#### **LAPORAN**

# PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2

### MODUL 3

#### **FUNGSI**



#### Oleh:

NAMA: NUFAIL ALAUDDIN TSAQIF

NIM: 10311240084

KELAS: IF-12-01

# S1 TEKNIK INFORMATIKA TELKOM UNIVERSITY PURWOKERTO

2025

#### I. DASAR TEORI

#### 1. Definisi Function

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

- 1. Ada deklarasi tipe nilai yang dikembalikan, dan
- 2. Terdapat kata kunci return dalam badan subprogram.

Maka fungsi digunakan jika suatu nilai biasanya diperlukan, seperti:

- Assignment nilai ke suatu variabel
- 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, ...

#### 2. 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.

# II. GUIDED 1

```
guided1 > 🕶 guided1.go > ...
      package main
      import "fmt"
       func main() {
          var a, b int
          fmt.Scan(&a, &b)
          if a >= b {
               fmt.Println(permutasi(a, b))
               fmt.Println(permutasi(b, a))
       func faktorial(n int) int {
          hasil := 1
           for i := 1; i <= n; i++ {
               hasil *= i
          return hasil
       func permutasi(n, r int) int {
              return 0
          return faktorial(n) / faktorial(n-r)
 28
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\2 4

12

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3>
```

Program ini menghitung permutasi dua angka yang diberikan oleh pengguna, yaitu a dan b. Jika a lebih besar atau sama dengan b, program akan memanggil fungsi permutasi(a, b); jika tidak, fungsi permutasi(b, a) dipanggil. Fungsi permutasi menggunakan rumus faktorial, yang dihitung melalui fungsi faktorial dengan mengalikan angka dari 1 hingga n untuk menghasilkan hasil yang digunakan dalam rumus permutasi.

#### **GUIDED 2**

#### **SURCE CODE**

```
guided2 > 🕶 guided2.go > ...
       package main
       func celciusToFahrenheit(celcius float64) float64 {
           return (celcius * 9.0 / 5.0) + 32
       func main() {
           var N int
           fmt.Print("Masukan jumlah data: ")
           _, err := fmt.Scan(&N)
if err != nil || N <= 0 [
               fmt.Println("Input tidak valid, masukan angka positif.")
           temperatures := make([]float64, N)
           fmt.Println("Masukan suhu dalam celcius: ")
               if err = fmt.Scan(&temperatures[i])
if err != nil [
                   fmt.Println("Input tidak valid, pastikan masukan angka.")
           fmt.Println("Suhu dalam Fahrenheit: ")
           for _, temp := range temperatures (
               fmt.Printf("%.2f\n", celciusToFahrenheit(temp))
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\
Masukan jumlah data: 3

Masukan suhu dalam celcius:
16 22 26

Suhu dalam Fahrenheit:
60.80

71.60

78.80

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3>
```

Program ini menerima input suhu dalam Celsius dari pengguna dan mengonversinya ke Fahrenheit. Setelah meminta jumlah data suhu, program akan meminta input suhu dan menyimpannya dalam array. Setiap suhu kemudian dikonversi menggunakan rumus (Celsius \* 9.0 / 5.0) + 32 dan hasilnya ditampilkan. Jika input jumlah data tidak valid, program menampilkan pesan kesalahan.

#### **GUIDED 3**

#### **SOURCE CODE**

```
guided3 > \infty guided3.go > ...
       import (
           "fmt"
            "math
       func luasPermukaanTabung(r, t float64) float64 {
   return 2 * math.Pi * r * (r + t)
       // Fungsi untuk menghitung volume tabung
func volumeTabung(r, t float64) float64 {
   return math.Pi * math.Pow(r, 2) * t
       func main() {
           fmt.Print("Masukkan jari-jari tabung: ")
           _, errR := fmt.Scan(&r)
           fmt.Print("Masukkan tinggi tabung: ")
           _, errT := fmt.Scan(&t)
           if errR != nil || errT != nil {
               fmt.Println("Input tidak valid! Harap masukkan angka yang benar.")
           // Memeriksa apakah jari-jari dan tinggi bernilai positif
           if r <= 0 || t <= 0 {
                fmt.Println("Jari-jari dan tinggi tabung harus lebih dari nol.")
           luas := luasPermukaanTabung(r, t)
           volume := volumeTabung(r, t)
           fmt.Println("======="")
           fmt.Printf("Luas Permukaan Tabung: %.2f satuan<sup>2</sup>\n", luas)
           fmt.Printf("Volume Tabung: %.2f satuan3\n", volume)
           fmt.Println("=======")
```

#### **OUTPUT**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIP Masukkan jari-jari tabung: 4

Masukkan tinggi tabung: 10

Luas Permukaan Tabung: 351.86 satuan²

Volume Tabung: 502.65 satuan³

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> |
```

#### **DEKSRISI**

Program ini menghitung luas permukaan dan volume tabung berdasarkan input jarijari dan tinggi tabung dari pengguna. Jika input valid (positif), program menghitung luas permukaan dan volume menggunakan rumus yang sesuai dan menampilkan hasilnya. Jika input tidak valid, program menampilkan pesan kesalahan.

#### III. UNGUIDED

#### **UNGUIDED 1**

#### **SOURCE CODE**

