**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III**

**Fungsi**

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

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Fariz Ilham / 2311102275**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. **DASAR TEORI**

* Definisi Fungsi

Fungsi (function) adalah suatu rangkaian instruksi yang dirancang untuk memproses input dan mengembalikan nilai tertentu sebagai output. Pada dasarnya, fungsi berperan dalam memetakan satu atau lebih nilai input ke suatu hasil atau keluaran yang spesifik, sesuai dengan instruksi yang diberikan. Dalam konteks pemrograman, fungsi sangat penting untuk modularitas dan pengorganisasian kode.

* Syarat Sebuah Fungsi

Sebuah subprogram dikatakan sebagai fungsi jika memenuhi dua syarat utama:

1. Deklarasi Tipe Nilai yang Dikembalikan: Fungsi harus menentukan tipe data dari nilai yang akan dikembalikan (misalnya, integer, float, string, dsb.).

2. Kata Kunci `return`: Fungsi harus memiliki perintah `return` di dalam tubuhnya untuk mengembalikan nilai hasil dari proses tersebut.

* Kegunaan Fungsi

Fungsi digunakan dalam banyak situasi, terutama ketika sebuah nilai diperlukan sebagai bagian dari proses lebih lanjut, seperti:

- Penugasan nilai ke sebuah variabel.

- Bagian dari ekspresi aritmatika atau logika.

- Argumen yang akan diteruskan ke subprogram lain.

* Penamaan Fungsi

Nama fungsi sebaiknya dipilih sedemikian rupa sehingga secara jelas menggambarkan nilai yang dikembalikan atau proses yang dilakukan oleh fungsi tersebut. Biasanya, nama fungsi menggunakan kata benda atau kata sifat untuk mencerminkan hasil atau proses yang terkait. Contoh penamaan fungsi yang baik adalah:

- `median` (mengembalikan nilai median),

- `rerata` (mengembalikan nilai rata-rata),

- `nilaiTerbesar` (mengembalikan nilai terbesar),

- `ketemu` (mengembalikan hasil pencarian),

- `selesai` (mengindikasikan suatu proses telah selesai).

1. **GUIDED**

**Soal Studi Case**

1. Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilal falitorial dan permutasi

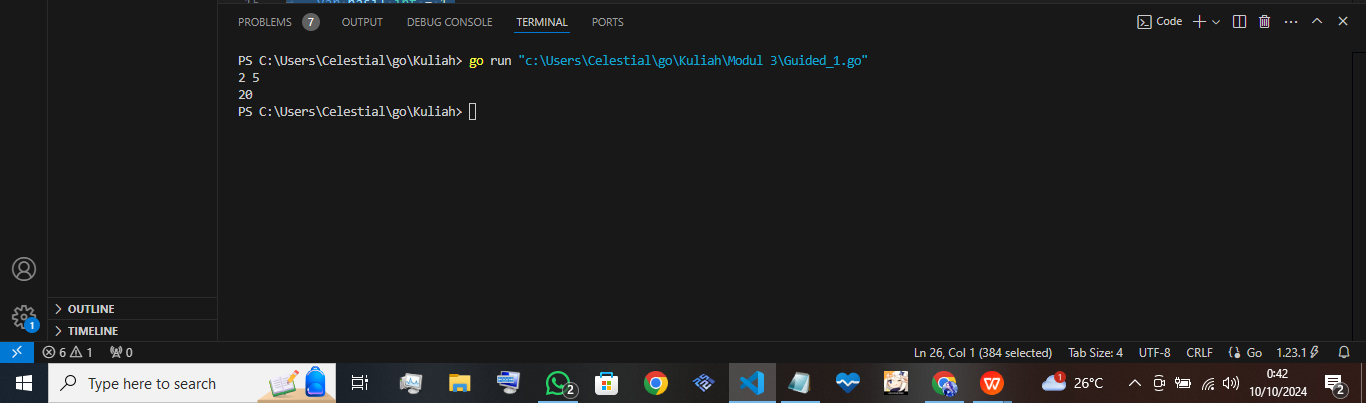
**Masukan** terdiri dari dua buah bilangan positif a dan b

**Keluaran** berupa sebuah bilangan bulat yang menyatakan nilal a permutasi b apabila a >= b atau b pemutasi a untuk kemungkinan yang lain.

