

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 3
FUNGSI**



Oleh: Dimas Fanny Hebrasianto Permadi

NAMA: Dimas Ramadhani

NIM: 103112400065

KELAS: IF-12-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

1. PENGERTIAN FUNGSI

Fungsi adalah struktur yang berisi beberapa instruksi yang menghasilkan nilai sesuai dengan input yang dibutuhkan. Dikatakan fungsi karena memiliki:

- Deklarasi tipe data yang dikembalikan
- Terdapat kata kunci return untuk mengembalikan nilai atau hasil

Fungsi dapat dimanfaatkan pada:

- Assignment nilai ke variabel
- Bagian dari ekspresi matematika
- Argumen dalam suatu subprogram lain

2. DEKLARASI FUNGSI

Deklarasi fungsi mirip dengan prosedur yang berada pada blok terpisah dari program utama. Pada bagian fungsi setelah parameter ada tipe data dari nilai yang dikembalikan, lalu pada badan fungsi ada return dari nilai yang dikembalikan. Contoh notasi dalam GO:

```
func <nama function>(<parameter>) <type> {  
    return <value/variabel>  
}
```

3. PEMANGGILAN FUNGSI

Pemanggilan fungsi cukup memanggil nama fungsinya saja beserta argument yang diminta. Perbedaannya dengan prosedur adalah fungsi bisa di-assign ke suatu variabel, menjadi bagian dari ekspresi, dan argumen dari suatu subprogram.

```
func main() {  
    var r,t int  
    var v1,v2 float64  
    r = 5  
    t = 10  
    v1 = volumeTabung(r,t)  
    v2 = volumeTabung(r,t) + volumeTabung(15,t)  
    fmt.Println(volumeTabung(14,100))  
}
```

Dalam program diatas fungsi dalam di assign ke variabel lain pada saat pemanggilan, bisa dioperasikan sesuai dengan tipe data yang dikembalikan, dan juga bisa langsung ditampilkan dengan perintah output ataupun print.

GUIDED

1. Program 1

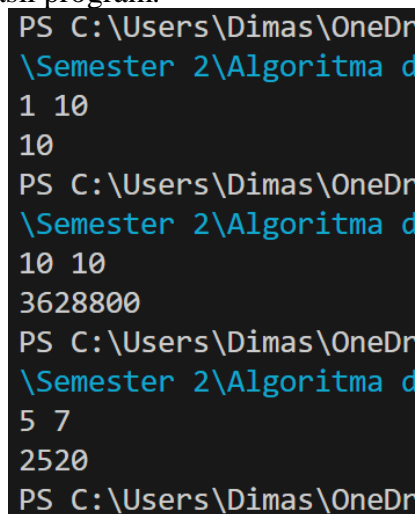
- Source Code:

```
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 {
    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)
}
```

- Screenshot hasil program:



```
PS C:\Users\Dimas\OneDr
\Semester 2\Algoritma d
1 10
10
PS C:\Users\Dimas\OneDr
\Semester 2\Algoritma d
10 10
3628800
PS C:\Users\Dimas\OneDr
\Semester 2\Algoritma d
5 7
2520
PS C:\Users\Dimas\OneDr
```

- **Penjelasan:**

Program ini berfungsi untuk menghitung permutasi dari dua variabel atau dua angka. Tipe data yang ada seperti a, b, permutasi(), factorial() adalah integer. Program berjalan dengan memasukkan inputan yang akan terbaca sebagai a dan b, lalu masuk ke percabangan Dimana jika a lebih besar maka urutan variabel untuk fungsi permutasi(a, b) dan akan tercetak hasilnya jika selain itu akan permutasi(b, a) akan tercetak juga hasilnya. Didalm program ada fungsi permutasi dengan variabel n dan juga r sebagai variabel a atau b, didalam fungsi ada percabangan jika nilai r lebih besar maka program akan berhenti selain itu akan me return fungsi factorial(n)/factorial(n-r). Untuk funsi factorial didalmnya akan mereturn hasil dengan rumus hasil sesuai dengan kodingan.