```
package main
// NUFAIL ALAUCOIN TSAQIF
//103112/ACCORD
import "fmt"

func main() {
    var a, b, c, d int
    fmt.Scan(8a, 8b, 8c, 8d)

    pl:-permutasi(a, c)
    cl := kombinasi(a, c)
    fmt.Printf("%d %d\n", pl, cl)

    p2 :=permutasi(b, d)
    c2 := kombinasi(b, d)
    fmt.Printf("%d %d\n", p2, c2)
}

func faktorial(n int) int {
    if n <= 1 (
        return 1
    )
    return n * faktorial(n-1)
}

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

func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}</pre>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROG 5 10 3 10 60 10 3628800 1

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3>
```

Program ini menghitung permutasi dan kombinasi dengan menggunakan fungsi faktorial. Setelah menerima empat input, program memeriksa apakah kondisi a >= b dan c >= d terpenuhi. Jika ya, hasil permutasi dan kombinasi ditampilkan; jika tidak, program menampilkan pesan "Input tidak sesuai".

#### **UNGUIDED 2**

#### **SOURCE CODE**

```
guided2 > 🚥 Uunguided2.go 🗦 ...
    package main
    import "fmt"
    func kuadrat(n int) int {
        return n * n
    func kurang(n int) int {
        return n - 2
    func tambah(n int) int {
    func fogoh(n int) int {
        return kuadrat(kurang(tambah(n)))
    func gohof(n int) int {
        return kurang(tambah(kuadrat(n)))
    func hofog(n int) int {
        return tambah(kuadrat(kurang(n)))
    func main() {
        var a, b, c int
        fmt.Scan(&a, &b, &c)
        fmt.Println(fogoh(a))
        fmt.Println(gohof(b))
        fmt.Println(hofog(c))
```

Program ini menghitung nilai dari tiga fungsi komposisi yang melibatkan operasi kuadrat, pengurangan 2, dan penambahan 1. Tiga fungsi komposisi utama, yaitu fogoh, gohof, dan hofog, mengkombinasikan ketiga operasi ini dalam urutan yang berbeda. Program menerima tiga input (a, b, dan c), lalu menghitung dan menampilkan hasil komposisi masing-masing fungsi.

#### **UNGUIDED 3**

#### **SOURCE CODE**

```
import "fmt"
func jarak(a, b, c, d float64) float64 {
   return ((a - c) * (a - c)) + ((b - d) * (b - d))
func didalam(cx, cy, r, x, y float64) bool (
    return jarak(cx, cy, x, y) \leftarrow (r * r)
func main() (
    var cx1, cy1, r1 float64
    var cx2, cy2, r2 float64
    var x, y float64
    fmt.Scan(&cx1, &cy1, &r1)
    fmt.Scan(&cx2, &cy2, &r2)
    fmt.Scan(&x, &y)
    dalamLingkaran1 := didalam(cx1, cy1, r1, x, y)
    dalamtingkaran2 := didalam(cx2, cy2, r2, x, y)
    if dalamtingkaran1 && dalamtingkaran2 |
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalamLingkaran1 (
fmt.Println("Titik di dalam lingkaran 1")
    ) else if dalamtingkaran2 (
        fmt.Println("Titik di dalam lingkaran 2")
        fmt.Println("Titik di luar lingkaran 1 dan 2")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\183112488884_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\183112488884_MODUL3> []
```

Program ini memeriksa apakah sebuah titik (x, y) berada di dalam dua lingkaran. Fungsi jarak menghitung jarak antara titik dan pusat lingkaran menggunakan rumus Euclidean, sementara fungsi didalam memeriksa apakah titik berada dalam lingkaran dengan membandingkan jarak titik dengan jari-jari lingkaran. Program meminta input untuk pusat, jari-jari kedua lingkaran, dan koordinat titik, kemudian menentukan posisi titik apakah berada di dalam kedua lingkaran, satu lingkaran, atau di luar keduanya.

#### IV. KESIMPULAN

Kesimpulan dari laporan praktikum ini adalah bahwa pemahaman tentang fungsi dalam bahasa Go sangat penting untuk memecahkan berbagai masalah komputasi secara terstruktur dan efisien. Fungsi memungkinkan pemrogram untuk menyusun program dengan cara yang modular, meningkatkan keterbacaan, dan memudahkan pemeliharaan kode. Praktikum ini menunjukkan penerapan fungsi dalam berbagai konteks, seperti menghitung permutasi, konversi suhu, dan perhitungan geometri. Melalui penggunaan fungsi-fungsi dasar seperti faktorial, komposisi fungsi, dan pengujian kondisi, praktikum ini memberikan pemahaman yang lebih mendalam tentang cara mendefinisikan, memanggil, dan mengembalikan nilai dari fungsi dalam pemrograman Go.

## **REFERENSI**

## MODUL 3-FUNGSI-ALGORITMA PEMOGRAMAN 2