

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

FUNGSI



Disusun Oleh :

Muhammad Hamzah Haifan Ma'ruf

2311102091

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

Function atau fungsi dalam bahasa pemrograman merupakan sebuah blok kode atau subprogram yang dirancang untuk menjalankan sebuah tugas tertentu atau sejumlah pernyataan secara terpisah dari bagian utama program. Function atau Fungsi ini digunakan untuk mengorganisasi, mengelompokkan, dan mengabstraksi kode sehingga seorang Backend Developer dapat menggunakannya kembali tanpa harus menulis ulang pernyataan yang sama pada sebuah program yang dibuat

Konsep Function Pada Bahasa Pemrograman Golang

- **Parameter dan Argumen**
Sebuah Function atau Fungsi dapat menerima satu atau lebih parameter sebagai nilai yang dimasukkan atau diinput. Parameter adalah variabel yang digunakan dalam fungsi untuk melakukan operasi. Ketika seorang Backend Developer memanggil function atau fungsi, maka seorang Backend Developer akan memberikan argumen yaitu nilai aktual yang diberikan ke parameter fungsi.
- **Tipe Data Kembalian**
Function atau fungsi dapat mengembalikan hasil atau nilai yang dihitung. Jika fungsi tidak mengembalikan nilai, seorang Backend Developer dapat menggunakan `void` atau tidak menyebutkan tipe data kembalian.
- **Memanggil Function atau Fungsi**
Fungsi dipanggil dengan menyebutkan namanya diikuti oleh argumen dalam tanda kurung. Hasil dari fungsi dapat disimpan dalam variabel atau bisa digunakan secara langsung dalam pernyataan.
- **Fungsi Variadic (Variadic Function)**
Bahasa Pemrograman Golang mendukung parameter variadic, yang memungkinkan seorang Backend Developer untuk mengirim jumlah argumen yang tidak terbatas ke fungsi. Parameter variadic ini ditandai dengan `...`.
- **Fungsi Anonim (Anonym Function)**
Seorang Backend Developer dapat membuat fungsi tanpa nama atau yang lebih dikenal sebagai fungsi anonim (Anonym Function), yang digunakan

dalam ekspresi lambda. Fungsi Anonim (Anonym Function) ini sangat berguna dalam kasus penggunaan singkat.

- **Fungsi Metode (Method Function)**
Dalam Bahasa Pemrograman Golang, seorang Backend Developer dapat memiliki metode yang terkait dengan tipe data kustom sendiri. Metode adalah fungsi yang “terikat” ke tipe data tertentu dan memungkinkan seorang Backend Developer melakukan operasi pada instance dari tipe data tersebut.
- **Fungsi Rekursif (Recursive Function)**
Function atau Fungsi dapat memanggil dirinya sendiri, yang dikenal sebagai rekursi. Fungsi rekursif (Recursive Function) berguna untuk menyelesaikan masalah berulang atau memecahkan secara berulang.
- **Fungsi Closure**
Bahasa Pemrograman Golang mendukung closure, di mana fungsi dalam fungsi lain dapat mengakses variabel yang dideklarasikan di luar lingkup fungsi tersebut.
- **Error Handling**
Function atau Fungsi sering mengembalikan nilai kesalahan (error) untuk mengindikasikan jika operasi fungsi gagal. Hal ini membantu dalam penanganan kesalahan atau error yang baik.
- **Penggunaan Goroutine**
Function atau Fungsi dapat digunakan dalam pembuatan goroutine, yang merupakan fitur konkurensi dari bahasa pemrograman Golang untuk menjalankan tugas secara bersamaan.

