

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

FUNGSI



Disusun Oleh :

Agnes Refilina Fiska / 2311102126

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 merupakan kumpulan instruksi yang diberi nama untuk memfasilitasi penggunaannya. Melalui implementasi fungsi yang tepat, kode menjadi lebih modular dan mengadopsi prinsip DRY (Don't Repeat Yourself). Hal ini berarti bahwa kita tidak perlu menulis ulang kode serupa berkali-kali; cukup mendefinisikan fungsi tersebut sekali dan memanggilnya ketika 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 <nama function> (<params>) -> <type></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 <value/variabel></code>
9	<code>endfunction</code>
	Notasi dalam bahasa Go
10	<code>func <nama function> (<params>) <type> {</code>
11	<code> /* deklarasi variabel lokal dari fungsi */</code>
12	<code> ...</code>
13	<code> /* badan algoritma fungsi*/</code>
14	<code> ...</code>
15	<code> return <value/variabel></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) {cara pemanggilan #1}
9	v2 <- volumeTabung(r,t) + volumeTabung(15,t) {cara pemanggilan #2}
10	output(volumeTabung(14,100)) {cara pemanggilan #3}
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) // cara pemanggilan #1
18	v2 = volumeTabung(r,t) + volumeTabung(15,t) // cara pemanggilan #2
19	fmt.Println(volumeTabung(14,100)) // cara pemanggilan #3
20	}

II. GUIDED

1. Guided 1

Soal Studi Case

Contoh program dengan function

Sourcecode

```
package main

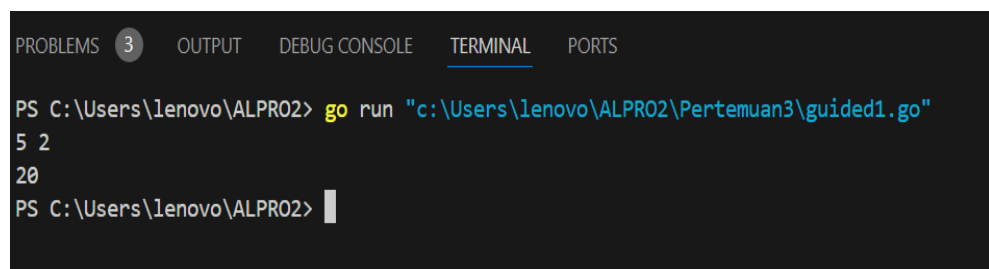
import "fmt"

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 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\ALPRO2> go run "c:\Users\lenovo\ALPRO2\Pertemuan3\guided1.go"
5 2
20
PS C:\Users\lenovo\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. **Deklarasi Variabel:**
 - **a** dan **b** adalah variabel integer yang digunakan untuk menyimpan input bilangan bulat.
2. **Input:**
 - **fmt.Scan(&a, &b):** Kode ini membaca dua bilangan bulat dari input pengguna dan menyimpannya ke dalam variabel **a** dan **b**.
3. **Perbandingan:**
 - **if a >= b:** Kode ini memeriksa apakah **a** lebih besar atau sama dengan **b**. Jika ya, maka program akan memanggil fungsi **permutasi(a, b)**.
 - **else:** Jika **a** lebih kecil dari **b**, maka program akan memanggil fungsi **permutasi(b, a)**.
4. **Fungsi faktorial():**
 - Fungsi ini menghitung faktorial dari bilangan bulat yang diberikan sebagai argumen.
 - **var hasil int = 1:** Deklarasi variabel **hasil** untuk menyimpan hasil faktorial.
 - **for i = 1; i <= n; i++:** Loop **for** yang mengulang dari 1 hingga **n**.
 - **hasil = hasil * i:** Pada setiap iterasi, **hasil** dikalikan dengan **i** untuk menghitung faktorial.
 - **return hasil:** Fungsi mengembalikan nilai faktorial yang dihitung.
5. **Fungsi permutasi():**
 - Fungsi ini menghitung permutasi dari dua bilangan bulat yang diberikan sebagai argumen.
 - **return faktorial(n) / faktorial(n-r):** Fungsi mengembalikan hasil pembagian faktorial **n** dengan faktorial **n-r**.
6. **Output:**
 - **fmt.Println(permutasi(a, b)):** Kode ini mencetak hasil permutasi yang dihitung oleh fungsi **permutasi()**.

Cara Kerja Program:

Program meminta pengguna untuk memasukkan dua bilangan bulat, **a** dan **b**. Kemudian, program memeriksa apakah **a** lebih besar atau sama dengan **b**. Jika ya, maka program akan menghitung permutasi dari **a** dan **b**. Jika tidak, program akan menghitung permutasi dari **b** dan **a**.

Fungsi **permutasi()** menggunakan fungsi **faktorial()**

Contoh Penggunaan:

Jika pengguna memasukkan nilai $a = 5$ dan $b = 2$, program akan menghitung $P(5,2)$ 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"

func main() {
    var panjang, lebar int

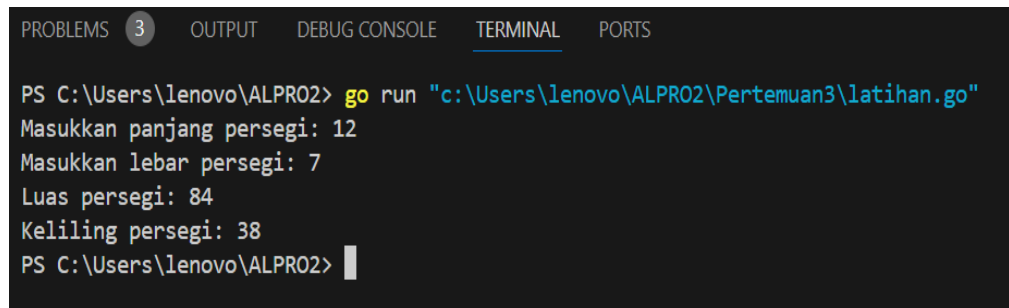
    fmt.Print("Masukkan panjang persegi: ")
    fmt.Scanln(&panjang)

    fmt.Print("Masukkan lebar persegi: ")
    fmt.Scanln(&lebar)

    luas := panjang * lebar
    keliling := 2 * (panjang + lebar)

    fmt.Printf("Luas persegi: %d\n", luas)
    fmt.Printf("Keliling persegi: %d\n", keliling)
}
```

