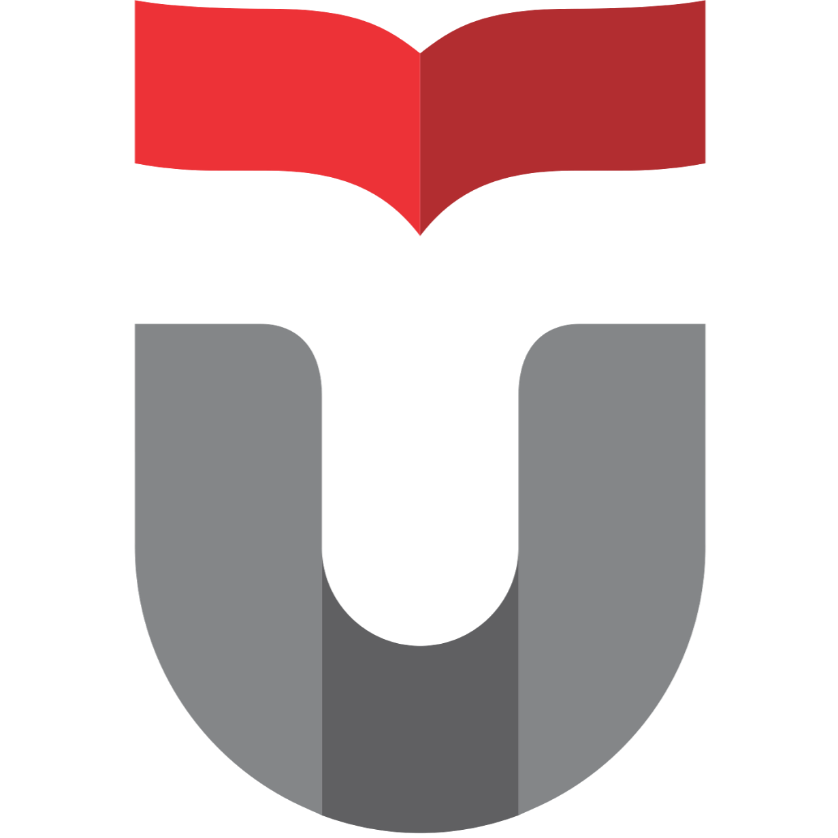
**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL III**

 **FUNGSI**

Oleh:

AULIA RADIX PUTRA WINARKO

2311102056

S1IF-11-02

**S1 TEKNIK INRMATIKA**

**INSTITUT TEKNOLOGI TLKOM PURWOKERTO**

**2024**

1. **DASAR TEORI**
2. **Definisi Function**

Fungsi merupakan satu kesatuan rangkaian instruksi yang memberikan atau menghasilkan suatu nilai dan biasanya memetakkan input ke suatu nilai yang lain. Oleh karena itu, fungsi selalu menghasilkan/mengembalikan nilai. Suatu subprogram dikatakan fungsi apabila:

1. deklarasi tipe nilai yang dikembalikan, dan
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.

Karena itu selalu pilih nama fungsi yang menggambarkan nilai, seperti kata benda dan kata sifat. Contoh nama-nama fungsi: median, rerata, nilalTerbesar, ketemu, selesal, ...

1. **Deklarası Function**

Pada bagian deklarasi terlihat setelah parameter terdapat tipe data dari nilat yang dikembalikan, sedangkan pada bagian badan fungsi terdapat return dari nilal yang dikembalikan.

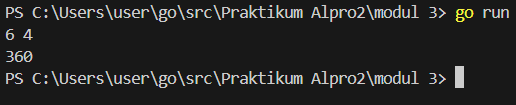
1. **Cara Pemanggilan Function**

Sama halnya dengan prosedur, dengan prosedur, pemanggilan fungsi cukup dilakukan dengan penulisan nama fungsi beserta argumen yang diminta oleh parameter dari fungsi. Perbedaannya dengan prosedur adalah fungsi bisa di-assign ke suatu variabel, menjadi bagian dari ekspresi, dan argumen dari suatu subprogram.

Pada contoh pemanggilan fungsi di atas terlihat tidak ada perbedaan pada saat pemanggilan fungsi pada pseudocode ataupun Golang. Di sini terlihat fungsi bisa di-assign ke suatu variabel pada saat pemanggilan, bisa dioperasikan sesuai dengan tipe data yang dikembalikan, dan juga bisa langsung. ditampilkan dengan perintah output ataupun print.

1. **GUIDED**
2. **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))  } |

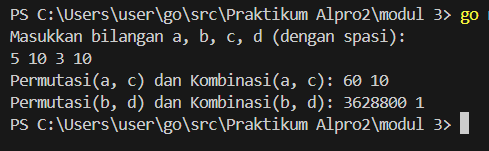
**Output****Keterangan**

Program di atas mengoprasikan dua bilangan dari inputan (a dan b), dengan bilangan terbesar dijadikan sebagai n dan yang bilangan kecil sebagai r, lalu dihitung dengan rumus permutasi yang ada

1. **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**

****

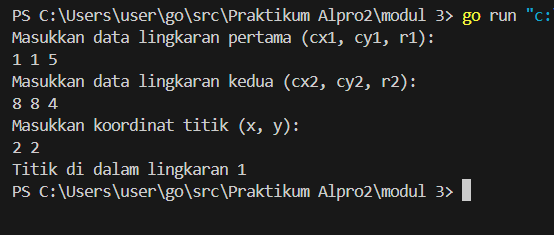
**Keterangan**

Program diatas menerima input 4 bilangan dari user dan menghitung permutasi untuk dua pasang bilangan (a, c) dan (b, d), jika syarat a >= c dan b >= d terpenuhi. Jika tidak, program menampilkan pesan bahwa syarat tidak dipenuhi.

1. **UNGUIDED**
2. **Source Code**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func distance(a, b, c, d float64) float64 {      return math.Sqrt((a-c)\*(a-c) + (b-d)\*(b-d))  }  func isInside(cx, cy, r, x, y float64) bool {      return distance(cx, cy, x, y) <= r  }  func main() {        var cx1, cy1, r1, cx2, cy2, r2 float64      fmt.Println("Masukkan data lingkaran pertama (cx1, cy1, r1):")      fmt.Scan(&cx1, &cy1, &r1)      fmt.Println("Masukkan data lingkaran kedua (cx2, cy2, r2):")      fmt.Scan(&cx2, &cy2, &r2)        var x, y float64      fmt.Println("Masukkan koordinat titik (x, y):")      fmt.Scan(&x, &y)        inCircle1 := isInside(cx1, cy1, r1, x, y)      inCircle2 := isInside(cx2, cy2, r2, x, y)        if inCircle1 && inCircle2 {          fmt.Println("Titik di dalam lingkaran 1 dan 2")      } else if inCircle1 {          fmt.Println("Titik di dalam lingkaran 1")      } else if inCircle2 {          fmt.Println("Titik di dalam lingkaran 2")      } else {          fmt.Println("Titik di luar lingkaran 1 dan 2")      }  } |

**Output**



**Keterangan**

Program diatas memeriksa apakah sebuah titik berada di dalam atau di luar dua lingkaran. Program diatas menerima input dari user berupa data dua lingkaran dan titik yang ingin diperiksa, kemudian dihitung jarak antara titik tersebut dan pusat lingkaran untuk memutuskan apakah titik tersebut berada di dalam atau di luar lingkaran.