**Sourcecode**

|  |
| --- |
| 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 ditulis menggunakan bahasa pemrograman Go untuk menghitung “permutasi” dari dua angka yang dimasukkan oleh pengguna. Program ini mengambil dua angka dari pengguna, kemudian menghitung hasil permutasinya dengan rumus “P(n, r) = n! / (n-r)!” Untuk mempermudah perhitungan, program memastikan bahwa angka yang lebih besar digunakan sebagai “n” (bilangan total) dan angka yang lebih kecil sebagai “r” (bilangan yang dipilih).

* Algoritma Program

1. Inisialisasi dan Input Pengguna:

Program menginisialisasi dua variabel integer, yaitu `a` dan `b`, yang akan diisi dengan dua angka dari pengguna.

2. Membandingkan Angka:

Setelah pengguna memasukkan angka, program membandingkan nilai “a” dan “b”. Jika “a >= b”, maka program langsung menggunakan nilai tersebut dalam perhitungan permutasi. Jika “b > a”, program akan menukar nilai “a” dan “b” sehingga nilai yang lebih besar akan selalu digunakan sebagai “n”.

3. Menghitung Faktorial:

Program menggunakan fungsi `faktorial(n)` untuk menghitung faktorial, yaitu hasil perkalian berturut-turut dari 1 hingga \( n \). Fungsi ini digunakan dalam perhitungan permutasi.

4. Menghitung Permutasi:

Fungsi `permutasi(n, r)` kemudian menghitung permutasi dengan menggunakan rumus: “P(n,r) = n! / (n-r)!” Rumus ini menghitung berapa banyak cara mengatur “r” objek dari “n” objek total.

5. Menampilkan Hasil:

Setelah hasil permutasi dihitung, program akan menampilkan hasil

* Cara Kerja Program

1. Pengguna memasukkan input:

Program meminta pengguna memasukkan dua angka bulat “a” dan “b” .

2. Membandingkan dan menukar nilai:

Program membandingkan dua angka yang dimasukkan. Jika “a” lebih besar atau sama dengan “b”, maka nilai tersebut akan langsung digunakan. Jika tidak, nilai akan ditukar sehingga angka terbesar selalu digunakan sebagai “n”, dan angka terkecil digunakan sebagai “r”.

3. Perhitungan Faktorial:

Program kemudian menghitung faktorial dari “n” menggunakan fungsi `faktorial(n)`. Faktorial adalah hasil kali dari semua bilangan bulat positif dari 1 hingga “n”.

4. Menghitung Permutasi:

* Program menghitung permutasi dengan menggunakan hasil dari faktorial
* Sebagai contoh, jika pengguna memasukkan angka “a = 2” dan “b = 5”, program akan menghitung sesuai dengan formula “P(n, r) = n! / (n-r)!”. “5! = 120” dan ( (5 - 2)! = 3! = 6 ), sehingga hasil akhirnya adalah “120/6 = 20”.