II. .GUIDED

1. Guided 1

Soal Studi Case

Contoh program dengan function

Sourcecode

```
package main

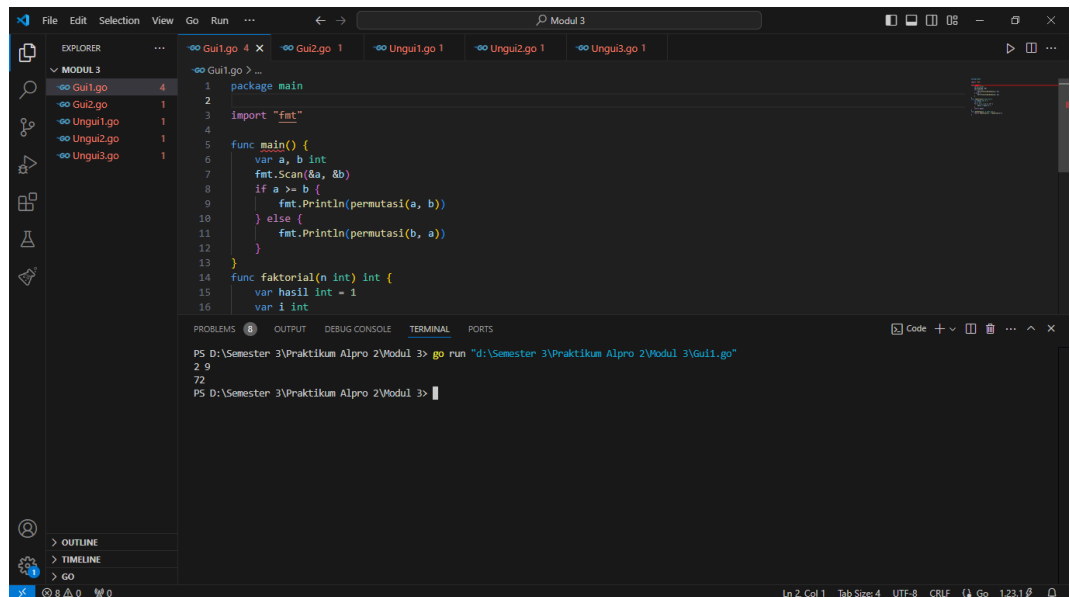
import "fmt"

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



Deskripsi Program

Program diatas dapat menghitung nilai permutasi dari dua angka yang dimasukkan pengguna. Program memulai dengan fungsi main(), meminta dua bilangan bulat dan membandingkannya. Fungsi permutasi() disebut sebagai argumen pertama dan argumen kedua, dan hasil perhitungan dicetak ke layar. Dua fungsi tambahan digunakan oleh program ini. Faktorial() menghitung faktorial sebuah angka dengan menggunakan loop, dan permutasi() menghitung permutasi dengan menggunakan rumus $P(n,r) = n! / (n-r)!$. Program ini menghitung jumlah cara untuk memilih r item dari n item tanpa mengembalikannya dan dengan memperhatikan urutan dengan menggunakan konsep permutasi dalam matematika.

2. Guided 2

Soal Studi Case

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

Sourcecode

```
package main

import "fmt"

