

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

FUNGSI



Oleh:

NAMA: Lutfi Shidqi Mardian

NIM: 103112400077

KELAS: IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Dalam pemrograman, fungsi (function) merupakan salah satu konsep fundamental yang memungkinkan suatu program untuk dibagi menjadi bagian-bagian yang lebih kecil dan lebih terorganisir. Fungsi dalam bahasa Go (Golang) adalah sekumpulan instruksi yang menerima input, melakukan operasi tertentu, dan mengembalikan output. Penggunaan fungsi membuat kode lebih modular, dapat digunakan kembali, dan lebih mudah dibaca.

1. Pengertian Function

Fungsi adalah subprogram yang menghasilkan atau mengembalikan suatu nilai berdasarkan parameter yang diberikan. Dalam Golang, suatu subprogram dikategorikan sebagai fungsi jika memenuhi dua syarat berikut:

- **Memiliki deklarasi tipe nilai yang dikembalikan.**
- **Menggunakan kata kunci return dalam badan fungsi untuk mengembalikan nilai.**

Fungsi dalam Golang digunakan untuk berbagai keperluan, seperti:

- **Menetapkan nilai ke suatu variabel.**
- **Sebagai bagian dari ekspresi dalam operasi matematika atau logika.**
- **Sebagai argumen dalam pemanggilan fungsi lain.**

Contoh deklarasi fungsi dalam Golang:

```
func namaFungsi(parameter1, parameter2 tipeData)
tipeKembalian {
    // badan fungsi
    return nilai
}
```

2. Deklarasi Function

Deklarasi fungsi dalam Golang mirip dengan prosedur, tetapi fungsi memiliki tipe pengembalian nilai.

Contoh fungsi untuk menghitung volume tabung berdasarkan jari-jari alas dan tinggi:

```
func volumeTabung(jari_jari, tinggi int) float64 {  
    var luasAlas, volume float64  
  
    luasAlas = 3.14 * float64(jari_jari * jari_jari)  
  
    volume = luasAlas * float64(tinggi)  
  
    return volume  
}
```

Pada contoh di atas, fungsi `volumeTabung` menerima dua parameter (`jari_jari` dan `tinggi`), melakukan perhitungan, lalu mengembalikan nilai `volume`.

3. Pemanggilan Function

Pemanggilan fungsi dilakukan dengan menyebutkan nama fungsi dan memberikan argumen yang sesuai dengan parameter fungsi. Fungsi dapat:

- Disimpan dalam variabel.
- Digunakan dalam ekspresi matematis.
- Langsung dicetak menggunakan `fmt.Println()`.

Contoh pemanggilan fungsi:

```
func main() {  
    var r, t int = 5, 10  
    var v float64  
    v = volumeTabung(r, t) // Cara pemanggilan #1  
    fmt.Println(v)  
    fmt.Println(volumeTabung(7, 15)) // Cara pemanggilan #2  
}
```

4. Contoh Penggunaan Function dalam Program

Fungsi sering digunakan dalam program untuk menghindari duplikasi kode.
Contoh penggunaan fungsi dalam perhitungan faktorial dan permutasi:

```
func faktorial(n int) int {  
    hasil := 1  
    for i := 1; i <= n; i++ {  
        hasil *= i  
    }  
    return hasil  
}  
  
func permutasi(n, r int) int {  
    return faktorial(n) / faktorial(n-r)  
}  
  
func main() {  
    fmt.Println(permutasi(5, 3)) // Output: 60}
```

Pada contoh di atas, fungsi faktorial digunakan dalam fungsi permutasi, sehingga kode menjadi lebih modular dan dapat digunakan kembali.

II. GUIDED

1.

```
package main

import "fmt"

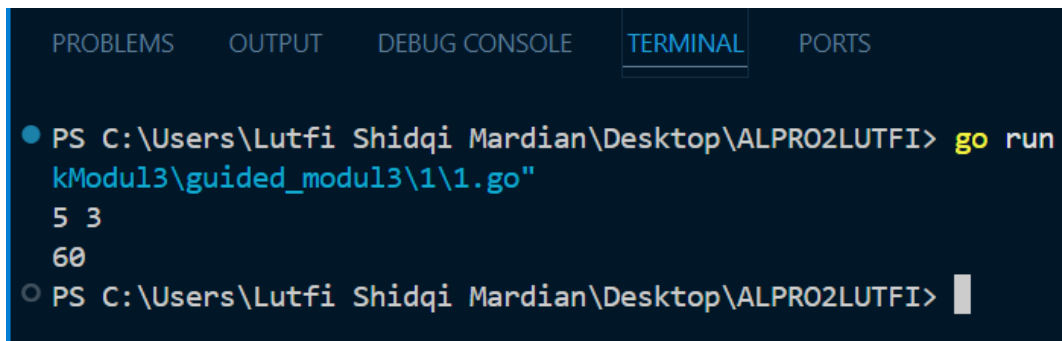
func main() {
    var a, b int
    fmt.Scan(&a, &b)

    if a >= b {
        fmt.Print(permutasi(a, b))
    } else {
        fmt.Print(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)  
}
```

