

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

MODUL III

FUNGSI



Oleh:

NAMA : AHMAD TITANA NANDA PRAMUDYA

NIM : 2311102042

KELAS : IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Definisi Function

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

- Ada deklarasi tipe nilai yang dikembalikan, dan
- Terdapat kata kunci return dalam badan subprogram.
- Maka fungsi digunakan jika suatu nilai biasanya diperlukan, seperti:
- Assignment nilai ke suatu variable
- 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, nilaiTerbesar, ketemu, selesai

Deklarasi Function

Deklarasi fungsi sama dengan prosedur, yaitu berada pada blok yang terpisah dengan program utama. Pada bagian deklarasi terlihat setelah parameter terdapat tipe data dari nilai yang dikembalikan, sedangkan pada bagian badan fungsi terdapat return dari nilai yang dikembalikan. Berikut adalah contoh fungsi untuk menghitung volume dari tabung apabila jari-jari alas dan tinggi tabung diketahui.

Pada bagian deklarasi terlihat setelah parameter terdapat tipe data dari nilai yang dikembalikan, sedangkan pada bagian badan fungsi terdapat return dari nilai yang dikembalikan. Berikut adalah contoh fungsi untuk menghitung volume dari tabung apabila jari-jari alas dan tinggi tabung diketahui.

Cara Pemanggilan Function

Sama halnya 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.

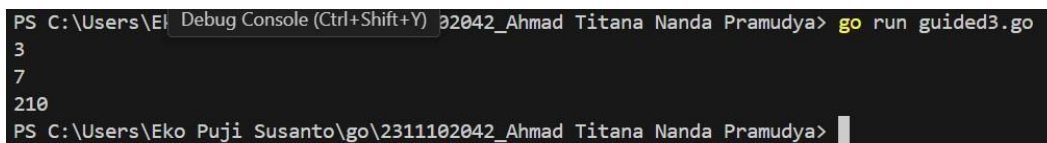
II. GUIDED

No 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 :



```
PS C:\Users\Eko Puji Susanto\go\2311102042_Ahmad Titana Nanda Pramudya> go run guided3.go
3
7
210
PS C:\Users\Eko Puji Susanto\go\2311102042_Ahmad Titana Nanda Pramudya>
```

Penjelasan :

Program ini meminta dua angka dari pengguna (a dan b) dan menggunakannya untuk menghitung permutasi. Permutasi dihitung dengan menggunakan faktorial. Dan Program memastikan bahwa nilai yang lebih besar dari kedua angka dijadikan sebagai n dalam perhitungan permutasi, karena syarat $n \geq r$ harus dipenuhi dalam permutasi.

NO.2

Sourcode :

```
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.")
    }
}
```

Output :

```
PS C:\Users\Eko Puji Susanto\go\2311102042_Ahmad Titana Nanda Pramudya\GUIDED 3> go run guided3.2.go
Masukkan bilangan a, b, c, d (dengan spasi):
5 4 2 3
Permutasi(a, c) dan Kombinasi(a, c): 20 10
Permutasi(b, d) dan Kombinasi(b, d): 24 4
PS C:\Users\Eko Puji Susanto\go\2311102042_Ahmad Titana Nanda Pramudya\GUIDED 3> go run guided3.2.go
Masukkan bilangan a, b, c, d (dengan spasi):
1 2 3
Syarat a >= c dan b >= d tidak terpenuhi.
PS C:\Users\Eko Puji Susanto\go\2311102042_Ahmad Titana Nanda Pramudya\GUIDED 3>
```

Penjelasan :

Program ini untuk menghitung nilai permutasi dan kombinasi dari dua set bilangan yang diberikan oleh pengguna, dengan syarat bahwa elemen yang dipilih (c dan d) tidak boleh lebih besar dari total elemen yang tersedia (a dan b). Permutasi memperhitungkan urutan, sedangkan kombinasi tidak. Dan Jika kondisi input tidak memenuhi syarat (misalnya $a < c$ atau $b < d$), program akan menghentikan perhitungan dan menampilkan pesan kesalahan.

III. UNGUIDE

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 :

```
PS C:\Users\Eko Puji Susanto\go\2311102042_Ahmad Titana Nanda Pramudya> go run unguided3.go
Masukkan data lingkaran pertama (cx1, cy1, r1):
1 1 5
Masukkan data lingkaran kedua (cx2, cy2, r2):
8 8 4
Masukkan koordinat titik (x, y):
2 2
Titik di dalam lingkaran 1
PS C:\Users\Eko Puji Susanto\go\2311102042_Ahmad Titana Nanda Pramudya> |
```

Penjelasan :

Program ini dirancang untuk menentukan apakah sebuah titik berada di dalam satu atau dua lingkaran berdasarkan koordinat pusat dan jari-jari dua lingkaran. Berikut adalah poin-poin utama yang bisa disimpulkan:

- **Perhitungan Jarak:** Program menggunakan rumus jarak Euclidean antara dua titik untuk menghitung jarak dari titik yang diuji ke pusat lingkaran.
- **Kondisi Keberadaan Titik**
Program mengecek apakah jarak dari titik ke pusat lingkaran lebih kecil atau sama dengan jari-jari lingkaran. Jika jaraknya kurang atau sama dengan jari-jari, maka titik berada di dalam lingkaran tersebut.
- **Output Berjenjang:** Program mengidentifikasi empat kemungkinan hasil:
 - Titik di dalam lingkaran 1 dan 2: Jika titik berada di dalam kedua lingkaran.
 - Titik di dalam lingkaran 1: Jika titik hanya berada di dalam lingkaran pertama.
 - Titik di dalam lingkaran 2: Jika titik hanya berada di dalam lingkaran kedua.
 - Titik di luar lingkaran 1 dan 2: Jika titik berada di luar kedua lingkaran.
- **Penggunaan Fungsi Matematika:** Fungsi `math.Sqrt()` digunakan untuk menghitung akar kuadrat dari nilai jarak, dan memastikan perhitungan jarak antara titik dan pusat lingkaran sesuai.