5. Menampilkan hasil:

Hasil perhitungan permutasi akan ditampilkan

* Output Program

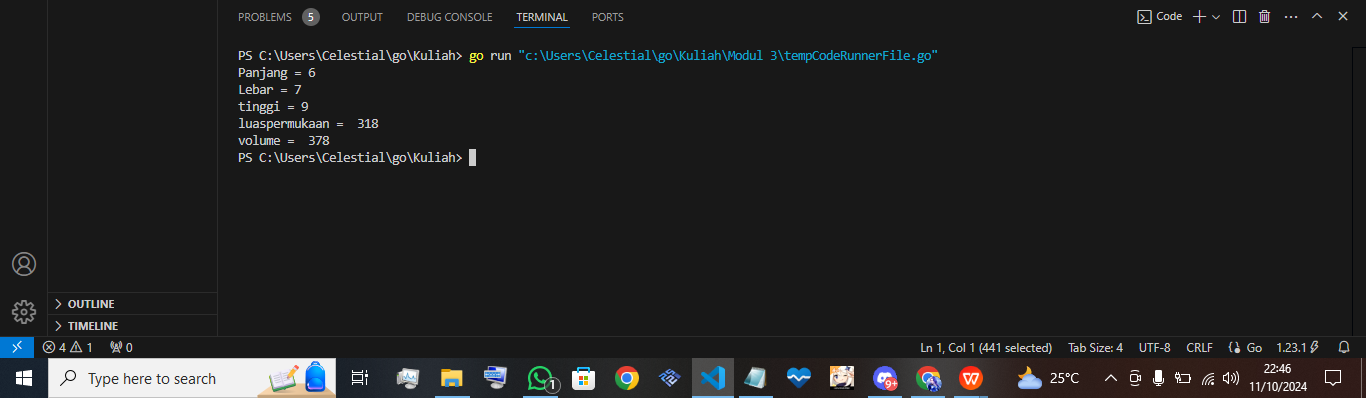
Setelah pengguna memasukkan angka, program akan menampilkan hasil perhitungan permutasi. Misalnya, jika pengguna memasukkan angka “a = 2” dan “b = 5” program akan menampilkan output: 20.

1. Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilal volume dan luas permukaan balok

**Sourcecode**

|  |
| --- |
| package main  import (  "fmt"  )  func main() {  var panjang, lebar, tinggi float64  fmt.Print("Panjang = ")  fmt.Scanln(&panjang)  fmt.Print("Lebar = ")  fmt.Scanln(&lebar)  fmt.Print("tinggi = ")  fmt.Scanln(&tinggi)  luaspermukaan := 2 \* (panjang\*lebar + panjang\*tinggi + lebar\*tinggi)  volume := panjang \* lebar \* tinggi  fmt.Println("luaspermukaan = ", luaspermukaan)  fmt.Println("volume = ", volume)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini menghitung luas permukaan dan volume balok dengan cara menerima input dari pengguna untuk nilai panjang, lebar, dan tinggi. Setelah nilai-nilai tersebut dimasukkan, program akan melakukan perhitungan menggunakan rumus matematika untuk luas permukaan dan volume balok, lalu menampilkan hasilnya.

* Algoritma

1. Inisialisasi Variabel:

Tiga variabel `panjang`, `lebar`, dan `tinggi` bertipe `float64` digunakan untuk menyimpan nilai input dari pengguna.

2. Pengambilan Input:

Program meminta pengguna untuk memasukkan nilai panjang, lebar, dan tinggi balok.

3. Perhitungan:

* Program menghitung “luas permukaan dan volume balok”
* Rumus luaspermukaan = 2 \* (panjang\*lebar + panjang\*tinggi + lebar\*tinggi)
* Rumus volume = panjang \* lebar \* tinggi

4. Menampilkan Hasil:

Program akan menampilkan hasil perhitungan luas permukaan dan volume setelah dihitung.

* Cara Kerja Program

1. Input Pengguna:

Program meminta pengguna untuk memasukkan nilai panjang, lebar, dan tinggi balok.

2. Perhitungan:

* Program menghitung luas permukaan balok dengan rumus luas permukaan balok
* Program juga menghitung volume balok dengan rumus volume balok

3. Output Hasil:

Setelah perhitungan selesai, hasil akan ditampilkan.

* Output Program
* Input:

Panjang = 6, Lebar = 7, Tinggi = 9

* Output:

luaspermukaan = 318, volume = 378

1. Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakan kallan membantu Jonas? (tidak tentunya ya :p)

**Masukan** terdiri dari empat buah bilangan asli a, b, c, dan d yang dipisahkan oleh spasi, dengan syarat a >= c dan b >= d

**Keluaran** terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi a terhadap e, sedangkan baris kedua adalah hasil permutasi dan kombinasi b terhadap d.