func luas_persegi(s int) int {
    luas := s * s
```

```

        return luas
    }

    func keliling_persegi(s int) int {

        keliling := 4 * s
        return keliling

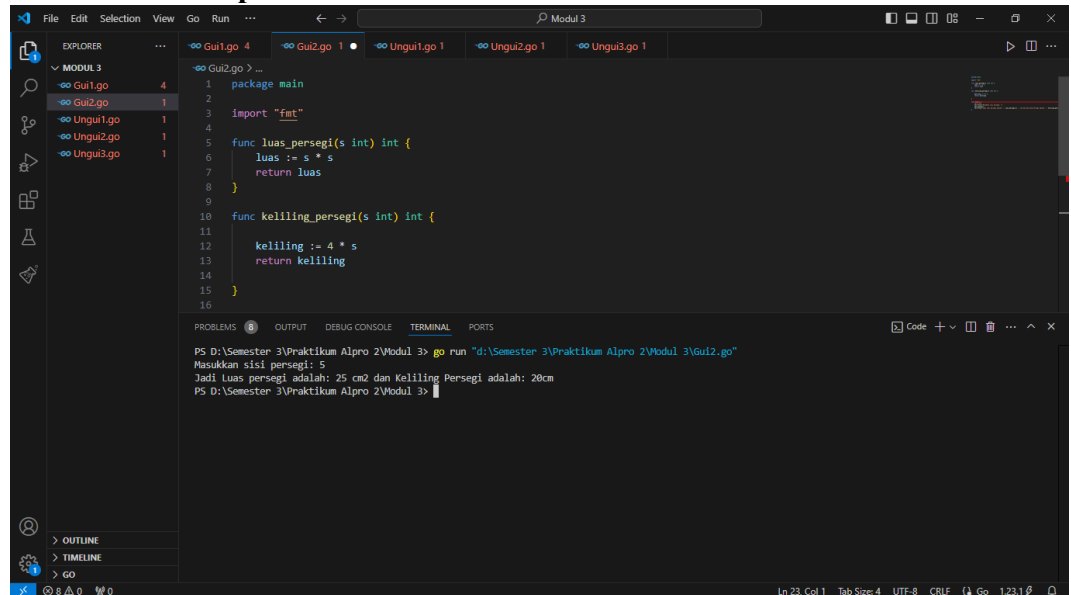
    }

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

        fmt.Print("Jadi Luas persegi adalah: ",
        luas_persegi(s), " cm2 dan Keliling Persegi
        adalah: ", keliling_persegi(s), "cm")
    }

```

Screenshoot Output



Deskripsi Program

Program diatas memiliki tiga fungsi utama program yaitu `main()`, `luas_persegi()`, dan `keliling_persegi`. Fungsi `main()` berfungsi sebagai titik awal program, di mana pengguna diminta untuk memasukkan panjang sisi persegi. Setelah menerima input, program memanggil dua fungsi tambahan

untuk menghitung. Fungsi `luas_persegi()` menghitung luas persegi dengan mengalikan sisi dengan dirinya sendiri ($s * s$), dan fungsi `keliling_persegi()` menghitung keliling persegi dengan mengalikan 4 dengan panjang sisi ($4 * s$). Selanjutnya, hasil perhitungan dicetak ke layar dengan ukuran yang mudah dibaca yang menunjukkan baik luas persegi (dalam cm^2) maupun keliling persegi. Program ini menggunakan pustaka `fmt` untuk operasi input/output; `fmt.Print()` menampilkan prompt dan hasil, dan `fmt.Scan()` menerima input pengguna.

III. UNGUIDED

1. Unguided1

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"
)

// menghitung faktorial
func factorial(n int) int {
    if n == 0 {
        return 1
    }
    result := 1
```

```

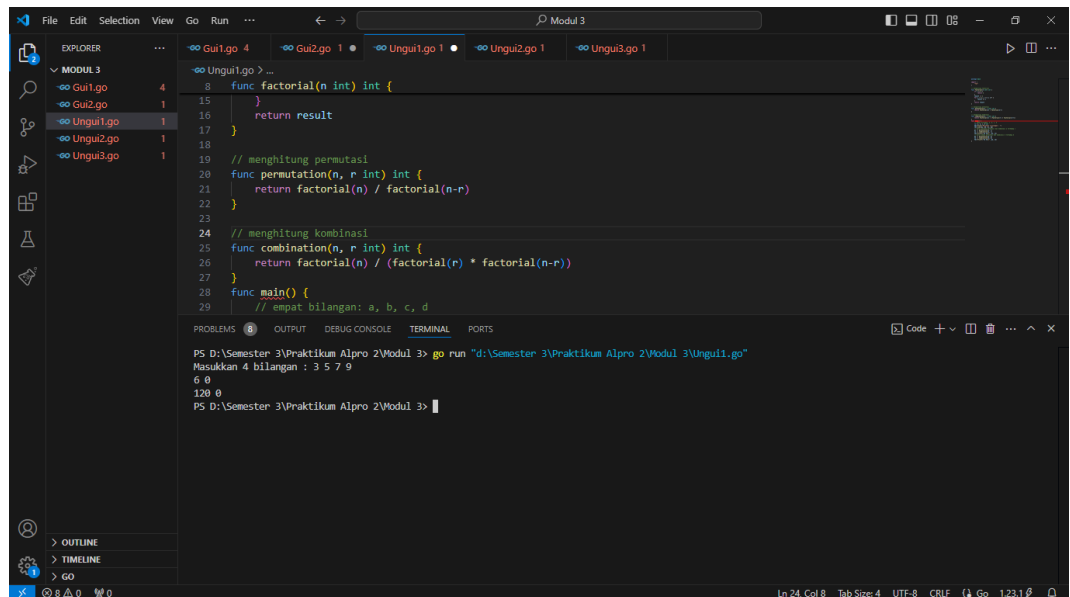
        for i := 1; i <= n; i++ {
            result *= i
        }
        return result
    }

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

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

```

Screenshot Output



```
8 func factorial(n int) int {
15 }
16 return result
17 }
18
19 // menghitung permutasi
20 func permutation(n, r int) int {
21     return factorial(n) / factorial(n-r)
22 }
23
24 // menghitung kombinasi
25 func combination(n, r int) int {
26     return factorial(n) / (factorial(r) * factorial(n-r))
27 }
28
29 func main() {
    // empat bilangan: a, b, c, d
    ...
}
```

PS D:\Semester 3\Praktikum Alpro 2\Modul 3> go run "d:\Semester 3\Praktikum Alpro 2\Modul 3\Ungui1.go"

Masukkan 4 bilangan : 3 5 7 9

60

1200

PS D:\Semester 3\Praktikum Alpro 2\Modul 3>

Deskripsi Program

Program diatas memiliki empat fungsi utama program yaitu factorial(), permutation(), combination(), dan main(). Fungsi factorial menghitung faktorial dari suatu bilangan menggunakan loop, dengan penanganan khusus untuk kasus $n = 0$. Fungsi permutasi dan kombinasi menggunakan hasil perhitungan faktorial untuk menghitung permutasi dan kombinasi sesuai dengan rumus matematika yang berlaku. Dalam fungsi main(), program meminta pengguna memasukkan empat bilangan (a, b, c, dan d) menggunakan input standar. Kemudian, program menghitung dan mencetak hasil permutasi dan kombinasi dalam dua baris, dengan permutasi dan kombinasi a terhadap c dan permutasi dan kombinasi b terhadap d di baris pertama. Untuk operasi input/output, program ini menggunakan pustaka fmt; fmt.Scan() menerima input dan fmt.Printf() mencetak hasil dengan format tertentu.

