

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 3**

**FUNCTION**



Oleh:

Raja Muhammad Lufhti

103112400027

12 IF 01

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

# **I. DASAR TEORI**

## **1. Pengertian Fungsi**

Fungsi adalah bagian dari program yang dirancang untuk melakukan tugas tertentu. Fungsi menerima masukan (input), memprosesnya, dan menghasilkan keluaran (output).

## **2. Karakteristik Fungsi**

- ↳ Mempunyai Nama: Sebagai identitas fungsi.
- ↳ Menerima Parameter (Opsional): Masukan yang digunakan dalam proses.
- ↳ Melakukan Proses: Menjalankan perintah tertentu.
- ↳ Mengembalikan Nilai (Opsional): Hasil dari proses dapat digunakan kembali.

## **3. Manfaat Fungsi**

- ↳ Modularitas: Program lebih terstruktur dan mudah dikelola.
- ↳ Menghindari Duplikasi Kode: Fungsi bisa digunakan berulang tanpa menulis ulang kode.
- ↳ Mempermudah Pemeliharaan: Perubahan hanya perlu dilakukan pada fungsi terkait.
- ↳ Meningkatkan Kejelasan: Membuat kode lebih mudah dibaca dan dipahami.

## **4. Cara Kerja Fungsi**

1. Pendefinisian: Fungsi dibuat dengan nama, parameter, dan instruksi.
2. Pemanggilan: Fungsi dipanggil di dalam program untuk dieksekusi.
3. Proses Data: Fungsi memproses input sesuai instruksi.
4. Pengembalian Hasil (Opsional): Fungsi memberikan output jika diperlukan.

## **5. Contoh Penerapan dalam Kehidupan Sehari-hari**

- ↳ Mesin ATM: Masukkan jumlah uang → ATM memproses → Uang keluar.
- ↳ Kalkulator: Masukkan angka → Proses perhitungan → Hasil ditampilkan.
- ↳ Memasak Teh: Tambah teh dan gula → Seduh air panas → Teh siap disajikan.

## **6. Kesimpulan**

Fungsi membantu membuat program lebih rapi, efisien, dan mudah dikelola. Dengan memahami konsep ini, program dapat dikembangkan dengan lebih sistematis dan profesional.

## II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

Contoh 1

```
//Raja Muhammad Lufhti 103112400027

package main

import "fmt"

func main() {
    var to, lol int
    fmt.Scan(&to, &lol)

    if to >= lol {
        fmt.Println(permutasi(to, lol))
    } else {
        fmt.Println(permutasi(lol, to))
    }
}

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)
}
```

Screenshots Output

```
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\guided 1\1.go"
5 3
60
PS D:\ALPRO 2\modul 3 alpro> █
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menghitung **permutasi** dari dua angka yang dimasukkan pengguna. Jika angka pertama lebih kecil, posisinya ditukar. Permutasi dihitung dengan rumus  $P(n,r)=\frac{n!}{(n-r)!}$  menggunakan fungsi faktorial. Jika angka kedua lebih besar dari yang pertama, hasilnya **0**. Hasil akhirnya ditampilkan ke layar.

## Contoh 2

```
package main

import (
    "fmt"
)

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)

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

    }

}

fmt.Println("Suhu dalam Fahrenheit:")

for _, temp := range temperatures {

    fmt.Printf("%.2f\n",
celsiusToFahrenheit(temp))

}

}

```

Screenshots Output

```
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\guided 2\2.go"
Masukkan jumlah data: 3
Masukkan suhu dalam Celsius:
16
25
32
Suhu dalam Fahrenheit:
60.80
77.00
89.60
PS D:\ALPRO 2\modul 3 alpro> █
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menghitung hasil dari tiga fungsi matematika yang dikombinasikan, yaitu  $a(x)=x^2$ ,  $b(x)=x-2$ , dan  $c(x)=x+1$ . Pengguna memasukkan tiga angka, lalu program menghitung hasil dari  $fogog(x) = a(b(c(x)))$ ,  $gohof(x) = b(c(a(x)))$ , dan  $hofog(x) = c(a(b(x)))$  dengan menerapkan fungsi secara berurutan. Hasil akhirnya ditampilkan ke layar.

### Contoh 3

```
//Raja Muhammad Lufhti 103112400027
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
    }
}
```

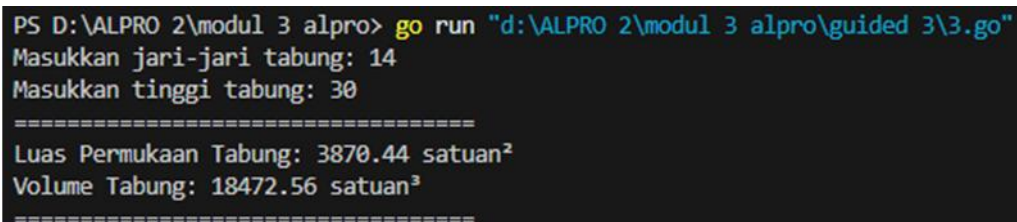
```

// 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("=====")
}

```

#### Screenshots Output



```

PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\guided 3\3.go"
Masukkan jari-jari tabung: 14
Masukkan tinggi tabung: 30
=====
Luas Permukaan Tabung: 3870.44 satuan²
Volume Tabung: 18472.56 satuan³
=====

```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menghitung **luas permukaan** dan **volume tabung** berdasarkan jari-jari dan tinggi yang dimasukkan pengguna. Setelah memastikan input valid dan positif, program menghitung luas dengan rumus  $2 \pi r(r+t)$  dan volume dengan  $\pi r^2 t$ , lalu menampilkan hasilnya. Jika input salah atau negatif, program memberi peringatan



