

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III**

**FUNGSI**



**Disusun Oleh :**

**Erwin Rivaldo Silaban/2311102248**

**S1-IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi S.Kom, M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Fungsi dalam sebuah program adalah blok kode yang dirancang untuk melakukan tugas tertentu. Fungsi memiliki beberapa kegunaan penting:

- a. Modularitas: Memecah program menjadi bagian-bagian yang lebih kecil dan dapat dikelola.
- b. Penggunaan ulang kode: Kode yang ditulis sekali dapat digunakan berkali-kali di berbagai bagian program.
- c. Abstraksi: Menyembunyikan detail implementasi, memungkinkan penggunaan fungsi tanpa perlu mengetahui cara kerjanya secara rinci.
- d. Organisasi: Membantu mengorganisir kode dengan lebih baik, meningkatkan keterbacaan dan pemeliharaan.
- e. Efisiensi: Mengurangi duplikasi kode, menghemat memori dan waktu eksekusi.
- f. Pemecahan masalah: Memungkinkan pemecahan masalah kompleks menjadi langkah-langkah yang lebih sederhana.
- g. Pengujian: Memudahkan pengujian dan debugging karena setiap fungsi dapat diuji secara terpisah.

## II. GUIDED

### 1. Guided 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))
    }
}

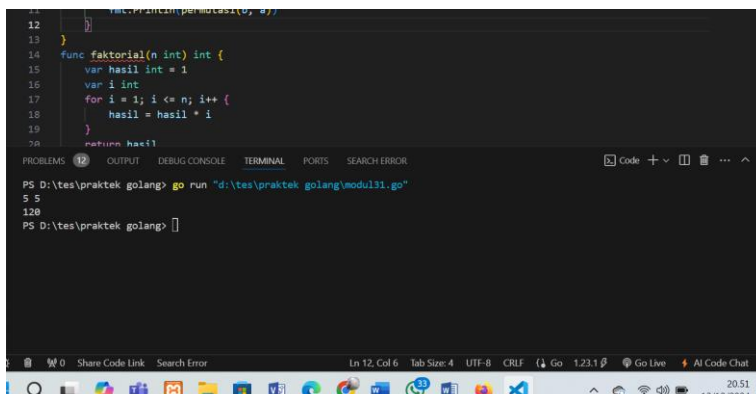
func faktorial(n int) int {
    var hasil int = 1
    var i int
    for i = 1; i <= n; i++ {
        hasil = hasil * i
    }
    return hasil
}

func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}
```

- Deskripsi Program

Program akan menghitung faktorial suatu angka, yang kemudian digunakan dalam fungsi 'permutasi()' untuk menghitung permutasi dari dua angka yang dimasukkan pengguna. Fungsi 'main()' menangani input, memastikan angka yang lebih besar digunakan sebagai 'n' dalam perhitungan permutasi, dan menampilkan hasilnya.

- Screenshot Hasil



## 2. Guided 2

- Souce Code

```
package main

import "fmt"

var p, l, t int

func luasPermukaanBalok(p, l, t int) int {
    luas := 2 * (p*l + l*t + p*t)
    return luas
}

func volumeBalok(p, l, t int) int {
    volume := p * l * t
    return volume
}

func input() {
    fmt.Print("Masukkan panjang balok : ")
    fmt.Scan(&p)
    fmt.Print("Masukkan lebar balok : ")
    fmt.Scan(&l)
    fmt.Print("Masukkan tinggi balok : ")
    fmt.Scan(&t)
}

func main() {

    input()

    lausBalok := luasPermukaanBalok(p, l, t)
    volume := volumeBalok(p, l, t)

    fmt.Printf("Luas permukaan baloknya adalah %v dan volumenya adalah %v", lausBalok,
volume)
}
```

- Deskripsi Program

Program akan melakukan perhitungan, dan menampilkan hasil. Fungsi 'input()' meminta pengguna memasukkan dimensi balok, sementara 'luasPermukaanBalok()' dan 'volumeBalok()' melakukan perhitungan yang diperlukan. Fungsi 'main()' mengkoordinasikan seluruh proses dan menampilkan hasil akhir.

- Screenshot Hasil

The screenshot shows a Go program in a code editor and its execution in a terminal. The code defines a function `volume` that calculates the volume of a rectangular prism given its length (`p`), width (`l`), and height (`t`). It also includes an `input` function that prompts the user for these values. The terminal output shows the program being run, the user entering the values 12, 8, and 7, and the program outputting the surface area (472) and volume (672).

