**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III**

**FUNGSI**

****

**DISUSUN OLEH :**

**WILDAN MAULANA ZIDAN**

**2311102162**

**IF 11 02**

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

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

Fungsi Merupakan satu kesatuan rangkaian intruksi yang memberikan suatu nilai dan biasanaya memetakan input ke suatu nilai yang lain, fungsi selalu menghasilkan atau mengembalikan nilai. Fungsi dapat digunkan apabila jika suatu nilai biasanya diperlukan seperti: assignment nilai ke suatu variabel dan bagian dari ekspersi.

* 1. **Deklarasi Function**

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

|  |
| --- |
| **Notasi Algoritma** |
| function <nama function> (<params>) -> <type>  kamus  {deklarasi variabel lokal dari fungsi}  algoritma  {badan algoritma fungsi}  return <value/variabel>  endfunction |
| **Notasi dalam go** |
| func <nama function> (<params>) <type> {  /\* deklarasi variabel lokal dari fungsi \*/  /\* badan algoritma fungsi\*/  return <value/variabel>  } |

* 1. **Pemanggilan Function**

Pemanggilan fungsi cukup dilakukan dengan penulisan nama fungsi

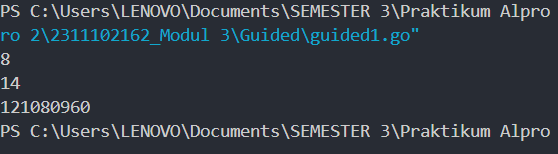
berserta argument yang diminta oleh parameter dari fungsi.

|  |
| --- |
| Hasil := namaFunc (argumen1, argumen2) |

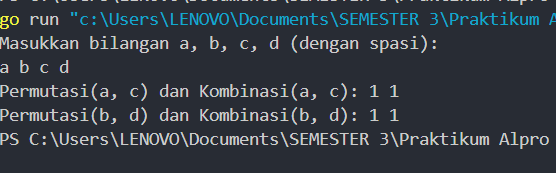
1. **Guided**

Guided 1

|  |
| --- |
| 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)  } |

program di atas meminta dua bilangan bulat "a" dan "b". Setelah menerima masukan, program membandingkan kedua nilai tersebut. Jika "a" lebih besar dari "b", program akan menghitung dan mencetak keluaran "a`P`b", tetapi jika "b" lebih besar, program akan menghitung "b`P "a". . Fungsi "bagi" digunakan untuk menghitung nilai faktorial suatu bilangan, dan fungsi "tentukan" menghitung nilai faktorial menggunakan rumus "n"! /(tidak)!  
  
Guided 2

|  |
| --- |
| package main  import "fmt"  func factorial(n int) int {  if n == 0 || n == 1 {  return 1  }  result := 1  for i := 2; i <= n; i++ {  result \*= i  }  return result  }  func permutation(n, r int) int {  return factorial(n) / factorial(n-r)  }  func combination(n, r int) int {  return factorial(n) / (factorial(r) \* factorial(n-r))  }  func main() {  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)  if a >= c && b >= d {  permutasiAC := permutation(a, c)  kombinasiAC := combination(a, c)  permutasiBD := permutation(b, d)  kombinasiBD := combination(b, d)  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.")  }  } |



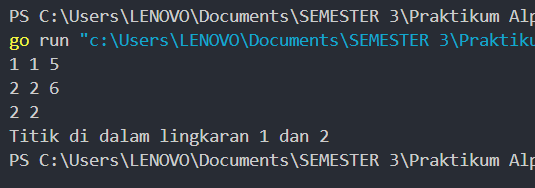
program di atas meminta input empat bilangan bulat "a", "b", "c" dan "d" lalu memeriksa apakah "a>= c" dan "b>= d ". Jika kondisi ini terpenuhi, program menghitung nilai yang dimodifikasi dan digabungkan untuk pasangan "(a, c)" dan "(b, d)" menggunakan fungsi "transisi" dan "kombinasi". Fungsi "faktor" digunakan untuk menghitung faktor dari kedua fungsi tersebut. Jika kondisi tidak terpenuhi, program akan mencetak pesan kondisi tidak terpenuhi.

1. **UNGUIDED**

Unguided 3

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.

|  |
| --- |
| package main  import (  "fmt"  "math"  )  func jarak(a, b, c, d float64) float64 {  return math.Sqrt(math.Pow(a-c, 2) + math.Pow(b-d, 2))  }  func didalam(cx, cy, r, x, y float64) bool {  return jarak(x, y, cx, cy) <= r  }  func main() {  var segiR = [2][3]float64{}  var x, y float64  var cekR = [2]bool{}  for i := 0; i < 2; i++ {  fmt.Scan(&segiR[i][0], &segiR[i][1], &segiR[i][2])  }  fmt.Scan(&x, &y)  for z := 0; z < 2; z++ {  cekR[z] = didalam(segiR[z][0], segiR[z][1], segiR[z][2], x, y)  cekR[z] = didalam(segiR[z][0], segiR[z][1], segiR[z][2], x, y)  }  if cekR[0] && cekR[1] {  fmt.Println("Titik di dalam lingkaran 1 dan 2")  } else if cekR[0] {  fmt.Println("Titik di dalam lingkaran 1")  } else if cekR[1] {  fmt.Println("Titik di dalam lingkaran 2")  } else {  fmt.Println("Titik di luar lingkaran")  }  } |



Kode di atas menghitung apakah suatu titik "(x, y)" berada di dalam salah satu atau kedua lingkaran. Kedua lingkaran tersebut ditentukan oleh matriks "segiR", yang masing-masing berisi koordinat pusat lingkaran "(cx, cy)" dan jari-jari "r". Fungsi “jarak” digunakan untuk menghitung jarak antara suatu titik dengan pusat lingkaran, sedangkan fungsi “dalam” menentukan apakah suatu titik berada di dalam lingkaran berdasarkan jarak dan jari-jarinya. Setelah pengguna memasukkan dua lingkaran dengan simbol, program memeriksa apakah simbol tersebut berada di dalam lingkaran 1, lingkaran 2, keduanya, atau di luar keduanya, dan mencetak hasilnya sesuai kondisi.