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"
)

// definisikan fungsi f, g, dan h
func f(x int) int {
    return x * x
}

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

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

// buat fungsi komposisi
func fogoh(x int) int {
    return f(g(h(x)))
}

func gohof(x int) int {
    return g(h(f(x)))
}

func hofog(x int) int {
    return h(f(g(x)))
}
```

```

}

func main() {
    //definisikan inputan
    var a, b, c int

    //input
    fmt.Println("Masukkan 3 angka (a, b, c) : ")
    fmt.Scanf("%d %d %d", &a, &b, &c)

    //output
    fmt.Printf("f(g(h(%d))) = %d\n", a,
fogoh(a))
    fmt.Printf("g(h(f(%d))) = %d\n", b,
gohof(b))
    fmt.Printf("h(f(g(%d))) = %d\n", c,
hofog(c))
}

```

Screenshot Output

The screenshot shows a Go IDE with the following components:

- EXPLORER:** Shows a project structure with files `Gui1.go`, `Gui2.go`, `Ungu1.go`, `Ungu2.go`, and `Ungu3.go`.
- EDITOR:** Displays the source code of `Ungu2.go`, which includes:


```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 // definisikan fungsi f, g, dan h
8 func f(x int) int {
9     return x * x
10 }
11
12 func g(x int) int {
13     return x - 2
14 }
15
16 func h(x int) int {

```
- TERMINAL:** Shows the execution output:


```

PS D:\Semester 3\Praktikum Alpro 2\Modul 3> go run "d:\Semester 3\Praktikum Alpro 2\Modul 3\Ungu2.go"
Masukkan 3 angka (a, b, c) :
1 3 5
f(g(h(1))) = 0
g(h(f(3))) = 8
h(f(g(5))) = 10
PS D:\Semester 3\Praktikum Alpro 2\Modul 3>

```

Deskripsi Program

Program diatas memiliki tiga fungsi dasar yang didefinisikan oleh program ini: $f(x)$ yang mengembalikan kuadrat dari x , $g(x)$ yang mengurangi 2 dari x , dan $h(x)$ yang menambahkan 1 ke x . Selain itu, program mendefinisikan tiga fungsi komposisi, yaitu $fogoh(x)$ sebagai komposisi $f(g(h(x)))$, $gohof(x)$

sebagai komposisi $g(h(f(x)))$, dan $h \circ f \circ g(x)$ sebagai komposisi $h(f(g(x)))$. Dalam fungsi `main()`, program meminta pengguna memasukkan tiga angka (a, b, c) melalui input standar. Kemudian, program menghitung dan mencetak hasil dari ketiga fungsi komposisi, masing-masing menggunakan salah satu angka input. Output ditampilkan dalam format yang mudah dipahami, menunjukkan fungsi komposisi yang digunakan dan hasilnya untuk setiap input.

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((a-c)*(a-c) + (b-d)*(b-d))
}

// fungsi untuk memeriksa apakah sebuah titik
(x, y) berada di dalam lingkaran dengan pusat
(cx, cy) dan jari-jari r
func diDalamLingkaran(x, y, cx, cy, r float64)
bool {
    return jarak(x, y, cx, cy) <= r
}

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

    //input lingkaran 1: pusat (cx1, cy1) dan
    jari-jari r1
    fmt.Println("Masukkan koordinat dan jari-
    jari lingkaran 1 : ")
    fmt.Scan(&cx1, &cy1, &r1)

    //input lingkaran 2: pusat (cx2, cy2) dan
    jari-jari r2
    fmt.Println("Masukkan koordinat dan jari-
    jari lingkaran 2 : ")
    fmt.Scan(&cx2, &cy2, &r2)

    //input titik (x, y)
    fmt.Println("Masukkan titik (x, y) untuk
    diperiksa : ")
    fmt.Scan(&x, &y)

    //periksa apakah titik berada di dalam
    lingkaran 1
    diLingkaran1 := diDalamLingkaran(x, y, cx1,
    cy1, r1)