```
13     volume := p * l * t
14     return volume
15 }
16
17 func input() {
18     fmt.Print("Masukkan panjang balok : ")
19     fmt.Scan(&p)
20     fmt.Print("Masukkan lebar balok : ")
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

PS D:\tes\praktek golang> go run "d:\tes\praktek golang\modul32.go"

Masukkan panjang balok : 12

Masukkan lebar balok : 8

Masukkan tinggi balok : 7

Luas permukaan baloknya adalah 472 dan volumenya adalah 672

PS D:\tes\praktek golang>

### III. UNGUIDED

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 bersediakah kalian membantu Jonas? (tidak tentunya ya :p)

Masukan terdiri dari empat buah bilangan asli  $a$ ,  $b$ ,  $c$ , dan  $d$  yang dipisahkan oleh spasi, dengan syarat  $a \geq c$  dan  $b \geq d$

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

Catatan: permutasi (P) dan kombinasi (C) dari  $n$  terhadap  $r$  ( $n \geq r$ ) dapat dihitung dengan menggunakan persamaan berikut!

- Source Code

```
package main

import "fmt"

func main() {
    var nilai1, nilai2, nilai3, nilai4 int
    fmt.Print("Masukkan nilai: ")
    fmt.Scan(&nilai1, &nilai2, &nilai3, &nilai4)
    if nilai1 >= nilai3 && nilai2 >= nilai4 {
        fmt.Println("Permutasi 1:", hitungPermutasi(nilai1, nilai3))
        fmt.Println("Kombinasi 1:", hitungKombinasi(nilai1, nilai3))
        fmt.Println("Permutasi 2:", hitungPermutasi(nilai2, nilai4))
        fmt.Println("Kombinasi 2:", hitungKombinasi(nilai2, nilai4))
    } else {
        fmt.Println("Tidak memenuhi kondisi")
    }
}

func hitungFaktorial(angka int) int {
    hasil_248 := 1
    for i := 1; i <= angka; i++ {
        hasil_248 *= i
    }
    return hasil_248
}

func hitungPermutasi(n, r int) int {
    return hitungFaktorial(n) / hitungFaktorial(n-r)
}

func hitungKombinasi(n, r int) int {
    return hitungFaktorial(n) / (hitungFaktorial(r) * hitungFaktorial(n-r))
}
```

- Deskripsi

Program ini meminta pengguna memasukkan 4 nilai integer. Kemudian, program memeriksa apakah  $\text{nilai1} \geq \text{nilai3}$  dan  $\text{nilai2} \geq \text{nilai4}$ . Jika kondisi ini terpenuhi, program menghitung dan menampilkan permutasi dan kombinasi untuk dua pasang nilai ( $\text{nilai1}$  dengan  $\text{nilai3}$ , dan  $\text{nilai2}$  dengan  $\text{nilai4}$ ). Program menggunakan fungsi terpisah untuk menghitung faktorial, permutasi ( $P(n,r) = n! / (n-r)!$ ), dan kombinasi ( $C(n,r) = n! / (r! * (n-r)!)$ ). Jika kondisi tidak terpenuhi, program hanya menampilkan pesan "Tidak memenuhi kondisi".

- Screenshot Hasil

```

23     }
24     return hasil_248
25 }
26
27 func hitungPermutasi(n, r int) int {
28     return hitungFaktorial(n) / hitungFaktorial(n-r)
29 }
30
31 func hitungKombinasi(n, r int) int {
32     return hitungFaktorial(n) / (hitungFaktorial(r) * hitungFaktorial(n-r))
33 }
34
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\tes\praktek golang> go run "d:\tes\praktek golang\modul311.go"
Masukkan nilai: 12 11 33 44
Tidak memenuhi kondisi
PS D:\tes\praktek golang>

```

2. Diberikan tiga buah fungsi matematika yaitu  $f(x) = x^2$ ,  $g(x) = x - 2$  dan  $h(x) = x + 1$ . Fungsi komposisi  $(f \circ g \circ h)(x)$  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  $(f \circ g \circ h)(a)$ , baris kedua  $(g \circ h \circ f)(b)$ , dan baris ketiga adalah  $(h \circ f \circ g)(c)$ !

- Source code

```

package main

import "fmt"

func main() {
    var win_248, b, c int

    fmt.Scan(&win_248, &b, &c)

    fmt.Println(fogoh(win_248))
}

```

```

        fmt.Println(gohof(b))
        fmt.Println(hofog(c))
    }

    func y(x int) int {
        return x * x
    }

    func z(x int) int {
        return x - 2
    }

    func o(x int) int {
        return x + 1
    }

    func fogoh(x int) int {
        return y(z(o(x)))
    }

    func gohof(x int) int {
        return z(o(y(x)))
    }

    func hofog(x int) int {
        return o(y(z(x)))
    }