Screenshoot Output

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS (3), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS. The command executed is 'go run "c:\Users\lenovo\ALPRO2\Pertemuan3\latihan.go"'. The output shows the program prompting for the length and width of a square, calculating the area and perimeter, and displaying the results.

```
PS C:\Users\lenovo\ALPRO2> go run "c:\Users\lenovo\ALPRO2\Pertemuan3\latihan.go"
Masukkan panjang persegi: 12
Masukkan lebar persegi: 7
Luas persegi: 84
Keliling persegi: 38
PS C:\Users\lenovo\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. **Deklarasi variabel:**
 - **panjang** dan **lebar** dideklarasikan sebagai variabel integer untuk menyimpan panjang dan lebar persegi.
2. **Input dari pengguna:**
 - Program meminta pengguna untuk memasukkan panjang dan lebar persegi menggunakan **fmt.Print()** dan menyimpan input ke variabel **panjang** dan **lebar** menggunakan **fmt.Scanln()**.
3. **Perhitungan luas:**
 - **luas** dihitung dengan mengalikan **panjang** dan **lebar** menggunakan **luas := panjang * lebar**.
4. **Perhitungan keliling:**
 - **keliling** dihitung dengan mengalikan 2 dengan jumlah **panjang** dan **lebar** menggunakan **keliling := 2 *(panjang +lebar)**.
5. **Menampilkan hasil:**
 - Program menampilkan luas dan keliling persegi menggunakan **fmt.Printf()**.
6. **Output:**
 - Program akan mencetak hasil perhitungan **luas** dan **keliling** dengan format desimal dua angka di belakang koma.

Contoh Penggunaan:

Jika pengguna memasukkan panjang 12 dan lebar 7, 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, d , dan yang dipisahkan oleh spasi, dengan syarat $a \geq b$ dan $c \geq d$. Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi terhadap a, b , sedangkan baris kedua adalah hasil permutasi dan kombinasi terhadap c, d .

Catatan: permutasi (P) dan kombinasi (C) dari terhadap (n, 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 DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\ALPRO2> go run "c:\Users\lenovo\ALPRO2\Pertemuan3\unguided1.go"
Masukkan 4 bilangan: 8 0 2 0
56 28
1 1
PS C:\Users\lenovo\ALPRO2>

```

Deskripsi Program

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menghitung nilai faktorial, permutasi, dan kombinasi dari bilangan bulat yang dimasukkan oleh pengguna. Program ini menggunakan beberapa fungsi untuk melakukan perhitungan matematis yang diperlukan.

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:

- $P(5,3)P(5,2)P(5,1)$ dan $C(5,3)C(5,2)C(5,1)$.
- $P(0,0)P(0,0)P(0,0)$ dan $C(0,0)C(0,0)C(0,0)$.