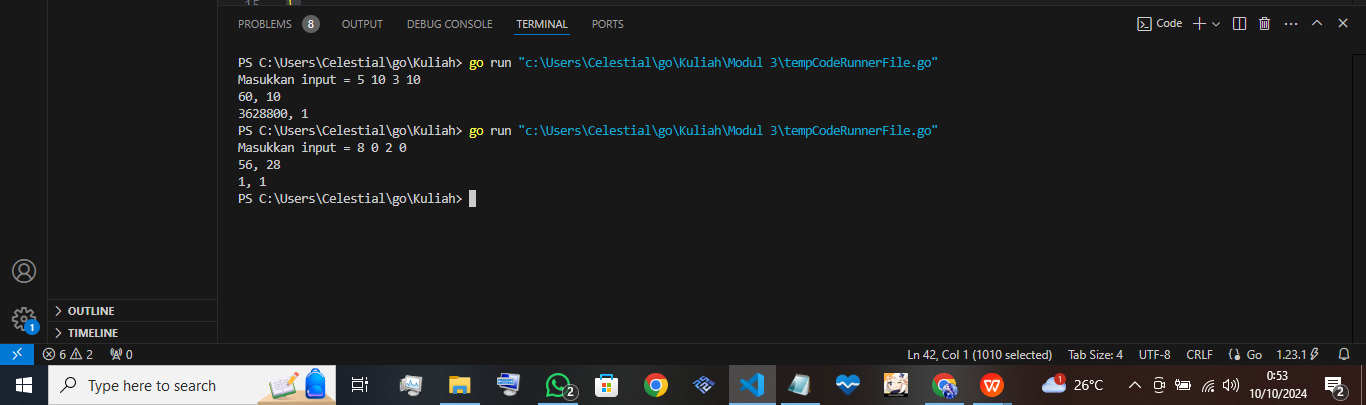
Catatan: permutasi (P) dan kombinasi (C) dari in terhadap r (n >= r) dapat dihitung dengan menggunakan persamaan berikut!

P(n,r)= n!/(n-r)!sedangkan C(n,r)= n!/ r!(n-r)!

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  var a, b, c, d int  // Fungsi untuk menghitung faktorial dari bilangan n  func faktorial(n int) int {  hasil := 1  // Loop untuk menghitung faktorial, dimulai dari 1 hingga n  for i := 1; i <= n; i++ {  hasil = hasil \* i  }  return hasil  }  // Fungsi untuk menghitung permutasi P(n, r) = n! / (n-r)!  func permutasi(n, r int) int {  return faktorial(n) / faktorial(n-r)  }  // Fungsi untuk menghitung kombinasi C(n, r) = n! / (r! \* (n-r)!)  func kombinasi(n, r int) int {  return faktorial(n) / (faktorial(r) \* faktorial(n-r))  }  func main() {  fmt.Print("Masukkan input = ")  fmt.Scan(&a, &b, &c, &d)  if a >= c && b >= d {  // Baris pertama: Permutasi dan Kombinasi a terhadap c  fmt.Printf("%d, %d\n", permutasi(a, c), kombinasi(a, c))  // Baris kedua: Permutasi dan Kombinasi b terhadap d  fmt.Printf("%d, %d\n", permutasi(b, d), kombinasi(b, d))  } else {  fmt.Println("Syarat tidak terpenuhi: a harus >= c dan b harus >= d")  }  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini mengambil empat input angka dari pengguna: “a”, “b”, “c”, dan “d”. Program kemudian memeriksa apakah “a >= c” dan “b >= d”. Jika syarat ini terpenuhi, program akan menghitung dan menampilkan hasil dari :

* Permutasi dan kombinasi “a” terhadap “c”
* Permutasi dan kombinasi “a” terhadap “c”

Jika syarat tidak terpenuhi, program akan menampilkan pesan ("Syarat tidak terpenuhi: a harus >= c dan b harus >= d")

* Algoritma Program

1. Inisialisasi Variabel:

Empat variabel `a`, `b`, `c`, dan `d` bertipe `int` diinisialisasi untuk menyimpan input dari pengguna.

2. Pengambilan Input:

Program meminta pengguna untuk memasukkan nilai untuk keempat variabel.

3. Validasi Kondisi:

* Program memeriksa apakah “a >= c” dan “b >= d”. Jika syarat ini terpenuhi, perhitungan permutasi dan kombinasi dilakukan.
* Jika syarat tidak terpenuhi, program akan menampilkan pesan kesalahan.