```

- Deskripsi**  
 Program ini mendemonstrasikan konsep komposisi fungsi dalam matematika. Terdapat tiga fungsi dasar:  $y(x) = x^2$ ,  $z(x) = x - 2$ , dan  $o(x) = x + 1$ . Program meminta tiga input integer dan menerapkan tiga komposisi fungsi berbeda pada input tersebut:  $\text{fogoh}(x) = y(z(o(x)))$ ,  $\text{gohof}(x) = z(o(y(x)))$ , dan  $\text{hofog}(x) = o(y(z(x)))$ . Hasil dari ketiga komposisi fungsi ini dihitung dan ditampilkan untuk masing-masing input, menunjukkan bagaimana urutan penerapan fungsi mempengaruhi hasil akhir.
- Screenshot Hasil**



```
1  9
3  10      fmt.Println(fogoh(win_248))
3  11      fmt.Println(gohof(b))
1  12      fmt.Println(hofog(c))
1  13  }
1  14
1  15  func y(x int) int {
1  16      return x * x
1  17  }
1  18
1  19  func z(x int) int {
1  20      return x - 2
1  21  }
1  22
1  23  func o(x int) int {

PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\tes\praktek golang> go run "d:\tes\praktek golang\modul33.go"
8 9 10
49
80
65
PS D:\tes\praktek golang> |
```

3. [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".

- Source Code

```
package main

import (
    "fmt"
    "math"
)

type Lingkaran struct {
    koordinatX, koordinatY, jariJari float64
}

func titikDiDalam(lingkaran Lingkaran, x, y float64) bool {
    jarak := math.Sqrt(math.Pow(x-lingkaran.koordinatX, 2) + math.Pow(y-
    lingkaran.koordinatY, 2))
    return jarak <= lingkaran.jariJari
}

func tentukanPosisi(lingkaran1, lingkaran2 Lingkaran, x, y float64) string
{
    diLingkaran1 := titikDiDalam(lingkaran1, x, y)
```

```

diLingkaran2 := titikDiDalam(lingkaran2, x, y)

if diLingkaran1 && diLingkaran2 {
    return "Titik di dalam kedua lingkaran"
} else if diLingkaran1 {
    return "Titik di dalam lingkaran 1"
} else if diLingkaran2 {
    return "Titik di dalam lingkaran 2"
}
return "Titik di luar kedua lingkaran"
}

func main() {
    var lingkaran1, lingkaran2 Lingkaran
    var x, y float64

    fmt.Println("Masukkan koordinat pusat dan jari-jari lingkaran 1 (x y r):")
    fmt.Scan(&lingkaran1.koordinatX, &lingkaran1.koordinatY,
    &lingkaran1.jariJari)

    fmt.Println("Masukkan koordinat pusat dan jari-jari lingkaran 2 (x y r):")
    fmt.Scan(&lingkaran2.koordinatX, &lingkaran2.koordinatY,
    &lingkaran2.jariJari)

    fmt.Println("Masukkan koordinat titik yang akan dicek (x y):")
    fmt.Scan(&x, &y)

    hasil := tentukanPosisi(lingkaran1, lingkaran2, x, y)
    fmt.Println(hasil)
}

```

- **Deskripsi**  
Program ini menentukan posisi sebuah titik relatif terhadap dua lingkaran di bidang dua dimensi. Program mendefinisikan struct Lingkaran yang menyimpan koordinat pusat (x,y) dan jari-jari. Pengguna diminta memasukkan koordinat pusat dan jari-jari untuk dua lingkaran, serta koordinat titik yang akan dicek. Program menggunakan rumus jarak Euclidean untuk menentukan apakah titik berada di dalam lingkaran. Fungsi tentukanPosisi mengecek posisi titik terhadap kedua lingkaran dan mengembalikan hasil berupa string yang menunjukkan apakah titik tersebut berada di dalam kedua lingkaran, hanya di dalam salah satu lingkaran, atau di luar kedua lingkaran.
- **Screenshot Hasil**

```
15 }
16
17 func tentukanPosisi(lingkaran1, lingkaran2 Lingkaran, x, y float64) string {
18     dilingkaran1 := titikDiDalam(lingkaran1, x, y)
19     dilingkaran2 := titikDiDalam(lingkaran2, x, y)
20
21     if dilingkaran1 && dilingkaran2 {
22         return "Titik di dalam kedua lingkaran"
23     } else if dilingkaran1 {
24         return "Titik di dalam lingkaran 1"
25     } else if dilingkaran2 {
26         return "Titik di dalam lingkaran 2"
27     }
28     return "Titik di luar kedua lingkaran"
29 }
30
31 func main() {
32     var lingkaran1, lingkaran2 Lingkaran
33     var x, y float64
34 }
```

PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

PS D:\tes\praktek golang> go run "d:\tes\praktek golang\modul323.go"

Masukkan koordinat pusat dan jari-jari lingkaran 1 (x y r):

1 2 3

Masukkan koordinat pusat dan jari-jari lingkaran 2 (x y r):

3 9 7

Masukkan koordinat titik yang akan dicek (x y):

6 7

Titik di dalam lingkaran 2

PS D:\tes\praktek golang>

Ln 27, Col 6 Tab Size: 4 UTF-8 CRLF Go 1.23.1 Go Live AI Code Chat

22.43 13/10/2024