c2)
}

```

### Screenshoot Output



```

PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\
Masukkan nilai a, b, c, d: 5 10 3 10
P(5, 3) = 60
C(5, 3) = 10
P(10, 10) = 3628800
C(10, 10) = 1
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\
Masukkan nilai a, b, c, d: 8 0 2 0
P(8, 2) = 56
C(8, 2) = 28
P(0, 0) = 1
C(0, 0) = 1
PS C:\Users\ASUS\Documents\Semester3\Modul3>

```

### Deskripsi Program

Program ini dirancang untuk menghitung permutasi dan kombinasi dari dua pasang bilangan bulat yang dimasukkan oleh pengguna, yaitu

`a`, `b`, `c`, dan `d`. Di dalam fungsi `main`, pengguna diminta untuk memasukkan nilai-nilai tersebut, yang kemudian diperiksa apakah memenuhi syarat `a >= c` dan `b >= d`. Jika syarat ini tidak terpenuhi, program akan menampilkan pesan kesalahan dan berhenti. Jika syarat terpenuhi, program melanjutkan dengan menghitung permutasi  $P(n, r)$  dan kombinasi  $C(n, r)$  menggunakan fungsi `permutation` dan `combination`, yang masing-masing memanggil fungsi `factorial` untuk menghitung faktorial dari bilangan yang diperlukan. Output program berisi nilai permutasi dan kombinasi untuk kedua pasangan bilangan yang dimasukkan.

2. 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  $(h \circ f)(c)$ !

#### 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
}

func main() {
    var a, b, c int

    // Input dari pengguna
    fmt.Print("Masukkan nilai a, b, c: ")
    fmt.Scanf("%d %d %d", &a, &b, &c)
```

```

// Perhitungan untuk komposisi fungsi
fogoh := f(g(h(a))) // (fogoh)(a)
gohof := g(h(f(b))) // (gohof)(b)
hofog := h(f(g(c))) // (hofog)(c)

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

```

### Screenshoot Program

```

PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\
Masukkan nilai a, b, c: 7 2 10
• Hasil (fogoh)(a): 36
  Hasil (gohof)(b): 3
  Hasil (hofog)(c): 65
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\
Masukkan nilai a, b, c: 5 5 5
• Hasil (fogoh)(a): 16
  Hasil (gohof)(b): 24
  Hasil (hofog)(c): 10
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\
• Masukkan nilai a, b, c: 3 8 4
  Hasil (fogoh)(a): 4
  Hasil (gohof)(b): 63
  Hasil (hofog)(c): 5

```

### Deskripsi Program

Program ini bertujuan untuk menghitung hasil komposisi dari tiga fungsi matematika, yaitu  $f(x) = x^2$ ,  $g(x) = x - 2$  dan  $h(x) = x + 1$ , menggunakan tiga nilai integer yang dimasukkan oleh pengguna. Di dalam fungsi `main`, pengguna diminta untuk memasukkan nilai `a`, `b`, dan `c`. Nilai-nilai ini kemudian digunakan untuk menghitung tiga komposisi fungsi:  $(fogoh)(a)$ ,  $(gohof)(b)$ , dan  $(hofog)(c)$ . Komposisi fungsi dilakukan dengan memanggil fungsi secara berurutan: pertama, nilai `a` diproses melalui fungsi `h`, hasilnya melalui fungsi `g`, dan akhirnya melalui fungsi `f` untuk mendapatkan `fogoh`. Proses yang sama dilakukan untuk menghitung `gohof` dan `hofog` dengan nilai `b` dan `c`. Setelah semua perhitungan selesai, hasil dari setiap komposisi fungsi dicetak ke layar, menampilkan hasil akhir dari komposisi fungsi untuk masing-masing nilai yang dimasukkan.

3. 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"**.

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.

#### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung jarak antara dua titik
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 suatu titik
berada di dalam lingkaran
func didalam(cx, cy, r, x, y float64) bool {
    return jarak(cx, cy, x, y) <= r
}

func main() {
    // Input koordinat lingkaran 1
    var cx1, cy1, r1 float64
    fmt.Print("Masukan koordinat titik pusat dan
radius lingkaran 1: ")
    fmt.Scan(&cx1, &cy1, &r1)
```

```

// Input koordinat lingkaran 2
var cx2, cy2, r2 float64
fmt.Print("Masukan koordinat titik pusat dan
radius lingkaran 2: ")
fmt.Scan(&cx2, &cy2, &r2)

// Input titik sembarang
var x, y float64
fmt.Print("Masukan koordinat titik sembarang:
")
fmt.Scan(&x, &y)

// Periksa posisi titik terhadap lingkaran 1
dan lingkaran 2
diDalamLingkaran1 := didalam(cx1, cy1, r1, x,
y)
diDalamLingkaran2 := didalam(cx2, cy2, r2, x,
y)

// Output hasil
if diDalamLingkaran1 && diDalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 1 dan
2")
} else if diDalamLingkaran1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if diDalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan
2")
}
}

```

## Screenshoot Program

```
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\ASUS\Documents\Semester3\Modul3\Unguided\unguided3.go"
Masukan koordinat titik pusat dan radius lingkaran 1: 1 2 3
● Masukan koordinat titik pusat dan radius lingkaran 2: 4 5 6
Masukan koordinat titik sembarang: 7 8
Titik di dalam lingkaran 2
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\ASUS\Documents\Semester3\Modul3\Unguided\unguided3.go"
Masukan koordinat titik pusat dan radius lingkaran 1: 5 10 15
Masukan koordinat titik pusat dan radius lingkaran 2: -15 4 20
Masukan koordinat titik sembarang: 0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\ASUS\Documents\Semester3\Modul3\Unguided\unguided3.go"
Masukan koordinat titik pusat dan radius lingkaran 1: 1 1 5
Masukan koordinat titik pusat dan radius lingkaran 2: 8 8 4
Masukan koordinat titik sembarang: 15 20
Titik di luar lingkaran 1 dan 2
● PS C:\Users\ASUS\Documents\Semester3\Modul3> go run "c:\Users\ASUS\Documents\Semester3\Modul3\Unguided\unguided3.go"
Masukan koordinat titik pusat dan radius lingkaran 1: 1 1 5
Masukan koordinat titik pusat dan radius lingkaran 2: 8 8 4
Masukan koordinat titik sembarang: 2 2
Titik di dalam lingkaran 1
○ PS C:\Users\ASUS\Documents\Semester3\Modul3> 
```

## Deskripsi Program

Program ini dirancang untuk menentukan posisi suatu titik terhadap dua lingkaran berdasarkan koordinat dan radius yang dimasukkan oleh pengguna. Dalam fungsi `main`, pengguna diminta untuk memasukkan koordinat titik pusat dan radius untuk dua lingkaran, serta koordinat titik sembarang yang akan diperiksa. Setelah data dimasukkan, program menggunakan fungsi `didalam` untuk memeriksa apakah titik tersebut berada di dalam masing-masing lingkaran. Fungsi `didalam` memanggil fungsi `jarak`, yang menghitung jarak antara pusat lingkaran dan titik menggunakan rumus  $jarak = \sqrt{(a - c)^2 + (b - d)^2}$ . Jarak ini kemudian dibandingkan dengan radius lingkaran untuk menentukan apakah titik berada di dalam lingkaran atau tidak. Setelah melakukan pemeriksaan untuk kedua lingkaran, program mencetak hasil ke layar, menginformasikan apakah titik tersebut berada di dalam satu, kedua, atau di luar kedua lingkaran berdasarkan hasil pengecekan.

#### **IV. KESIMPULAN**

Fungsi adalah blok kode terorganisir yang dapat dipanggil untuk menjalankan tugas tertentu, memungkinkan pengkodean yang lebih modular dan menghindari pengulangan kode. Penggunaan fungsi dalam pemrograman sangat penting untuk menciptakan kode yang terstruktur, modular, dan efisien. Contoh program di atas menunjukkan bagaimana fungsi dapat digunakan untuk melakukan berbagai perhitungan matematis, termasuk faktorial, permutasi, kombinasi, dan komposisi fungsi, serta untuk menentukan posisi suatu titik terhadap lingkaran. Setiap fungsi dirancang untuk menyelesaikan tugas tertentu, dan dapat dipanggil sesuai kebutuhan dalam program.

#### **V. REFERENSI**

- [1] Novalagung. (n.d.). *Fungsi sebagai parameter*. Retrieved October 12, 2024, from <https://dasarpemrogramangolang.novalagung.com/A-fungsi-sebagai-parameter.html>