**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III**

**FUNGSI**

**Sebuah gambar berisi logo, simbol, Grafis, Font

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Avrizal Setyo Aji Nugroho**

**2311102145**

**IF-11-05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. **DASAR TEORI**

Dalam pemrograman Go (Golang), fungsi adalah blok kode yang didefinisikan dengan kata kunci func dan dimaksudkan untuk melakukan tugas tertentu. Fungsi memungkinkan pengorganisasian kode yang lebih baik, memudahkan penggunaan kembali, dan meningkatkan keterbacaan. Nama fungsi, daftar parameter dengan tipenya, dan tipe pengembalian adalah indikasi dasar untuk mendefinisikan fungsi. Fungsi dapat dikonfigurasi dengan parameter bertipe int dan mengembalikan satu atau lebih nilai. Sebuah contohnya adalah fungsi yang dapat menjumlahkan dua angka dan mengembalikan nilai bertipe int.

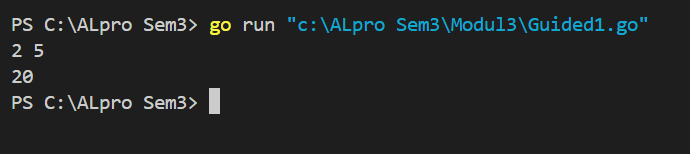
Go juga mendukung fungsi anonim, yaitu fungsi tanpa nama yang dapat disimpan dalam variabel, dan fungsi variadic, yaitu fungsi yang dapat menerima banyak argumen dari jenis yang sama. Selain itu, fungsi yang didefer akan dieksekusi setelah fungsi induknya selesai. Pengembang dapat membuat kode yang lebih modular, terstruktur, dan mudah dipelihara dengan memanfaatkan fungsi. Ini memudahkan pengembangan aplikasi yang lebih kompleks.

1. **GUIDED**
   * + 1. Berikut ini adalah contoh penulisan fungsi pada suatu program lengkap. Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilai faktorial dan permutasi. Masukan terdiri dari dua buah bilangan positif a dan b. Keluaran berupa sebuah bilangan bulat yang menyatakan nilai a permutasi b apabila a≥ b atau b pemutasi a untuk kemungkinan yang lain.

**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)  } |

**Screenshoot Output**

****

**Deskripsi Program**

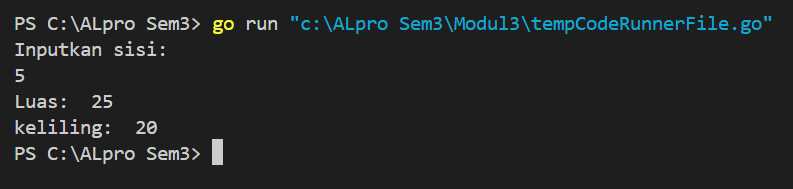
Program ini menerima dua bilangan bulat **a** dan **b** sebagai input, kemudian menghitung dan mencetak hasil permutasi dari dua angka tersebut. Permutasi dihitung dengan rumus **( P(n, r) = n!/(n-r)! )**, di mana **n** adalah angka yang lebih besar atau sama dengan **r**. Jika input **a** lebih besar atau sama dengan **b**, maka program menghitung permutasi **P(a, b)**,jika tidak, maka program menghitung **P(b, a)**.Fungsi **faktorial** digunakan untuk menghitung faktorial dari angka, yang kemudian digunakan dalam fungsi **permutasi** untuk menyelesaikan perhitungan permutasi.

* + - 1. Membuat progam mencari Luas dan Keliling Persegi

**Sourcecode**

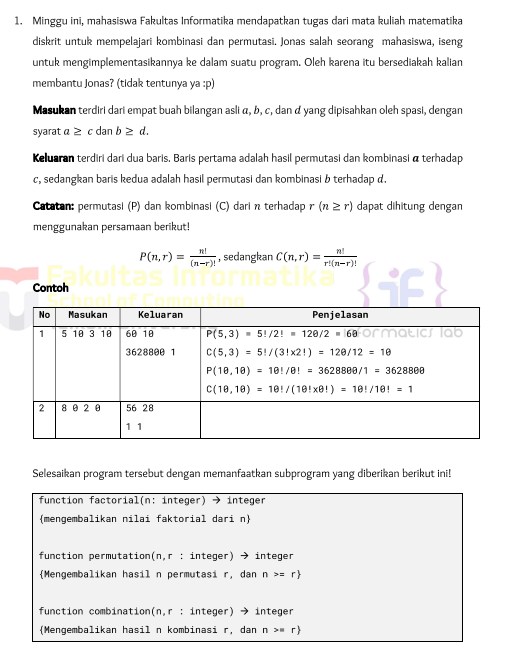
|  |
| --- |
| package main  import (      "fmt"  )  func Luas(s int) int{      var L int      L = s \* s      return L  }  func Keliling(s int) int{      var K int      K = s+s+s+s      return K  }  func main(){      var s int      fmt.Println("Inputkan sisi: ")      fmt.Scan(&s)      fmt.Println("Luas: ",Luas(s))      fmt.Println("keliling: ",Keliling(s))  } |