4. Perhitungan Permutasi dan Kombinasi:

* Fungsi `permutasi(n, r)` digunakan untuk menghitung permutasi “P(n, r) = n! / (n-r)!”.
* Fungsi `kombinasi(n, r)` digunakan untuk menghitung kombinasi “C(n, r) = n! / r! (n-r)!”.

5. Menampilkan Hasil:

Program menampilkan hasil permutasi dan kombinasi untuk “a” terhadap “c”, dan “b” terhadap “d”.

* Cara Kerja Program

1. Input Pengguna:

Pengguna diminta untuk memasukkan empat angka: “a”, “b”, “c”, dan “d”.

2. Validasi Input:

Program memeriksa nilai “a >= c” dan “b >= d”. Jika kondisi ini terpenuhi, program melanjutkan ke tahap perhitungan. Jika tidak, program mengeluarkan pesan kesalahan.

3. Perhitungan Permutasi dan Kombinasi:

* Jika kondisi terpenuhi, program menggunakan fungsi `permutasi` dan `kombinasi` untuk menghitung:
* Permutasi(a, c),dan kombinasi(a, c))
* Permutasi(b, d), kombinasi(b, d))

4. Output Hasil:

* Program menampilkan hasil perhitungan dalam dua baris:
* Baris pertama menampilkan permutasi dan kombinasi dari “a” terhadap “c”.
* Baris kedua menampilkan permutasi dan kombinasi dari “b” terhadap “d”.
* Output Program
* Input:

Masukkan input = 8 0 2 0

* Output:

56, 28

1, 1

* Penjelasan:
* ( P(8, 0) = 56) dan ( C(8, 0) = 28 )
* ( P(2, 0) = 1 ) dan ( C(2, 0) = 1 )

Jika syarat tidak terpenuhi, program akan menampilkan pesan ("Syarat tidak terpenuhi: a harus >= c dan b harus >= d")

1. **UNGUIDED**

**Soal Studi Case**

1. Diberikan tiga buah fungsi matematika yaitu f (x) = x², g(x) = x-2 dan h (x) = x + 1. Fungsi komposisi (fogoh)(x) artinya adalah f (g(h(x))). Tuliskan f(x), g(x) dan h(x) dalam bentuk function.

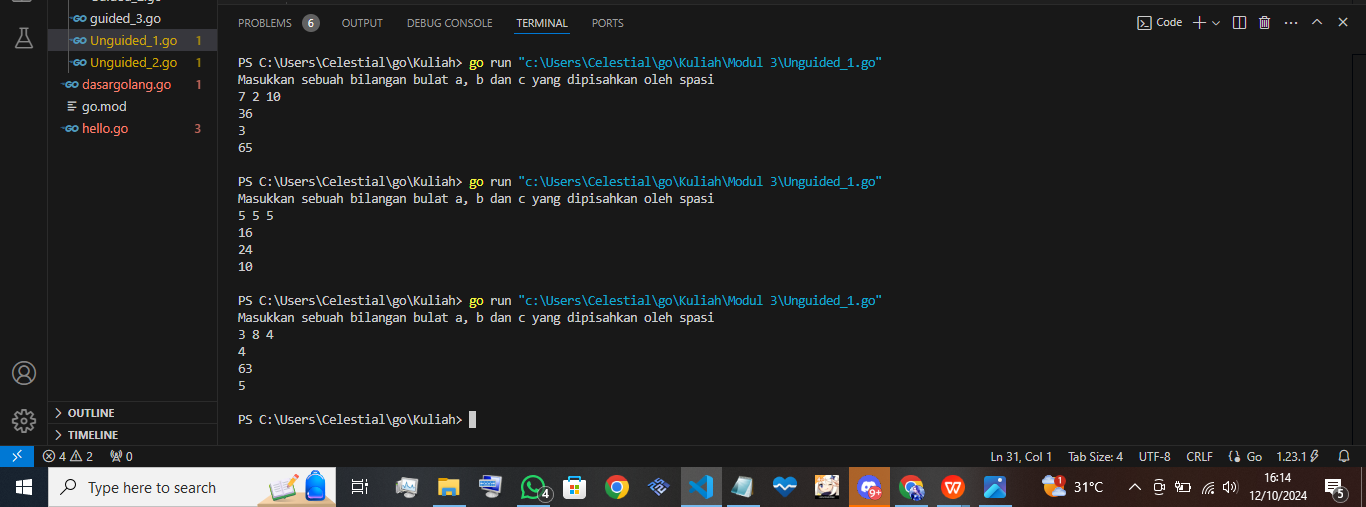
**Masukan** terdiri dari sebuah bilangan bulat a, b dan c yang dipisahkan oleh spasi.