```

```

        //periksa apakah titik berada di dalam
        lingkaran 2
        diLingkaran2 := diDalamLingkaran(x, y, cx2,
        cy2, r2)

        //tentukan posisi titik
        if diLingkaran1 && diLingkaran2 {
            fmt.Println("Titik berada di dalam
            lingkaran 1 dan 2")
        } else if diLingkaran1 {
            fmt.Println("Titik berada di dalam
            lingkaran 1")
        } else if diLingkaran2 {
            fmt.Println("Titik berada di dalam
            lingkaran 2")
        } else {
            fmt.Println("Titik berada di luar
            lingkaran 1 dan 2")
        }
    }
}

```

Screenshot Output

The screenshot shows a Go IDE with the following components:

- EXPLORER:** Lists files in 'MODUL 3' including `Gui1.go`, `Gui2.go`, `Ungu1.go`, `Ungu2.go`, and `Ungu3.go`.
- EDITOR:** Displays the source code for `Ungu3.go`. The code defines a `main` function that takes coordinates and radii for two circles and a point, then calls `diDalamLingkaran` to check if the point is inside either circle. It uses `fmt` for input/output.
- TERMINAL:** Shows the execution output:


```

PS D:\Semester 3\Praktikum Alpro 2\Modul 3> go run "d:\Semester 3\Praktikum Alpro 2\Modul 3\Ungu3.go"
Masukkan koordinat dan jari-jari lingkaran 1 :
9 7 5
Masukkan koordinat dan jari-jari lingkaran 2 :
1 2 3
Masukkan titik (x, y) untuk diperiksa :
2 4
Titik berada di dalam lingkaran 2
PS D:\Semester 3\Praktikum Alpro 2\Modul 3>

```

Deskripsi Program

Salah satu fungsi utama program adalah `jarak()` dan `diDalamLingkaran()`. Fungsi pertama menggunakan rumus matematika untuk menghitung jarak

geometris antara dua titik, menggunakan fungsi `math.Sqrt()` untuk akar kuadrat, dan fungsi kedua menggunakan hasil dari fungsi `jarak()` untuk menentukan apakah sebuah titik berada di dalam atau pada batas sebuah lingkaran. Dalam fungsi `main()`, program meminta pengguna untuk memasukkan koordinat pusat dan jari-jari dua lingkaran. Program ini memanfaatkan pustaka `fmt` untuk operasi input/output dan `math` untuk fungsi matematika.

IV. KESIMPULAN

Kode-kode program yang telah kita lihat mendemonstrasikan berbagai konsep penting dalam pemrograman menggunakan bahasa Go (Golang). Program-program ini mencakup beragam topik matematika dan geometri, termasuk perhitungan faktorial, permutasi, kombinasi, komposisi fungsi, dan analisis geometri lingkaran.

Secara umum, program-program tersebut menunjukkan:

1. Penggunaan fungsi: Setiap program membagi logika ke dalam fungsi-fungsi yang terpisah dan dapat digunakan kembali, menunjukkan prinsip modularitas dalam pemrograman.
2. Implementasi konsep matematika: Dari perhitungan sederhana seperti luas dan keliling persegi hingga konsep yang lebih kompleks seperti permutasi, kombinasi, dan komposisi fungsi, program-program ini menerjemahkan rumus matematika ke dalam kode yang dapat dieksekusi.
3. Interaksi pengguna: Semua program melibatkan input dari pengguna, mendemonstrasikan penggunaan `fmt.Scan()` dan `fmt.Scanf()` untuk mengambil input, serta `fmt.Print()` dan `fmt.Printf()` untuk menampilkan output.
4. Penggunaan struktur kontrol: Program-program ini menggunakan loop `for` dan pernyataan `if` untuk mengontrol alur eksekusi program.
5. Pemanfaatan pustaka standar: Penggunaan pustaka `fmt` untuk input/output dan `math` untuk fungsi matematika menunjukkan bagaimana memanfaatkan fungsionalitas bawaan Go.
6. Penerapan logika pemrograman: Dari perhitungan sederhana hingga pengecekan kondisi yang lebih kompleks, program-program ini menunjukkan bagaimana menerjemahkan logika ke dalam kode.

Kesimpulannya, kode-kode ini memberikan contoh yang baik tentang bagaimana konsep matematika dan geometri dapat diimplementasikan dalam pemrograman praktis menggunakan Go..

V. REFERENSI

- i. Modul 3 Praktikum Algoritma 2
- ii. Skill, M. (2023, October 16). Function on Golang - my skill - medium. Medium. <https://medium.com/@myskill.id/function-on-golang-27b6577e9cbe>