## LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

**MODUL III**

**FUNGSI**



Oleh:

NAMA : KARTIKA PRINGGO HUTOMO NIM : 2311102196

KELAS ; IF-11-02

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. **DASAR TEORI**

Fungsi adalah sekelompok pernyataan yang dikelompokkan dan diberi nama sehingga Anda dapat memanggilnya kapan pun diperlukan tanpa harus menulis ulang kode Anda.

Fungsi memungkinkan pemrogram untuk memecah program besar menjadi bagian-bagian yang lebih kecil dan dapat digunakan kembali.

Fungsi terdiri dari tiga komponen utama:

- Nama fungsi: Pengidentifikasi atau nama yang digunakan untuk mengenali dan memanggil fungsi.

- Parameter (opsional): Nilai yang dapat ditetapkan ke fungsi untuk digunakan dalam pemrosesan .

- Nilai kembalian (opsional): Hasil akhir yang dikembalikan oleh fungsi setelah eksekusi.

Semua fungsi dalam bahasa pemrograman biasanya memiliki struktur dasar yang terdiri dari:

- Deklarasi fungsi: mencakup nama, parameter (jika ada), dan tipe data kembalian (jika ada).

- Blok kode: Badan fungsi ini terdiri dari serangkaian instruksi yang dieksekusi ketika suatu fungsi dipanggil.

- Pernyataan pengembalian: (Opsional) Nilai dikembalikan ke pemanggil fungsi ketika fungsi mengembalikan hasil.

## GUIDED

**Guided 1**

Source code

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

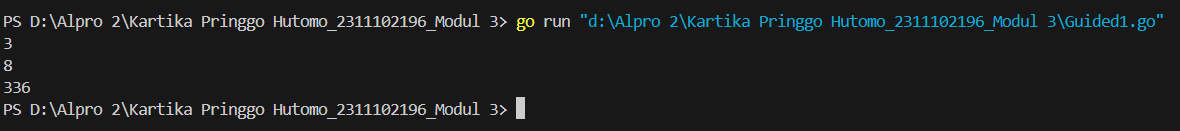
}

func permutasi (n,r int) int {

return faktorial(n) / faktorial(n-r)

}

Output :



Penjelasan

## Tujuan dari program ini adalah untuk menghitung permutasi dua angka yang dimasukkan oleh pengguna.

## Permutasi adalah susunan beberapa elemen yang diambil sebagian atau seluruhnya dari suatu himpunan.

## Program pertama-tama meminta dua bilangan bulat 'a' dan 'b'.

## Jika 'a' lebih besar atau sama dengan 'b', program menghitung permutasi 'a P b'.

## Sebaliknya, jika 'b' lebih besar dari 'a', program akan menghitung 'b P a'.

## Perhitungan permutasi dilakukan dengan menggunakan fungsi "permutasi(n, r)".

## Ini menggunakan rumus permutasi: \( P(n, r) = \frac{n!

## }{(n - r)!

## } \).

## Ini juga mencakup fungsi ``faktorial(n)'' yang menghitung faktorial suatu bilangan secara berulang.

## Program akan memastikan bahwa nilai terbesar adalah 'n' dan nilai yang lebih kecil adalah 'r' sehingga perhitungannya memenuhi persyaratan penggantian.

## Setelah dilakukan perhitungan, hasilnya akan ditampilkan di layar yang menunjukkan berapa banyak elemen yang dapat dibentuk dari kedua bilangan tersebut.

## Guided 2

Source code

package main

import "fmt"

// Fungsi buat menghitung faktorial

func factorial(n int) int {

    if n == 0 || n == 1 {

        return 1

    }

    result := 1

    for i := 2; i <= n; i++ {

        result \*= i

    }

    return result

}

// Fungsi buat menghitung permutasi

func permutation(n, r int) int {

    return factorial(n) / factorial(n-r)

}

// Fungsi buat menghitung kombinasi

func combination(n, r int) int {

    return factorial(n) / (factorial(r) \* factorial(n-r))

}

func main() {

    // Input 4 bilangan

    var a, b, c, d int

    fmt.Println("Masukkan bilangan a, b, c, d (dengan spasi): ")

    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    // Cek syarat a >= c dan b >= d

    if a >= c && b >= d {

        // Menghitung permutasi dan kombinasi a dan c

        permutasiAC := permutation(a, c)

        kombinasiAC := combination(a, c)

        // Menghitung permutasi dan kombinasi b dan d

        permutasiBD := permutation(b, d)

        kombinasiBD := combination(b, d)

        // Output hasil

        fmt.Println("Permutasi(a, c) dan Kombinasi(a, c):", permutasiAC, kombinasiAC)

        fmt.Println("Permutasi(b, d) dan Kombinasi(b, d):", permutasiBD, kombinasiBD)