Output Screenshot:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
● PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run  
  kModul3\guided_modul3\1\1.go  
  5 3  
  60  
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> |
```

Penjelasan:

Program ini menghitung dan mencetak nilai permutasi dari dua bilangan yang dimasukkan pengguna. Pengguna memberikan dua bilangan bulat a dan b, lalu program memastikan bahwa bilangan terbesar ditempatkan sebagai n dan yang lebih kecil sebagai r dalam perhitungan permutasi. Fungsi faktorial(n) menghitung faktorial suatu bilangan, sedangkan fungsi permutasi(n, r) menghitung permutasi menggunakan rumus yang sudah diberikan. Jika r lebih besar dari n, maka hasil permutasi dikembalikan sebagai 0.

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))
}
}

```

Output Screenshot:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
  kModul3\guided_modul3\2\2.go"
  Masukkan jumlah data: 3
  Masukkan suhu dalam Celsius:
  0
  100
  37.5
  Suhu dalam Fahrenheit:
  32.00
  212.00
  99.50
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>

```

Penjelasan:

Program ini mengonversi sejumlah suhu dari Celsius ke Fahrenheit. Pengguna diminta memasukkan jumlah data suhu N, lalu memasukkan N nilai suhu dalam derajat Celsius. Program menyimpan nilai suhu dalam slice `temperatures` dan memeriksa validitas input. Setiap suhu yang dimasukkan dikonversi menggunakan fungsi `celsiusToFahrenheit(celsius float64)`, yang menerapkan rumus

$$F = \frac{9}{5} C + 32$$

Hasil konversi ditampilkan dalam dua angka di belakang desimal. Jika input tidak valid, program akan menampilkan pesan kesalahan dan berhenti.

3.

```
package main

import (
    "fmt"
    "math"
)

func LuasPermukaanTabung(r, t float64) float64 {
    return 2 * math.Pi * r * (r + t)
}

func volumeTabung(r, t float64) float64 {
    return math.Pi * math.Pow(r, 2) * t
}

func main() {
    var r, t float64

    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.")

        return
    }

    if r <= 0 || t <= 0 {
        fmt.Println("Jari-jari dan tinggi tabung harus lebih dari
nol.")

        return
    }

    luas := luasPermukaanTabung(r, t)
    volume := volumeTabung(r, t)

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

```

Output Screenshot:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
  modul3\3\3.go"
  Masukkan jari-jari tabung: 7
  Masukkan tinggi tabung: 10
  =====
  Luas Permukaan Tabung: 747.70 satuan2
  Volume Tabung: 1539.38 satuan3
  =====
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> 
```

Penjelasan:

Program ini menghitung luas permukaan dan volume sebuah tabung berdasarkan jari-jari (r) dan tinggi (t) yang dimasukkan pengguna. Program pertama-tama meminta input r dan t , kemudian melakukan validasi agar input berupa angka positif. Jika input valid, program menghitung luas permukaan dengan rumus $2\pi r(r+t)$ dan volume dengan rumus $\pi r^2 t$ menggunakan fungsi `math.Pi` dan `math.Pow`. Hasil perhitungan ditampilkan dalam dua angka di belakang desimal dengan satuan kuadrat untuk luas dan kubik untuk volume. Jika input tidak valid, program akan menampilkan pesan kesalahan dan berhenti.

III. UNGUIDED

1. Kombinasi Permutasi

```
package main

import "fmt"

func main() {
    var a, b, c, d int
```

```

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

    if a >= c && b >= d {
        fmt.Printf("%d %d\n", permutasi(a, c), kombinasi(a,
c))
        fmt.Printf("%d %d\n", permutasi(b, d), kombinasi(b,
d))
    }
}

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)
}

func kombinasi(n, r int) int {

```

```
    if r > n {  
        return 0  
    }  
    return faktorial(n) / (faktorial(r) * faktorial(n-r))  
}
```

Output Screenshot:



```
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run  
kModul3\unguided_modul3\1\komb_permutasi.go  
● 5 10 3 10  
60 10  
3628800 1  
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run  
● kModul3\unguided_modul3\1\komb_permutasi.go  
8 0 2 0  
56 28  
1 1  
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>
```

Penjelasan:

Program ini menghitung nilai permutasi dan kombinasi berdasarkan empat bilangan yang dimasukkan oleh pengguna. Pengguna memberikan dua pasang bilangan (a, b) dan (c, d), lalu program memeriksa apakah setiap bilangan dalam pasangan pertama lebih besar atau sama dengan bilangan dalam pasangan kedua. Jika syarat tersebut terpenuhi, program menghitung dan mencetak hasil permutasi dan kombinasi untuk masing-masing pasangan. Permutasi menentukan jumlah susunan berbeda dari sekumpulan elemen yang diambil sebagian, sedangkan kombinasi menghitung jumlah cara memilih

sebagian elemen tanpa memperhatikan urutan. Jika kondisi awal tidak terpenuhi, program tidak memberikan output apa pun.