**Keluaran** terdiri dari tiga baris. Baris pertama adalah (fogoh)(a), baris kedua (gohof) (b), dan baris ketiga adalah (hofog)(c)!

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func f(x int) int {  return x \* x  }  func g(x int) int {  return x - 2  }  func h(x int) int {  return x + 1  }  func fogoh(x int) int {  return f(g(h(x)))  }  func gohof(x int) int {  return g(h(f(x)))  }  func hofog(x int) int {  return h(f(g(x)))  }  func main() {  var a, b, c int  fmt.Println("Masukkan sebuah bilangan bulat a, b dan c yang dipisahkan oleh spasi")  fmt.Scanln(&a, &b, &c)  fmt.Println(fogoh(a))  fmt.Println(gohof(b))  fmt.Println(hofog(c))  fmt.Println()  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini bertujuan untuk menghitung komposisi dari tiga fungsi, yaitu“(fogoh(x))”, “(gohof(x))”, dan (hofog(x)) menggunakan tiga variabel “a”, “b”, dan “c”. Dengan rumus “f = x \* x”, “g = x - 2”, “ h = x + 1”.

* Algoritma Program

1. Inisialisasi Fungsi:

Fungsi \( f(x) \), \( g(x) \), dan \( h(x) \) didefinisikan sebagai berikut:

* “f = x \* x”
* “g = x - 2”,
* “ h = x + 1”.

2. Komposisi Fungsi:

Tiga komposisi fungsi didefinisikan:

* `fogoh(x)` menghitung “f(g(h(x)))”
* `gohof(x)` menghitung “g(h(f(x)))”
* `hofog(x)` menghitung “h(f(g(x)))”

3. Pengambilan Input:

Program meminta pengguna untuk memasukkan tiga bilangan bulat “a”, “b”, dan “c”, dipisahkan oleh spasi.

4. Perhitungan dan Output:

Program kemudian menghitung hasil dari masing-masing komposisi fungsi untuk setiap nilai “a”, “b”, dan “c” dengan memanggil fungsi-fungsi komposisi yang telah didefinisikan dan hasilnya akan ditampilkan satu per satu.

* Cara Kerja Program

1. Input Pengguna:

Pengguna diminta untuk memasukkan tiga bilangan bulat “a”, “b”, dan “c”.

2. Perhitungan Komposisi Fungsi:

Program menghitung:

* “f(g(h(a)))” dengan memanggil fungsi `fogoh(a)`
* “g(h(f(b)))” dengan memanggil fungsi `gohof(b)`
* “h(f(g(c)))” dengan memanggil fungsi `hofog(c)`

3. Output Hasil:

Program menampilkan hasil dari perhitungan ketiga komposisi fungsi tersebut.

* Output Program
* Input:

Masukkan sebuah bilangan bulat a, b dan c yang dipisahkan oleh spasi

5 5 5

* Output:

16

24

10

* Penjelasan:
* fogoh(5):
* h(5) = 5 + 1 = 6
* g(5) = 5 - 2 = 3
* f(5) = 5 \* 5 = 25
* gohof(5):
* f(5) = 5 \* 5 = 25
* h(5) = 5 + 1 = 6
* g(5) = 5 - 2 = 3
* hofog(5):
* g(5) = 5 - 2 = 3
* f(5) = 5 \* 5 = 25
* h(5) = 5 + 1 = 6

1. [Lingkaran] 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.