**Screenshoot Output**

****

**Deskripsi Program**

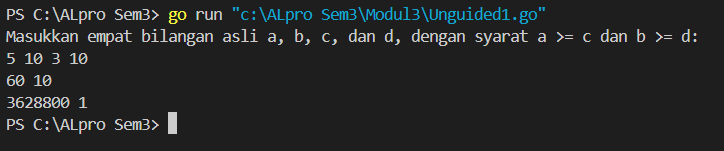
Program di atas menghitung luas dan keliling sebuah persegi berdasarkan input panjang sisi yang diberikan oleh pengguna. Fungsi **Luas(s int)** digunakan untuk menghitung luas persegi dengan rumus **(L= s \* s)**, sedangkan fungsi **Keliling(s int)** menghitung keliling persegi dengan rumus **( K = 4 \* s)**. Setelah pengguna memasukkan panjang sisi melalui input, program akan memanggil kedua fungsi tersebut dan mencetak hasil luas dan keliling persegi ke layar.

1. **UNGUIDED**
   * + 1. Unguided 1

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func factorial(n\_145 int) int {      if n\_145 == 0 {          return 1      }      return n\_145 \* factorial(n\_145-1)  }  func permutation(n\_145, r int) int {      if n\_145 < r {          return 0      }      return factorial(n\_145) / factorial(n\_145-r)  }  func combination(n\_145, r int) int {      if n\_145 < r {          return 0      }      return factorial(n\_145) / (factorial(r) \* factorial(n\_145-r))  }  func main() {      var a, b, c, d int      fmt.Println("Masukkan empat bilangan asli a, b, c, dan d, dengan syarat a >= c dan b >= d:")      fmt.Scanln(&a, &b, &c, &d)      p1 := permutation(a, c)      c1 := combination(a, c)      p2 := permutation(b, d)      c2 := combination(b, d)      fmt.Println(p1, c1)      fmt.Println(p2, c2)  } |

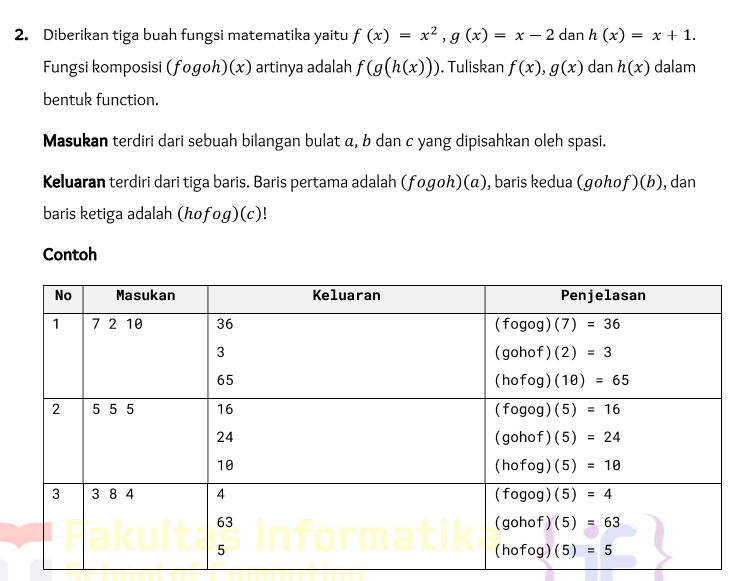
**Screenshoot Output**

****

**Deskripsi Program**

Program di atas menerima empat bilangan bulat positif **a**, **b**, **c**, dan **d** dengan syarat bahwa **a >= c** dan **b >= d**, kemudian menghitung serta mencetak hasil permutasi dan kombinasi dari pasangan bilangan tersebut. Permutasi dihitung dengan rumus **( P(n, r) = n!/(n-r)!)**, sedangkan kombinasi dihitung dengan rumus **( C(n, r) = n! /r!(n-r)! )**. Program ini menggunakan fungsi rekursif `factorial` untuk menghitung faktorial, serta fungsi ***permutation*** dan ***combination*** untuk menghitung permutasi dan kombinasi. Setelah itu, hasil dari perhitungan permutasi dan kombinasi untuk kedua pasangan angka tersebut dicetak sebagai output.