2. Program 2

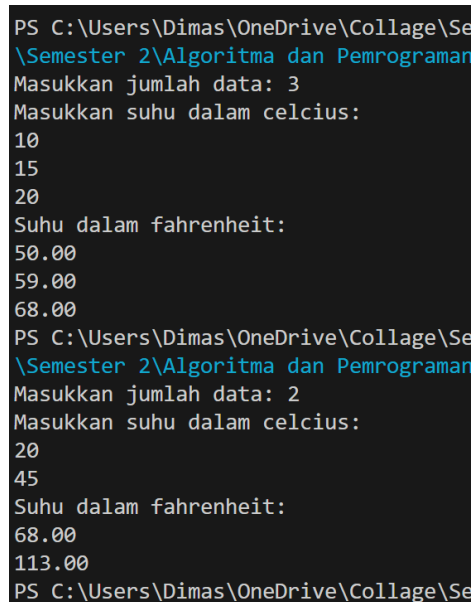
- Source Code:

```
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 masukkan angka positif")
        return
    }
    temperatures := make([]float64, N)

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

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

- Screenshoot hasil program:



```
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman>
Masukkan jumlah data: 3
Masukkan suhu dalam celcius:
10
15
20
Suhu dalam fahrenheit:
50.00
59.00
68.00
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman>
Masukkan jumlah data: 2
Masukkan suhu dalam celcius:
20
45
Suhu dalam fahrenheit:
68.00
113.00
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman>
```

- **Penjelasan:**

Program ini bertujuan untuk mengonversi suhu dari Celcius ke Fahrenheit. Pertama, program akan meminta pengguna memasukkan jumlah data suhu yang ingin diubah. Jika input yang dimasukkan bukan angka atau angka negatif, program akan menampilkan pesan "Input tidak valid, pastikan masukkan angka positif" dan berhenti. Setelah itu, program meminta pengguna untuk memasukkan suhu dalam Celcius sesuai jumlah data yang diminta. Jika input suhu tidak valid, program akan kembali menampilkan pesan kesalahan dan berhenti.

Setiap suhu yang dimasukkan akan diubah ke Fahrenheit menggunakan rumus:

$$Fahrenheit = \left(\frac{9}{5} \times Celcius \right) + 32$$

Program kemudian mencetak hasil konversi suhu dalam Fahrenheit dengan dua angka di belakang koma agar lebih mudah dibaca. Program ini memastikan input yang diberikan benar dan memberikan hasil yang sesuai berdasarkan rumus konversi suhu. Jika semua input benar, pengguna akan melihat daftar suhu dalam Fahrenheit sesuai dengan jumlah yang mereka masukkan sebelumnya.

3. Program 3

- Source code:

```
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)

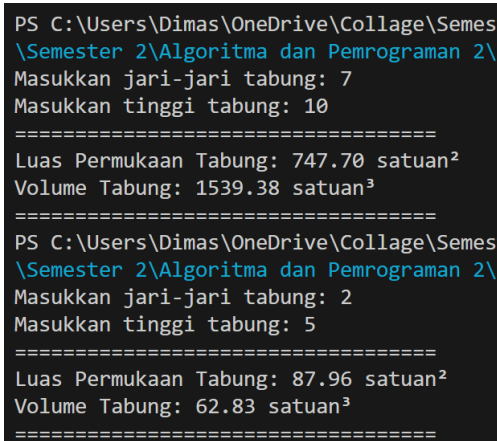
    // Memeriksa apakah input valid
    if errR != nil || errT != nil {
        fmt.Println("Input tidak valid! Harap masukkan angka yang benar.")
        return
    }

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

- Screenshoot hasil program:



```
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2\
Masukkan jari-jari tabung: 7
Masukkan tinggi tabung: 10
=====
Luas Permukaan Tabung: 747.70 satuan2
Volume Tabung: 1539.38 satuan3
=====
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2\
Masukkan jari-jari tabung: 2
Masukkan tinggi tabung: 5
=====
Luas Permukaan Tabung: 87.96 satuan2
Volume Tabung: 62.83 satuan3
=====
```

- Penjelasan:

Program di atas bertujuan untuk menghitung luas permukaan dan volume tabung menggunakan bahasa pemrograman Go. Program ini memiliki dua fungsi utama, yaitu `luasPermukaanTabung()` dan `volumeTabung()`. Fungsi `luasPermukaanTabung()` digunakan untuk menghitung luas permukaan tabung menggunakan rumus:

$$Luas\ permukaan = 2 \times \pi \times r \times (r + t)$$

Sementara itu, fungsi `volumeTabung()` digunakan untuk menghitung volume tabung dengan rumus:

$$Volume\ tabung = \pi \times r^2 \times t$$

Di dalam fungsi `main()`, program meminta pengguna untuk memasukkan jari-jari (r) dan tinggi tabung (t). Program memeriksa apakah input yang dimasukkan valid (berupa angka). Jika input tidak valid atau kurang dari atau sama dengan nol, program akan menampilkan pesan kesalahan dan berhenti. Jika input benar, program akan menghitung luas permukaan dan volume tabung menggunakan fungsi yang sudah dibuat sebelumnya.