2. 3 Fungsi

```
package main

import "fmt"

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

    fogoh := f(g(h(a)))
    gohof := g(h(f(b)))
    hofog := h(f(g(c)))

    fmt.Println(fogoh)
    fmt.Println(gohof)
    fmt.Println(hofog)
}
```

```
func f(n int) int {  
    return n * n  
}
```

```
func g(n int) int {  
    return n - 2  
}
```

```
func h(n int) int {  
    return n + 1  
}
```

Output Screenshot:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
kModul13\unguided_modul3\2\tiga_fungsi.go"  
7 2 10  
36  
3  
65  
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run  
● kModul13\unguided_modul3\2\tiga_fungsi.go"  
5 5 5  
16  
24  
10  
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run  
● kModul13\unguided_modul3\2\tiga_fungsi.go"  
3 8 4  
4  
63  
5  
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>
```

Penjelasan:

Program ini menerima tiga bilangan bulat sebagai input dan menghitung hasil dari tiga komposisi fungsi menggunakan fungsi $f(n)$, $g(n)$, dan $h(n)$. Fungsi $f(n)$ mengkuadratkan angka, $g(n)$ mengurangi angka sebesar 2, dan $h(n)$ menambahkan angka sebesar 1. Program kemudian menghitung $\text{fogoh} = f(g(h(a)))$, $\text{gohof} = g(h(f(b)))$, dan $\text{hofog} = h(f(g(c)))$, di mana setiap fungsi diterapkan secara berurutan pada input yang diberikan. Hasil dari masing-masing komposisi kemudian dicetak dalam tiga baris terpisah.

3. Circle Center Radius

```
package main

import (
    "fmt"
    "math"
)

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)

    dalam1 := dalam(cx1, cy1, r1, x, y)
    dalam2 := dalam(cx2, cy2, r2, x, y)

    if dalam1 && dalam2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalam1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dalam2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```



```

    }

}

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

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

```

Output Screenshot:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
kModul3\unguided_modul3\3\circle_center_radius.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
● kModul3\unguided_modul3\3\circle_center_radius.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
● kModul3\unguided_modul3\3\circle_center_radius.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
● kModul3\unguided_modul3\3\circle_center_radius.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>

```

Penjelasan:

Program ini menentukan apakah suatu titik berada di dalam salah satu atau kedua lingkaran yang diberikan. Pengguna memasukkan koordinat pusat dan jari-jari dua lingkaran, serta koordinat titik yang akan diuji. Program menggunakan fungsi jarak() untuk menghitung jarak Euclidean antara titik dan pusat lingkaran, lalu fungsi dalam() untuk memeriksa apakah jarak tersebut lebih kecil atau sama dengan jari-jari lingkaran. Berdasarkan hasil pengecekan, program mencetak apakah titik berada di dalam lingkaran pertama, kedua, keduanya, atau di luar kedua lingkaran.

IV. KESIMPULAN

Dari berbagai implementasi fungsi dalam program yang telah dibuat, dapat disimpulkan bahwa penggunaan fungsi dalam pemrograman Golang sangat membantu dalam meningkatkan modularitas, keterbacaan, dan efisiensi kode. Dengan membagi program menjadi fungsi-fungsi yang lebih kecil, kita dapat menghindari duplikasi kode serta mempermudah debugging dan maintenance. Selain itu, penggunaan konsep seperti komposisi fungsi dan validasi input memastikan bahwa program berjalan dengan benar dan sesuai dengan skenario yang diharapkan.

V. REFERENSI

1. Donovan, A. A., & Kernighan, B. W. (2015). *The Go Programming Language*. Addison-Wesley.
2. Official Golang Documentation. (n.d.). Functions in Go. Retrieved from https://go.dev/doc/effective_go#functions
3. W3Schools. (n.d.). Golang Functions. Retrieved from https://www.w3schools.com/go/go_functions.asp
4. Stack Overflow. (n.d.). Idiomatic way of documenting a Golang program. Retrieved from <https://stackoverflow.com/questions/43638249>
5. GeeksforGeeks. (n.d.). Functions in Golang. Retrieved from <https://www.geeksforgeeks.org/functions-in-golang/>