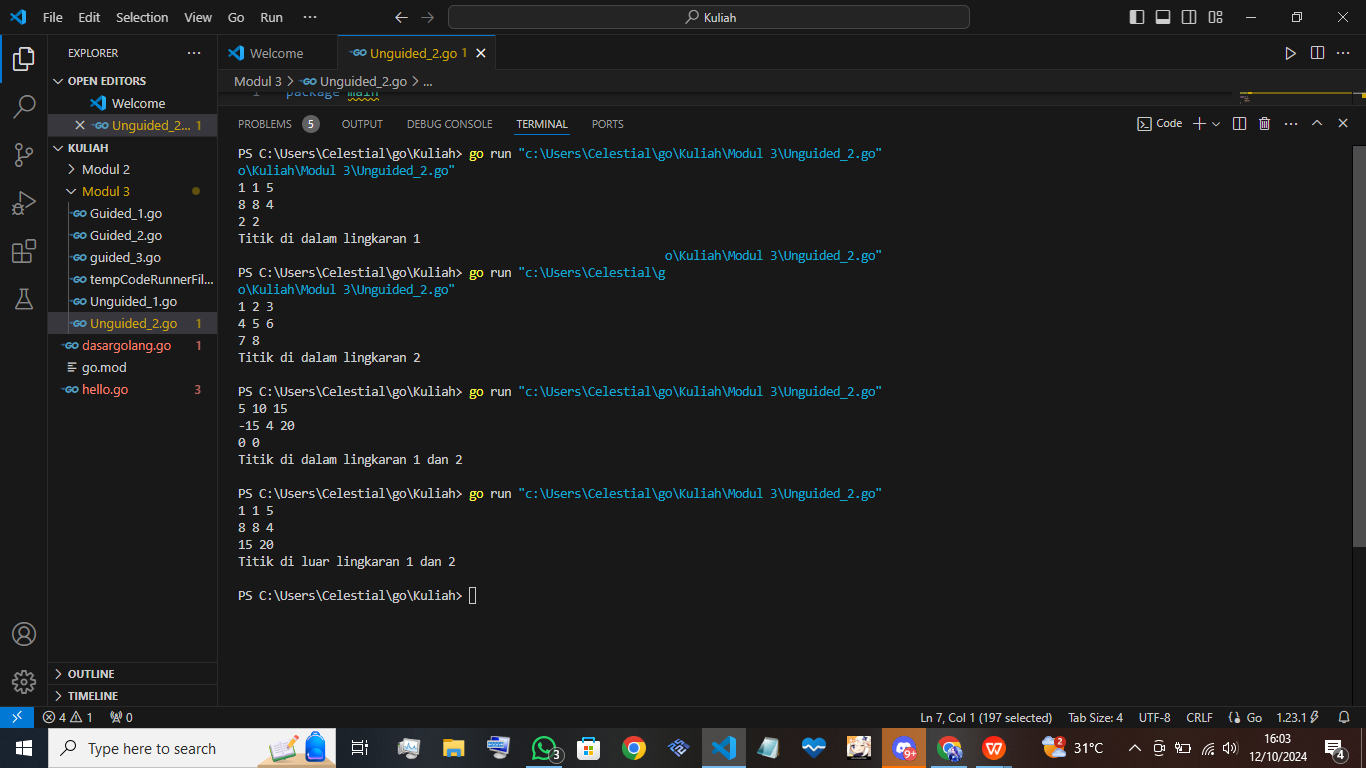
**Masukan** terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat.

**Keluaran** berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 2", "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2".

