

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

MATERI



Oleh:

MUHAMMAD ZAKY MUBAROK

103112400073

KELAS

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

MODUL 3. FUNGSI

II. GUIDED

```
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 {
    hasil := 1
    for i := 1; i <= n; i++ {
        hasil *= i
    }
    return hasil
}

func Permutasi (n, r int) int {
    if r > n {
        return 0
    }
    return faktorial(n) / faktorial(n-r)
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Guide 1\Guide1.go"
5 10
30240
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Guide 1\Guide1.go"
1 2
2
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Guide 1\Guide1.go"
10 5
30240
PS D:\ALPRO 2> 
```

Penjelasan :

Program ini menghitung permutasi dari dua angka yang diberikan pengguna. Program pertama-tama menerima dua input (a dan b), lalu memeriksa angka mana yang lebih besar. Fungsi Permutasi akan dipanggil dengan parameter yang lebih besar terlebih dahulu, untuk memastikan perhitungan permutasi yang benar. Permutasi dihitung menggunakan rumus , di mana faktorial dihitung oleh fungsi factorial

III. GUIDED

```
package main

import (
    "fmt"
)

// Fungsi untuk mengonversi suhu dari Celsius ke Fahrenheit
func celsiusToFahrenheit(celsius float64) float64 {
    return (9.0/5.0)*celsius + 32
}

func main() {
    var N int
    fmt.Print("Masukkan jumlah data: ")
    _, err := fmt.Scan(&N)
    if err != nil || N <= 0 {
        fmt.Println("Input tidak valid, pastikan memasukkan angka positif.")
        return
    }

    temperatures := make([]float64, N)

    // Membaca suhu dalam Celsius
    fmt.Println("Masukkan suhu dalam Celsius:")
    for i := 0; i < N; i++ {
        _, err := fmt.Scan(&temperatures[i])
        if err != nil {
            fmt.Println("Input tidak valid, pastikan memasukkan angka.")
            return
        }
    }

    // Mengonversi ke Fahrenheit dan mencetak hasil
    fmt.Println("Suhu dalam Fahrenheit:")
    for _, temp := range temperatures {
        fmt.Printf("%.2f\n", celsiusToFahrenheit(temp))
    }
}
```

```
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Guide 2\Guide2.go"
Masukkan jumlah data: 3
Masukkan suhu dalam Celsius:
3
2
1
Suhu dalam Fahrenheit:
37.40
35.60
33.80
PS D:\ALPRO 2> |
```

Penjelasan :

Program ini mengonversi suhu dari Celsius ke Fahrenheit. Pengguna diminta untuk memasukkan jumlah data suhu, lalu memasukkan suhu dalam Celsius. Setiap suhu kemudian dikonversi ke Fahrenheit menggunakan rumus , dan hasilnya ditampilkan.

IV. GUIDED

```
package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung luas permukaan tabung
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() {
    var r, t float64

    // Input jari-jari dan tinggi tabung dengan validasi
    fmt.Print("Masukkan jari-jari tabung: ")
    _, errR := fmt.Scan(&r)
    fmt.Print("Masukkan tinggi tabung: ")
    _, errT := fmt.Scan(&t)

    // Memeriksa apakah input valid
    if errR != nil || errT != nil {
        fmt.Println("Input tidak valid! Harap masukkan angka yang benar.")
        return
    }

    // Memeriksa apakah jari-jari dan tinggi bernilai positif
    if r <= 0 || t <= 0 {
        fmt.Println("Jari-jari dan tinggi tabung harus lebih dari nol.")
        return
    }

    // Menghitung luas permukaan dan volume
    luas := luasPermukaanTabung(r, t)
```

```

    volume := volumeTabung(r, t)

    // Menampilkan hasil
    fmt.Println("=====")
    fmt.Printf("Luas Permukaan Tabung: %.2f satuan²\n", luas)
    fmt.Printf("Volume Tabung: %.2f satuan³\n", volume)
    fmt.Println("=====")
}

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Guide 3\Guide3.go"
Masukkan jari-jari tabung: 10
Masukkan tinggi tabung: 15
=====
Luas Permukaan Tabung: 1570.80 satuan²
Volume Tabung: 4712.39 satuan³
=====
PS D:\ALPRO 2> 

```

Penjelasan :

Program ini menghitung luas permukaan dan volume tabung. Pengguna diminta untuk memasukkan jari-jari dan tinggi tabung. Program memvalidasi input untuk memastikan nilai positif. Kemudian, menghitung dan menampilkan:

Luas Permukaan tabung dengan rumus

Volume tabung dengan rumus

V. UNGUIDED

```
//Muhammad Zaky Mubarak
package main

import "fmt"

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

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))
}

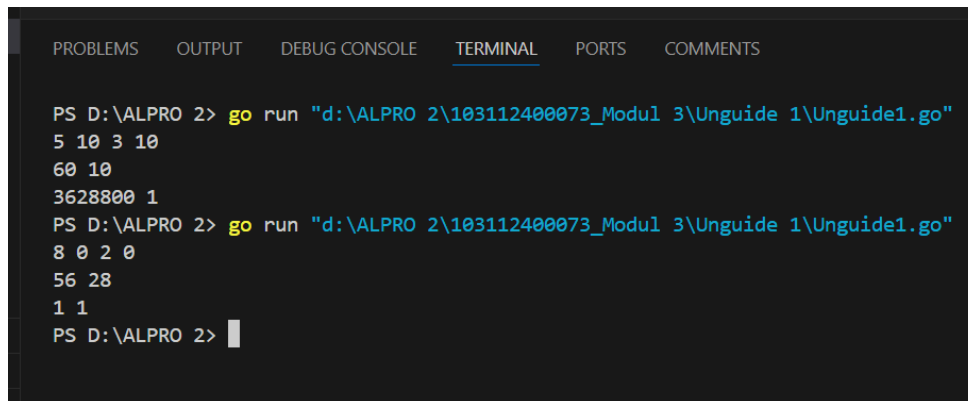
func main() {

    var a, b, c, d int
    fmt.Scan(&a, &b, &c, &d)

    fmt.Println( permutasi(a, c), kombinasi(a, c))

    fmt.Println( permutasi(b, d), kombinasi(b, d))

}
```



```
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 1\Unguide1.go"
5 10 3 10
60 10
3628800 1
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 1\Unguide1.go"
8 0 2 0
56 28
1 1
PS D:\ALPRO 2> 
```

Penjelasan :

1. rogram ini terdiri dari tiga fungsi utama:
 - **Fungsi Faktorial** (faktorial): Menghitung hasil kali semua bilangan dari 1 hingga bilangan tertentu (n).
 - **Fungsi Permutasi** (permutasi): Menghitung jumlah cara penyusunan objek berdasarkan jumlah total objek (n) dan berapa banyak objek yang diambil dalam satu waktu (r).
 - **Fungsi Kombinasi** (kombinasi): Menghitung jumlah cara memilih objek tanpa memperhatikan urutannya.
2. Fungsi main digunakan untuk:
 - Membaca input empat angka dari pengguna: a, b, c, dan d.
 - Menggunakan angka tersebut untuk menghitung:
 - Permutasi dan kombinasi antara a dan c.
 - Permutasi dan kombinasi antara b dan d.
 - Mencetak hasil perhitungan tersebut.
3. Hasil perhitungan diproses menggunakan fungsi permutasi dan kombinasi, yang masing-masing bergantung pada fungsi faktorial untuk menghitung nilai-nilai dasar.

Sebagai contoh, jika angka yang diinput adalah a = 5, b = 4, c = 2, d = 3, program akan menghitung permutasi dan kombinasi untuk pasangan 5 dan 2, serta 4 dan 3, lalu mencetak hasilnya ke layar. Outputnya menunjukkan banyaknya cara penyusunan (permutasi) dan pemilihan (kombinasi) berdasarkan input.

VI. UNGUIDED II

```
//Muhammad Zaky Mubarak
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 main() {
    var a, b, c int
    fmt.Scan(&a, &b, &c)

    f_g_h_a := f(g(h(a)))
    g_h_f_b := g(h(f(b)))
    h_f_g_c := h(f(g(c)))

    fmt.Println( f_g_h_a)
    fmt.Println( g_h_f_b)
    fmt.Println( h_f_g_c)
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 2\Unguide2.go"
7 2 10
36
3
65
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 2\Unguide2.go"
5 5 5
16
24
10
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 2\Unguide2.go"
3 8 4
4
63
5
PS D:\ALPRO 2> 
```

Penjelasan :

Program ini menggunakan tiga fungsi sederhana (f, g, dan h) untuk memanipulasi angka, lalu menggabungkan fungsi-fungsi tersebut secara berurutan pada tiga input bilangan (a, b, c). Berikut penjelasannya:

1. Fungsi f: Menghitung kuadrat dari angka yang diberikan.
2. Fungsi g: Mengurangi angka yang diberikan dengan 2.
3. Fungsi h: Menambahkan 1 pada angka yang diberikan.