### III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

#### Soal 1

```
//Raja Muhammad Lufhti 103112400027
package main

import (
    "fmt"
)

func main() {
    var a, b, c, d int
    fmt.Print("Masukkan empat angka (a b c d): ")
    fmt.Scan(&a, &b, &c, &d)

    fmt.Println("Hasil:")
    fmt.Printf("P(%d, %d) = %d\n", a, c, permutation(a, c))
    fmt.Printf("C(%d, %d) = %d\n", a, c, combination(a, c))
    fmt.Printf("P(%d, %d) = %d\n", b, d, permutation(b, d))
    fmt.Printf("C(%d, %d) = %d\n", b, d, combination(b, d))
}

func factorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * factorial(n-1)
}

func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}
```

Screenshots Output

```

PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un1\un1.go"
Masukkan empat angka (a b c d): 5 10 3 10
Hasil:
P(5, 3) = 60
C(5, 3) = 10
P(10, 10) = 3628800
C(10, 10) = 1
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un1\un1.go"
Masukkan empat angka (a b c d): 8 0 2 0
Hasil:
P(8, 2) = 56
C(8, 2) = 28
P(0, 0) = 1
C(0, 0) = 1
PS D:\ALPRO 2\modul 3 alpro> 

```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menghitung **permutasi** dan **kombinasi** dari empat angka yang dimasukkan pengguna. Setelah menerima input, program menggunakan fungsi factorial() untuk menghitung faktorial, lalu menghitung permutasi dengan rumus  $P(n,r)=\frac{n!}{(n-r)!}$  dan kombinasi dengan  $C(n,r)=\frac{n!}{r!(n-r)!}$ . Hasil perhitungan ditampilkan ke layar dalam format yang sesuai.

## Soal 2

```
//Raja Muhammad Lufhti 103112400027
```

```
package main
```

```
import "fmt"
```

```
func main() {  
    var k, l, m int  
    fmt.Scan(&k, &l, &m)  
    fmt.Println("fogog:", fogog(k))  
    fmt.Println("gohof:", gohof(l))  
    fmt.Println("hofog:", hofog(m))  
}
```

```
func a(x int) int {  
    return x * x  
}
```

```
func b(x int) int {  
    return x - 2  
}
```

```
func c(x int) int {  
    return x + 1  
}
```

```
func fogog(x int) int {  
    return a(b(c(x)))  
}
```

```
func gohof(x int) int {  
    return b(c(a(x)))  
}
```

```
func hofog(x int) int {  
    return c(a(b(x)))  
}
```

Screenshots Output

```
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un2\un2.go"
7 2 10
fogog: 36
gohof: 3
hofog: 65
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un2\un2.go"
5 5 5
fogog: 16
gohof: 24
hofog: 10
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un2\un2.go"
3 8 4
fogog: 4
gohof: 63
hofog: 5
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini adalah program dalam bahasa Go yang menghitung hampiran nilai akar dua berdasarkan jumlah iterasi yang diberikan sebagai input. Dengan nilai awal  $\text{root2} = \text{satu}$ , program menggunakan perulangan for untuk mengalikan  $\text{root2}$  dengan ekspresi **"(empat kali i ditambah dua) kuadrat dibagi dengan (empat kali i ditambah satu) dikali (empat kali i ditambah tiga)"** dari  $i = \text{nol}$  hingga nilai iterasi yang dimasukkan. Setelah perhitungan selesai, program mencetak hasil hampiran akar dua dengan presisi sepuluh angka di belakang koma.

### Soal 3

```
//Raja Muhammad Lufhti 103112400027
package main

import (
    "fmt"
    "math"
)

func jarak(p, b, m, n float64) float64 {
    return math.Sqrt(math.Pow(float64(m-p), 2) + math.Pow(float64(n-b), 2))
}

func dalam(xy, yx, b, x, y float64) bool {
    return jarak(xy, yx, x, y) <= float64(b)
}

func main() {
    var m1, P1, d1 float64
    fmt.Scan(&m1, &P1, &d1)

    var m2, P2, d2 float64
    fmt.Scan(&m2, &P2, &d2)

    var x, y float64
    fmt.Scan(&x, &y)

    dalamLingkaran1 := dalam(m1, P1, d1, x, y)
    dalamLingkaran2 := dalam(m2, P2, d2, x, y)

    if dalamLingkaran1 && dalamLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalamLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dalamLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

## Screenshots Output

```
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un3\un3.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un3\un3.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un3\un3.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\ALPRO 2\modul 3 alpro> go run "d:\ALPRO 2\modul 3 alpro\un3\un3.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menentukan apakah suatu titik berada di dalam satu atau dua lingkaran berdasarkan koordinat pusat dan jari-jari yang dimasukkan.

Program menghitung jarak titik ke pusat lingkaran dan mengecek apakah jaraknya lebih kecil dari jari-jari. Hasilnya ditampilkan sebagai titik di dalam lingkaran 1, lingkaran 2, keduanya, atau di luar keduanya.

## IV. KESIMPULAN

### Kesimpulan Modul 3: Fungsi dalam Pemrograman

Fungsi merupakan bagian penting dalam pemrograman yang digunakan untuk memecah program menjadi blok-blok kecil yang dapat digunakan kembali. Dengan fungsi, program menjadi lebih modular, efisien, dan mudah dikelola. Fungsi memiliki parameter sebagai masukan dan dapat mengembalikan nilai sebagai keluaran. Dalam implementasinya, fungsi dapat digunakan untuk berbagai perhitungan, seperti faktorial, permutasi, kombinasi, hingga penentuan posisi titik dalam suatu lingkaran. Dengan memahami konsep fungsi, programmer dapat menulis kode yang lebih rapi, terstruktur, dan mudah dipahami.

## **V.     REFERENSI**

MODUL 3 function