LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK

MODUL 3

FUNGSI

Oleh:

NAMA : CHRIST DANIEL SANTOSO

NIM : 2311102305

KELAS : IF-11-02

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

DASAR TEORI

Fungsi dalam algoritma pemrograman adalah blok kode mandiri yang dirancang untuk melakukan suatu tugas atau operasi spesifik. Fungsinya dapat dipanggil (dieksekusi) berulang kali di berbagai tempat dalam program, sehingga mengurangi pengulangan kode dan meningkatkan efisiensi. Fungsi biasanya terdiri dari tiga elemen utama: nama fungsi, parameter (argumen), dan nilai kembali (return value). Nama fungsi memungkinkan pemanggilan fungsi dari berbagai tempat dalam kode. Parameter adalah variabel yang memberikan masukan (input) kepada fungsi, sedangkan nilai kembali adalah hasil yang dikembalikan oleh fungsi setelah operasi selesai. Fungsi digunakan untuk memecah masalah besar menjadi bagian-bagian kecil yang lebih terstruktur dan dapat dikelola, yang mendukung prinsip modularitas, di mana program dibagi menjadi modul-modul terpisah yang dapat dikembangkan, diuji, dan diperbaiki secara independen. Hal ini juga memungkinkan abstraksi, di mana detail implementasi tersembunyi, dan pengguna hanya perlu mengetahui cara memanggil fungsi tanpa harus memahami cara kerjanya secara detail. Dalam jangka panjang, penggunaan fungsi mempermudah pemeliharaan, skalabilitas, dan kolaborasi tim dalam pengembangan perangkat lunak.

I. GUIDED

Guided 1

Source Code :

C: > Users > cynzw > OneDrive > Documents > GUIDED MODUL 3 > package main.go > hitungPermutasi

```go
package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x, &y)
    fmt.Println(hitungPermutasi(x, y))
}

func hitungFaktorial(m int) int {
    factorial := 1
    for j := 1; j <= m; j++ {
        factorial *= j
    }
    return factorial
}

func hitungPermutasi(a, b int) int {
    if a < b {
        a, b = b, a
    }
    return hitungFaktorial(a) / hitungFaktorial(a-b)
}
```

Output :

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\GUIDED MODUL 3\package main.go"
5 10
30240
PS C:\Users\cynzw>
```

Penjelasan :

Program Go ini dirancang untuk menghitung permutasi dari dua bilangan bulat yang diberikan oleh pengguna. Program ini terdiri dari tiga fungsi utama: fungsi main yang berfungsi sebagai titik awal eksekusi program, fungsi hitungFaktorial yang digunakan untuk menghitung faktorial dari sebuah bilangan, dan fungsi hitungPermutasi yang digunakan untuk menghitung permutasi dari dua bilangan dengan memanfaatkan hasil dari fungsi hitung Faktorial. Pada fungsi hitungPermutasi, terdapat pengecekan kondisi jika nilai b lebih besar dari a, maka nilai a dan b akan ditukar untuk memastikan perhitungan permutasi dilakukan dengan benar. Hasil akhir berupa nilai permutasi dari dua bilangan yang dimasukkan oleh pengguna kemudian ditampilkan ke layar. Singkatnya, program ini merupakan kalkulator sederhana untuk menghitung permutasi.

Guided 2

Source code:

C: > Users > cynzw > OneDrive > Documents > GUIDED MODUL 3 > guided no 2.go > main

```go
package main

import "fmt"

func hitungFaktorial(x int) int {
    if x == 0 || x == 1 {
        return 1
    }
    return x * hitungFaktorial(x-1)
}
func hitungPermutasi(x, y int) int {
    return hitungFaktorial(x) / hitungFaktorial(x-y)
}
func hitungKombinasi(x, y int) int {
    return hitungFaktorial(x) / (hitungFaktorial(y) * hitungFaktorial(x-y))
}
func main() {
    var p, q, r, s int
    fmt.Println("Masukkan empat bilangan (pisahkan dengan spasi): ")
    fmt.Scanf("%d %d %d %d", &p, &q, &r, &s)
    if p >= r && q >= s {
        perm1 := hitungPermutasi(p, r)
        komb1 := hitungKombinasi(p, r)

        perm2 := hitungPermutasi(q, s)
        komb2 := hitungKombinasi(q, s)

        fmt.Printf("Permutasi(%d, %d) dan Kombinasi(%d, %d): %d, %d\n", p, r, p, r, perm1, komb1)
        fmt.Printf("Permutasi(%d, %d) dan Kombinasi(%d, %d): %d, %d\n", q, s, q, s, perm2, komb2)
    } else {
        fmt.Println("Kondisi p >= r dan q >= s tidak dipenuhi.")
    }
}
```

Output:

```
                > go run "c:\Users\cynzw\OneDrive\Documents\GUIDED MODUL 3\guided no 2.go"
Masukkan empat bilangan (pisahkan dengan spasi):
4 3 2 1
Permutasi(4, 2) dan Kombinasi(4, 2): 12, 6
Permutasi(3, 1) dan Kombinasi(3, 1): 3, 3
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\GUIDED MODUL 3\guided no 2.go"
Masukkan empat bilangan (pisahkan dengan spasi):
1 2 3
Kondisi p >= r dan q >= s tidak dipenuhi.
PS C:\Users\cynzw>
```

Penjelasan :

Program di atas adalah aplikasi sederhana dalam bahasa Go yang menghitung permutasi dan kombinasi dari dua pasangan angka yang dimasukkan oleh pengguna. Fungsi `factorial` berfungsi untuk menghitung faktorial dari sebuah bilangan, sementara fungsi `permutation` dan `combination` memanfaatkan faktorial untuk melakukan perhitungan permutasi dan kombinasi. Di dalam fungsi `main`, program meminta pengguna untuk memasukkan empat bilangan (a, b, c, d). Jika kondisi `a >= c` dan `b >= d` terpenuhi, program akan menghitung dan menampilkan hasil permutasi serta kombinasi dari pasangan (a, c) dan (b, d). Jika tidak, program akan menampilkan pesan kesalahan.

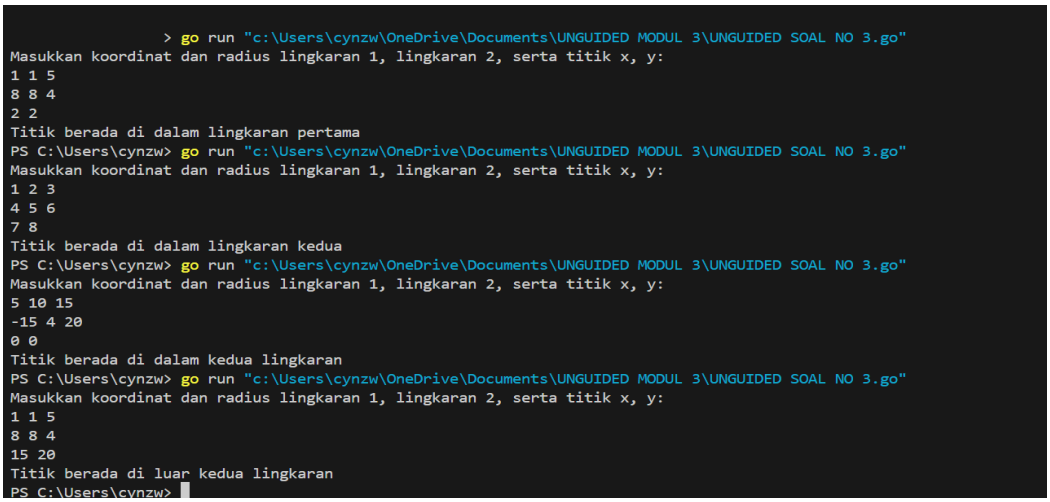
II. UNGUIDED

Source code:

C: > Users > cynzw > OneDrive > Documents > UNGUIDED MODUL 3 > UNGUIDED SOAL NO 3.go > main

```go
package main

import (
    "fmt"
)

func main() {
    var x1, y1, radius1, x2, y2, radius2, px, py int
    fmt.Println("Masukkan koordinat dan radius lingkaran 1, lingkaran 2, serta titik x, y:")
    fmt.Scan(&x1, &y1, &radius1, &x2, &y2, &radius2, &px, &py)

    hasilCekLingkaran(px, py, x1, y1, radius1, x2, y2, radius2)
}
func apakahDalamLingkaran(px, py, pusatX, pusatY, radius int) bool {
    jarakSquared := (px-pusatX)*(px-pusatX) + (py-pusatY)*(py-pusatY)
    return jarakSquared <= radius*radius
}
func hasilCekLingkaran(px, py, cx1, cy1, r1, cx2, cy2, r2 int) {
    diLingkaran1 := apakahDalamLingkaran(px, py, cx1, cy1, r1)
    diLingkaran2 := apakahDalamLingkaran(px, py, cx2, cy2, r2)

    if diLingkaran1 && diLingkaran2 {
        fmt.Println("Titik berada di dalam kedua lingkaran")
    } else if diLingkaran1 {
        fmt.Println("Titik berada di dalam lingkaran pertama")
    } else if diLingkaran2 {
        fmt.Println("Titik berada di dalam lingkaran kedua")
    } else {
        fmt.Println("Titik berada di luar kedua lingkaran")
    }
}

```

Output :



Penjelasan :

Program ini dirancang untuk menentukan posisi sebuah titik relatif terhadap dua lingkaran yang berbeda. Program terlebih dahulu menerima input berupa koordinat pusat dan jari-jari kedua lingkaran, serta koordinat titik yang akan dievaluasi. Setelah itu, program menggunakan fungsi cekDalamLingkaran untuk menghitung jarak antara titik dan pusat masing-masing lingkaran. Berdasarkan hasil perhitungan, program akan menentukan apakah titik tersebut berada di dalam kedua lingkaran, hanya di salah satu lingkaran, atau di luar keduanya. Fungsi cekDalamLingkaran menggunakan rumus jarak Euclidean untuk menghitung jarak dan membandingkannya dengan jari-jari lingkaran guna menentukan letak titik.berhasil memasukkan urutan warna yang benar dalam salah satu percobaan, program akan mencetak "true" sebagai hasil akhir, yang menandakan bahwa urutan warna yang benar telah ditemukan. Sebaliknya, jika dalam semua percobaan urutan warna yang dimasukkan selalu salah, program akan mencetak "false". Program ini mengilustrasikan penggunaan perulangan (loop) dan kondisi (if-else) dalam bahasa pemrograman Go untuk menyelesaikan suatu tugas yang berulang.