**Sourcecode**

|  |
| --- |
| package main  import (  "fmt"  "math"  )  func jarak(x1, y1, x2, y2 int) float64 {  return math.Sqrt(float64((x2-x1)\*(x2-x1) + (y2-y1)\*(y2-y1)))  }  func lingkaran(cx, cy, r, x, y int) bool {  return jarak(cx, cy, x, y) <= float64(r)  }  func main() {  var cx1, cy1, r1 int  fmt.Scanln(&cx1, &cy1, &r1)  var cx2, cy2, r2 int  fmt.Scanln(&cx2, &cy2, &r2)  var x, y int  fmt.Scan(&x, &y)  lingkaran1 := lingkaran(cx1, cy1, r1, x, y)  lingkaran2 := lingkaran(cx2, cy2, r2, x, y)  if lingkaran1 && lingkaran2 {  fmt.Println("Titik di dalam lingkaran 1 dan 2")  } else if lingkaran1 {  fmt.Println("Titik di dalam lingkaran 1")  } else if lingkaran2 {  fmt.Println("Titik di dalam lingkaran 2")  } else {  fmt.Println("Titik di luar lingkaran 1 dan 2")  }  fmt.Println()  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini bertujuan untuk mengecek apakah sebuah titik, berada di dalam atau di luar lingkaran satu atau dua. Program menerima koordinat pusat dua lingkaran, jari-jari masing-masing lingkaran, serta koordinat sebuah titik. Program kemudian mengecek apakah titik tersebut berada di dalam salah satu atau kedua lingkaran berdasarkan jarak titik ke pusat lingkaran.

* Algoritma

1. Inisialisasi Variabel:

* Dua lingkaran direpresentasikan oleh koordinat pusat lingkaran dan jari-jari: `(cx1, cy1, r1)` untuk lingkaran pertama dan `(cx2, cy2, r2)` untuk lingkaran kedua.
* Sebuah titik direpresentasikan oleh koordinat `(x, y)`.

2. Fungsi jarak(x1, y1, x2, y2):

* Fungsi ini menghitung jarak antara dua titik dengan rumus:

((x2-x1)\*(x2-x1) + (y2-y1)\*(y2-y1)))

* Jarak ini kemudian dikembalikan dalam bentuk float64.

3. Fungsi lingkaran(cx, cy, r, x, y):

Fungsi ini mengecek apakah jarak titik `(x, y)` ke pusat lingkaran `(cx, cy)` lebih kecil atau sama dengan jari-jari lingkaran `r`. Jika ya, berarti titik berada di dalam lingkaran, dan fungsi mengembalikan nilai `true`; jika tidak, fungsi mengembalikan nilai `false`.

4. Pengambilan Input:

Program meminta pengguna memasukkan koordinat pusat dan jari-jari dari dua lingkaran serta koordinat sebuah titik.

5. Pengecekan Posisi Titik:

Program menggunakan fungsi `lingkaran` untuk mengecek apakah titik berada di dalam lingkaran pertama, lingkaran kedua, atau keduanya.

6. Menampilkan Output:

* Berdasarkan hasil pengecekan, program akan menampilkan salah satu dari empat kemungkinan:
* Titik berada di dalam lingkaran pertama dan kedua.
* Titik hanya berada di dalam lingkaran pertama.
* Titik hanya berada di dalam lingkaran kedua.
* Titik berada di luar kedua lingkaran.
* Cara Kerja Program

1. Input Pengguna:

* Program meminta pengguna untuk memasukkan tiga nilai: koordinat pusat dan jari-jari lingkaran pertama `(cx1, cy1, r1)`.
* Kemudian, pengguna diminta untuk memasukkan tiga nilai berikutnya untuk lingkaran kedua `(cx2, cy2, r2)`.
* Terakhir, pengguna memasukkan koordinat sebuah titik `(x, y)`.

2. Perhitungan:

Program menghitung jarak titik dari pusat lingkaran pertama dan kedua, lalu mengecek apakah titik tersebut berada di dalam salah satu atau kedua lingkaran.

3. Output Hasil:

- Program menampilkan hasil pengecekan apakah titik tersebut berada di dalam satu lingkaran, dua lingkaran, atau di luar kedua lingkaran.

* Output Program
* Input:
* Masukkan cx1, cy1, r1: 1 2 3

Lingkaran pertama memiliki pusat di titik (1, 2) dan jari-jari sebesar 3.

* Masukkan cx2, cy2, r2: 4 5 6

Lingkaran kedua memiliki pusat di titik (4, 5) dan jari-jari sebesar 6.

* Masukkan x, y: 7 8

Titik yang akan dicek posisinya adalah (7, 8).

* Output:

Titik di dalam lingkaran 2