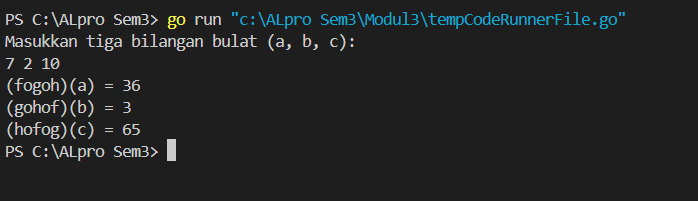
* + - 1. **Soal Studi Case**



**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func f(x\_145 int) int {      return x\_145 \* x\_145  }  func g(x\_145 int) int {      return x\_145 - 2  }  func h(x\_145 int) int {      return x\_145 + 1  }  func fogoh(x\_145 int) int {      return f(g(h(x\_145)))  }  func gohof(x\_145 int) int {      return g(h(f(x\_145)))  }  func hofog(x\_145 int) int {      return h(f(g(x\_145)))  }  func main() {      var a, b, c int      fmt.Println("Masukkan tiga bilangan bulat (a, b, c):")      fmt.Scanln(&a, &b, &c)      fmt.Println("(fogoh)(a) =", fogoh(a))      fmt.Println("(gohof)(b) =", gohof(b))      fmt.Println("(hofog)(c) =", hofog(c))  } |

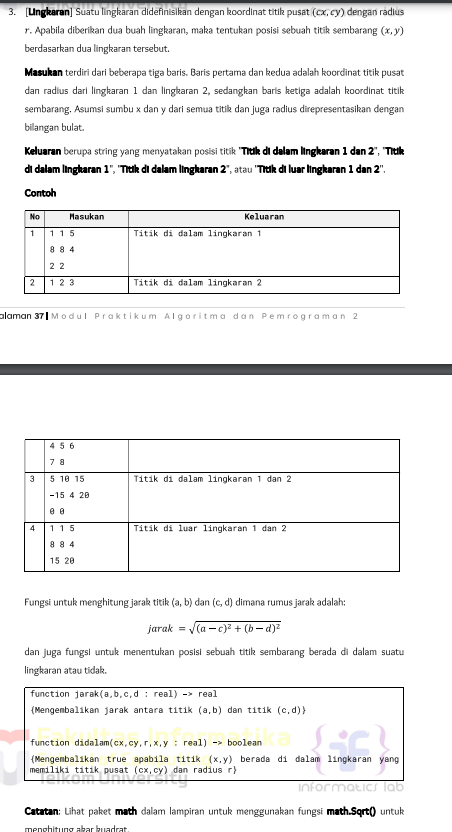
**Screenshoot Output**

****

**Deskripsi Program**

Program di atas menerima tiga bilangan bulat **a**, **b**, dan **c** dari pengguna, lalu menghitung hasil komposisi fungsi **f(x)**, **g(x)**, dan **h(x)** dengan tiga kombinasi berbeda. Fungsi **f(x)** mengembalikan nilai kuadrat dari **x**, fungsi **g(x)** mengurangi **x** dengan 2, dan fungsi **h(x)** menambahkan 1 pada **x**. Komposisi fungsi yang dihitung adalah **(fogoh)(a)** yang merupakan hasil dari f**(g(h(a)))**, **(gohof)(b)** sebagai hasil dari **g(h(f(b)))**, dan **(hofog)(c)** yang merupakan hasil dari **h(f(g(c)))**.Setelah perhitungan, program mencetak hasil dari masing-masing komposisi fungsi tersebut untuk nilai input **a**, **b**, dan **c**.

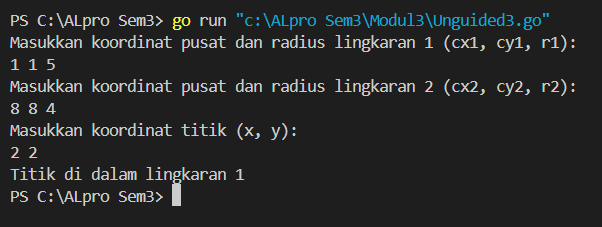
* + - 1. **Soal Studi Case**



**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func jarak(x1, y1, x2, y2 float64) float64 {      return math.Sqrt((x1-x2)\*(x1-x2) + (y1-y2)\*(y1-y2))  }  func didalam(cx, cy, r, x, y float64) bool {      return jarak(cx, cy, x, y) <= r  }  func main() {      var cx1, cy1, r1 float64      fmt.Println("Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1):")      fmt.Scan(&cx1, &cy1, &r1)      var cx2, cy2, r2 float64      fmt.Println("Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2):")      fmt.Scan(&cx2, &cy2, &r2)      var x, y float64      fmt.Println("Masukkan koordinat titik (x, y):")      fmt.Scan(&x, &y)      inCircle1 := didalam(cx1, cy1, r1, x, y)      inCircle2 := didalam(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")      }  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program di atas digunakan untuk menentukan apakah sebuah titik dengan koordinat **(x, y)** berada di dalam salah satu, kedua, atau tidak ada lingkaran sama sekali. Program pertama-tama meminta input koordinat pusat dan jari-jari untuk dua lingkaran (lingkaran 1 dan lingkaran 2) serta koordinat titik yang akan diuji. Fungsi **jarak** menghitung jarak antara dua titik, dan fungsi **di dalam** menentukan apakah **jarak** antara pusat lingkaran dan titik yang diuji lebih kecil atau sama dengan jari-jari lingkaran. Berdasarkan hasil dari kedua fungsi ini, program kemudian mengevaluasi apakah titik berada di dalam lingkaran pertama, lingkaran