Di bagian main:

- Program membaca tiga angka bulat dari pengguna: a, b, dan c.
- Untuk setiap angka, program menerapkan kombinasi fungsi dengan urutan yang berbeda:
 - $f(g(h(a)))$: Memproses a dengan fungsi h dulu, lalu g, dan terakhir f.
 - $g(h(f(b)))$: Memproses b dengan fungsi f dulu, lalu h, dan terakhir g.
 - $h(f(g(c)))$: Memproses c dengan fungsi g dulu, lalu f, dan terakhir h

VII. UNGUIDED III

```
//Muhammad Zaky Mubarak
package main

import (
    "fmt"
    "math"
)

func jarak(a, b, c, d int) float64 {
    return math.Sqrt(float64((a-c)*(a-c) + (b-d)*(b-d)))
}

func didalam(cx, cy, r, x, y int) bool {
    return jarak(x, y, cx, cy) <= float64(r)
}

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, x, y int

    fmt.Scan(&cx1, &cy1, &r1)
    fmt.Scan(&cx2, &cy2, &r2)
    fmt.Scan(&x, &y)

    diLingkaran1 := didalam(cx1, cy1, r1, x, y)
    diLingkaran2 := didalam(cx2, cy2, r2, x, y)

    if diLingkaran1 && diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 3\Unguide3.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 3\Unguide3.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 3\Unguide3.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\ALPRO 2> go run "d:\ALPRO 2\103112400073_Modul 3\Unguide 3\Unguide3.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
PS D:\ALPRO 2>
```

Penjelasan :

1. **Fungsi jarak:** Fungsi ini menghitung jarak Euclidean (jarak lurus) antara dua titik di bidang koordinat, yaitu titik (a, b) dan (c, d) . Jarak ini digunakan untuk menentukan apakah suatu titik berada di dalam lingkaran dengan menghitung jarak antara titik tersebut dan pusat lingkaran.
2. **Fungsi didalam:** Fungsi ini memeriksa apakah suatu titik (x, y) berada di dalam lingkaran dengan pusat (cx, cy) dan jari-jari r . Fungsi ini menggunakan hasil dari fungsi jarak. Jika jarak titik ke pusat lingkaran lebih kecil atau sama dengan jari-jari, maka titik dianggap berada di dalam lingkaran.
3. **Fungsi main:**
 - o Membaca data dari input pengguna, yaitu:
 - Pusat $(cx1, cy1)$ dan jari-jari $r1$ untuk lingkaran pertama.
 - Pusat $(cx2, cy2)$ dan jari-jari $r2$ untuk lingkaran kedua.
 - Koordinat titik (x, y) yang ingin diperiksa.
 - o Memanggil fungsi didalam untuk menentukan apakah titik (x, y) berada di dalam lingkaran pertama dan/atau kedua.
 - o Menentukan dan mencetak hasil:
 - Jika titik berada di kedua lingkaran, program mencetak: **"Titik di dalam lingkaran 1 dan 2"**.

- Jika hanya di lingkaran pertama, mencetak: "**Titik di dalam lingkaran 1**".
- Jika hanya di lingkaran kedua, mencetak: "**Titik di dalam lingkaran 2**".
- Jika tidak di kedua lingkaran, mencetak: "**Titik di luar lingkaran 1 dan 2**".

Contoh Input:

```
0 0 5
10 0 5
3 3
```

Penjelasan:

- Lingkaran pertama: pusat di $(0, 0)$ dengan jari-jari 5.
- Lingkaran kedua: pusat di $(10, 0)$ dengan jari-jari 5.
- Titik $(3, 3)$ diperiksa.

Output:

```
Titik di dalam lingkaran 1
```

Karena titik hanya berada di lingkaran pertama.

VIII. KESIMPULAN

Kesimpulan Modul 3: Fungsi

Modul ini menjelaskan tentang konsep fungsi dalam pemrograman, termasuk deklarasi, pemanggilan, dan penerapan fungsi dalam berbagai konteks. Fungsi digunakan untuk melakukan operasi tertentu dan mengembalikan nilai yang dapat digunakan di bagian lain dari program.

1. **Definisi Fungsi:** Fungsi adalah rangkaian instruksi yang menerima input (parameter), memprosesnya, dan mengembalikan suatu nilai. Fungsi selalu memiliki tipe nilai yang dikembalikan, dan menggunakan kata kunci `return` untuk mengembalikan hasil.
2. **Deklarasi dan Pemanggilan Fungsi:** Fungsi dideklarasikan terpisah dari program utama, dan dapat dipanggil menggunakan nama fungsi dengan argumen yang sesuai. Fungsi dapat digunakan dalam ekspresi atau disimpan dalam variabel.
3. **Contoh Program:** Modul ini memberikan contoh aplikasi fungsi dalam menghitung volume dan permutasi. Fungsi faktorial digunakan dalam perhitungan permutasi untuk menghitung hasil faktorial dan mengembalikan nilai yang diperlukan untuk permutasi.
4. **Penerapan Fungsi dalam Program:** Pemrograman dengan fungsi memudahkan pemisahan logika dalam bagian-bagian yang terstruktur. Penggunaan fungsi memungkinkan kode yang lebih bersih, modular, dan dapat digunakan kembali.
5. **Modul ini sangat berguna untuk memahami cara kerja fungsi dalam pemrograman dan bagaimana fungsi dapat digunakan untuk meningkatkan efisiensi dan keterbacaan kode.**

REFERENSI

MODUL 3. FUNGSI