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

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: ")
    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

```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\ALPRO2> go run "c:\Users\lenovo\ALPRO2\Pertemuan3\unguided2.go"
Masukkan 3 bilangan: 5 5 5
16
24
10
PS C:\Users\lenovo\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 (fungsi $F(X)=X^2$ contoh soal $f(x)=X^2$).
- $g(x)$ mengurangi dengan 2 (fungsi $G(X)=X-2$ contoh soal $=X-2$).
- $h(x)$ menambahkan 1 pada (fungsi $H(X)=X+1$ tinggi $(x)=X+1$).

2. Komposisi Fungsi:

- $fogoh(x)$ adalah komposisi dari fungsi contoh soal $f(x)$. Ini berarti bahwa nilai pertama-tama diproses oleh h , kemudian hasil diproses oleh g , dan akhirnya hasil dari diproses oleh f .
- $gohof(x)$ adalah komposisi dari Bahasa Indonesia: $g(h(f(x)))$.
- $hofog(x)$ adalah komposisi dari Bahasa Indonesia: $h(f(g(x)))$.

3. Proses:

Di dalam fungsinya , program meminta pengguna untuk memasukkan tiga bilangan bulat (, , dan).main abc

Setelah itu, program menghitung dan mencetak hasil dari masing-masing komposisi fungsi menggunakan nilai-nilai yang dimasukkan oleh pengguna.

4. Output:

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

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

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

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 didalam(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 := didalam(cx1, cy1, r1, x, y)
    inCircle2 := didalam(cx2, cy2, r2, x, y)

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

    // Validasi input radius
    for r1 < 0 {
        fmt.Println("Radius tidak boleh negatif.
        Silakan masukkan nilai yang valid.")
        fmt.Scan(&r1)
    }

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

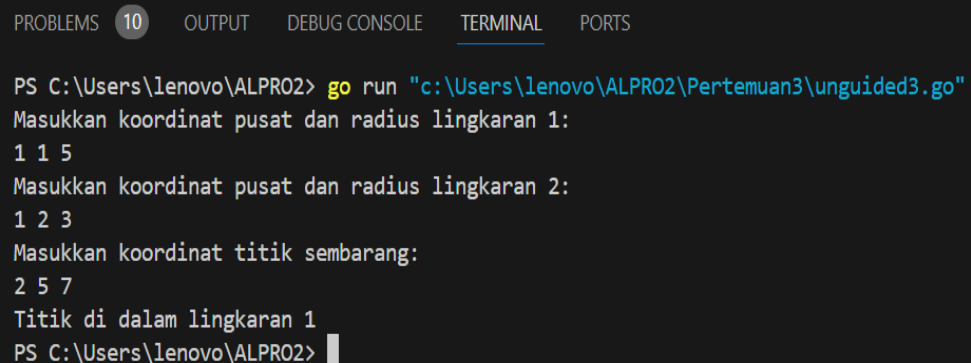
    // Validasi input radius
    for r2 < 0 {
        fmt.Println("Radius tidak boleh negatif.
        Silakan masukkan nilai yang valid.")
        fmt.Scan(&r2)
    }

    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 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\ALPRO2> go run "c:\Users\lenovo\ALPRO2\Pertemuan3\unguided3.go"
Masukkan koordinat pusat dan radius lingkaran 1:
1 1 5
Masukkan koordinat pusat dan radius lingkaran 2:
1 2 3
Masukkan koordinat titik sembarang:
2 5 7
Titik di dalam lingkaran 1
PS C:\Users\lenovo\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.

IV. KESIMPULAN

Setelah belajar tentang fungsi dalam bahasa Go, dapat disimpulkan bahwa fungsi memainkan peran penting dalam membuat kode lebih modular dan mendukung prinsip DRY (Don't Repeat Yourself), sehingga mengurangi berlipat ganda kode yang sama. Fungsi di Go dapat mengembalikan nilai dengan menyatakan jenis nilai yang akan dikembalikan dan menggunakan kata kunci untuk mengirimkannya kembali. Selain itu, fungsi juga bisa memiliki parameter, memungkinkan kita untuk memasukkan data ke dalam fungsi dan memprosesnya. Panggilan fungsi dilakukan dengan memberikan argumen yang sesuai dengan jumlah dan jenis parameter yang sudah didefinisikan. Misalnya, penggunaan fungsi built-in seperti menunjukkan bagaimana fungsi membantu memanipulasi data dengan lebih efektif. Secara keseluruhan, aplikasi fungsi dalam Go sangat penting untuk menjaga kode tetap terstruktur, jernih, dan efisien.`return strings.Join()`.

V. REFERENSI

- [1] Modul 3 Praktikum Algoritma 2
- [2] <https://www.staditek.id/insight/golang-adalah/>
- [3] https://pta.pilkommedia.org/progress/upload/AhmadFaisal_A1C615001_Dasar-DasarBahasaPemrogramanGolang.pdf