Hasil perhitungan ditampilkan di layar dalam format yang rapi dengan dua angka di belakang koma untuk memastikan hasil lebih akurat dan mudah dibaca. Program juga mencetak garis pemisah

(==) untuk memperjelas output. Dengan validasi input dan perhitungan yang akurat, program ini memastikan bahwa hasil yang ditampilkan sesuai dengan rumus matematika yang digunakan.

II. UNGUIDED

1. Nomor 1

- Source code:

```
// Dimas Ramadhani
// 103112400065

package main
import "fmt"
func main() {
    var a, b int
    fmt.Scan(&a, &b)
    if a >= b {
        fmt.Println(permutasi(a, b), kombinasi(a, b))
    } else {
        fmt.Println(permutasi(b, a), kombinasi(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)
}

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

- Screenshoot hasil program:

```

PS C:\Users\...\Semester 2\A
5 3
60 10
PS C:\Users\...\Semester 2\A
10 10
3628800 1
PS C:\Users\...\Semester 2\A

```

- Penjelasan:

Program ini bekerja dengan menghitung permutasi dan kombinasi dari dua bilangan yang diinputkan oleh pengguna.

Pertama, program meminta dua bilangan (a dan b) sebagai input. Setelah menerima input, program memeriksa apakah a lebih besar atau sama dengan b. Jika benar, program memanggil fungsi permutasi(a, b) dan kombinasi(a, b) untuk menghitung hasilnya. Jika b lebih besar dari a, program membalik urutan input menjadi permutasi(b, a) dan kombinasi(b, a) agar perhitungan berjalan dengan benar.

Untuk menghitung permutasi dan kombinasi, program menggunakan fungsi faktorial() yang menghitung faktorial dari suatu bilangan dengan mengalikan angka dari 1 hingga n. Fungsi permutasi() menghitung hasil permutasi dengan rumus:

$$P(n, r) = \frac{n!}{(n - r)!}$$

Sedangkan fungsi kombinasi() menghitung hasil kombinasi dengan rumus:

$$C(n, r) = \frac{n!}{r! \times (n - r)!}$$

Jika nilai r lebih besar dari n, kedua fungsi ini akan mengembalikan angka 0 sebagai tanda input tidak valid. Setelah menghitung hasil permutasi dan kombinasi, program mencetak kedua nilai tersebut di layar.

2. Nomor 2

- Source code:

```
// Dimas Ramadhani
// 103112400065

package main
import "fmt"
func main() {
    var a, b, c int
    fmt.Scan(&a, &b, &c)
    fmt.Println(fogoh(a))
    fmt.Println(gohof(b))
    fmt.Println(hofog(c))
}
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 fogoh(x int) int{
    return f(g(h(x)))
}
func gohof(x int) int{
    return g(h(f(x)))
}
func hofog(x int) int{
    return h(f(g(x)))
}
```

- Screenshoot hasil program:

```
PS C:\Users\Dimas\
\Semester 2\Algori
7 2 10
36
3
65
PS C:\Users\Dimas\
\Semester 2\Algori
5 5 5
16
24
10
PS C:\Users\Dimas\
\Semester 2\Algori
3 8 4
4
63
5
PS C:\Users\Dimas\
```

- Penjelasan:

Program ini bekerja untuk menghitung hasil dari tiga komposisi fungsi matematika menggunakan input dari pengguna.

Pertama, program meminta tiga bilangan bulat (a, b, dan c) sebagai input. Setelah menerima input, program memproses masing-masing bilangan menggunakan tiga fungsi komposisi yang berbeda: fogoh(), gohof(), dan hofog().

Fungsi f(x) menghitung kuadrat dari x ($x * x$), fungsi g(x) mengurangi x dengan 2, dan fungsi h(x) menambahkan 1 ke x. Program kemudian menjalankan komposisi fungsi sebagai berikut:

- fogoh(x) menghitung hasil dari $f(g(h(x)))f(g(h(x)))$, yaitu memproses x melalui fungsi h, lalu g, dan akhirnya f.
- gohof(x) menghitung hasil dari $g(h(f(x)))g(h(f(x)))$, yaitu memproses x melalui fungsi f, kemudian h, lalu g.
- hofog(x) menghitung hasil dari $h(f(g(x)))h(f(g(x)))$, yaitu memproses x melalui fungsi g, kemudian f, dan akhirnya h.

Setelah melakukan perhitungan untuk masing-masing fungsi komposisi, program mencetak hasilnya di layar secara berurutan. Dengan cara ini, program mampu menghitung dan menampilkan hasil dari tiga komposisi fungsi untuk input yang diberikan.

3. Nomor 3

- Source code:

```
// Dimas Ramadhani
// 103112400065

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)
    Lingkaran1:= didalam(cx1, cy1, r1, x, y)
    Lingkaran2:= didalam(cx2, cy2, r2, x, y)
    if Lingkaran1 && Lingkaran2 {
        fmt.Print("Titik di dalam lingkaran 1 dan 2")
    } else if Lingkaran1 {
        fmt.Print("Titik di dalam lingkaran 1")
    } else if Lingkaran2 {
        fmt.Print("Titik di dalam lingkaran 2")
    } else {
        fmt.Print("Titik di luar lingkaran 1 dan 2")
    }
}
func didalam(cx, cy, r, x, y int) bool {
    return jarak(cx, cy, x, y) <= float64(r)
}
func jarak(a, b, c, d int) float64 {
    return math.Sqrt(math.Pow(float64(a-c), 2) + math.Pow(float64(b-d), 2))
}
```

- Screenshoot hasil program:

```

PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2>
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2>
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2>
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2>
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2

```

- Penjelasan:

Program ini bekerja untuk menentukan apakah sebuah titik berada di dalam atau di luar dua lingkaran berdasarkan koordinat yang dimasukkan oleh pengguna.

Pertama, program meminta tujuh bilangan bulat sebagai input:

- cx_1, cy_1, r_1 → koordinat pusat dan jari-jari lingkaran pertama.
- cx_2, cy_2, r_2 → koordinat pusat dan jari-jari lingkaran kedua.
- x, y → koordinat titik yang akan dicek posisinya.

Setelah menerima input, program memanggil fungsi `didalam()` dua kali untuk memeriksa apakah titik berada di dalam masing-masing lingkaran. Fungsi `didalam()` bekerja dengan memanggil fungsi `jarak()`, yang menghitung jarak antara titik pusat lingkaran dan titik yang diberikan menggunakan rumus jarak Euclidean:

$$Jarak = \sqrt{(a - c)^2 + (b - d)^2}$$

Jika jarak tersebut lebih kecil atau sama dengan jari-jari lingkaran, maka fungsi `didalam()` mengembalikan nilai `true`, yang berarti titik berada di dalam lingkaran. Jika tidak, fungsi mengembalikan `false`. Program kemudian menggunakan struktur percabangan (`if-else`) untuk memeriksa posisi titik:

- Jika titik berada di dalam kedua lingkaran, program mencetak "Titik di dalam lingkaran 1 dan 2".
- Jika titik hanya berada di dalam lingkaran pertama, program mencetak "Titik di dalam lingkaran 1".
- Jika titik hanya berada di dalam lingkaran kedua, program mencetak "Titik di dalam lingkaran 2".

- Jika titik berada di luar kedua lingkaran, program mencetak "Titik di luar lingkaran 1 dan 2".

Dengan logika ini, program dapat menentukan secara akurat posisi titik terhadap dua lingkaran.

III. KESIMPULAN

Fungsi adalah bagian penting dalam pemrograman karena memungkinkan kode lebih modular dan mudah digunakan kembali. Dalam praktiknya, fungsi digunakan untuk menyederhanakan perhitungan dan meningkatkan efisiensi kode.

IV. REFERENSI

Prayogo, N. A. (2024). *Dasar Pemrograman Golang* (Versi 4.0.20240830). Retrieved from <https://github.com/novalagung/dasarpemrogramangolang>