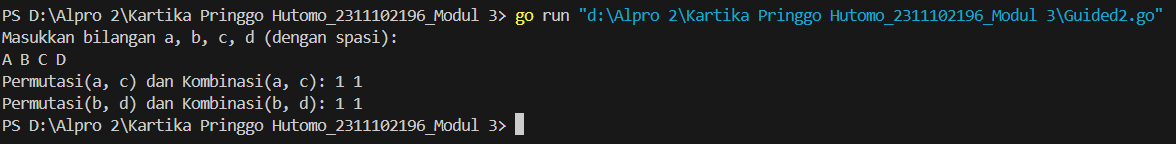
    } else {

        fmt.Println("Syarat a >= c dan b >= d tidak terpenuhi.")

    }

}

Output



Penjelasan

Program ini menghitung permutasi dan kombinasi dua set angka yang dimasukkan oleh pengguna: "(a, c)" dan "(b, d)".

Program ini meminta pengguna untuk memasukkan empat angka: "a", "b", "c", dan "d".

Program kemudian memeriksa apakah nilai 'a' lebih besar atau sama dengan 'c' dan 'b' lebih besar atau sama dengan 'd'.

Jika kondisi ini terpenuhi, program akan menggunakan rumus substitusi \(P(n, r) = \frac{n!

}{(n - r)!

}\) dan kombinasi \(C ( n, r ) = \ frac {n!

}{r!

(n - r)!

}\).

Fungsi faktorial digunakan untuk menghitung faktorial dari bilangan-bilangan tersebut.

Jika kondisi tidak terpenuhi, program akan menampilkan pesan yang menunjukkan bahwa kondisi tersebut tidak valid.

Kemudian menampilkan hasil penghitungan permutasi dan kombinasi kedua kumpulan angka tersebut.

Program ini cocok untuk menghitung penempatan dan pemilihan elemen berdasarkan serangkaian kondisi.

## UNGUIDED

### Source cod

package main

import (

    "fmt"

    "math"

)

func jarak(x1, y1, x2, y2 float64) float64 {

    return math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))

}

func dalamLingkaran(cx, cy, r, x, y float64) bool {

    return jarak(cx, cy, x, y) <= r

}

func main() {

    var cx1, cy1, r1, cx2, cy2, r2, x, y float64

    fmt.Print("Masukkan pusat dan radius lingkaran 1 (cx1, cy1, r1): ")

    fmt.Scan(&cx1, &cy1, &r1)

    fmt.Print("Masukkan pusat dan radius lingkaran 2 (cx2, cy2, r2): ")

    fmt.Scan(&cx2, &cy2, &r2)

    fmt.Print("Masukkan koordinat titik (x, y): ")

    fmt.Scan(&x, &y)

    dalamL1 := dalamLingkaran(cx1, cy1, r1, x, y)

    dalamL2 := dalamLingkaran(cx2, cy2, r2, x, y)

    if dalamL1 && dalamL2 {

        fmt.Println("Titik di dalam lingkaran 1 dan 2")

    } else if dalamL1 {

        fmt.Println("Titik di dalam lingkaran 1")

    } else if dalamL2 {

        fmt.Println("Titik di dalam lingkaran 2")

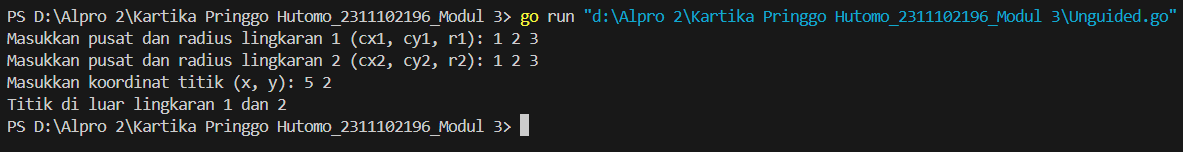
    } else {

        fmt.Println("Titik di luar lingkaran 1 dan 2")

    }

}

### Output :

****

**Penjelasan**

Program ini bertujuan untuk menentukan apakah suatu titik berada di dalam salah satu atau kedua lingkaran berdasarkan koordinat pusat lingkaran, jari-jari, dan koordinat titik tersebut.

Sebagai masukan, program menerima dari pengguna koordinat pusat dan jari-jari kedua lingkaran serta koordinat titik yang akan diperiksa.

Fungsi "jarak" menghitung jarak Euclidean antara pusat lingkaran dan titik yang diperiksa, dan fungsi "inCircle" memeriksa apakah titik tersebut berada di dalam lingkaran atau di tepi lingkaran.

Program kemudian menggunakan kondisi if-else untuk menentukan apakah titik tersebut berada di dalam kedua lingkaran, hanya satu lingkaran, atau di luar kedua lingkaran, dan menampilkan hasilnya di layar.