

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

FUNGSI



Oleh:

NAMA: ICHYA ULUMIDDIIN

NIM: 103112400076

KELAS: 12IF-01-A

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Dalam bahasa pemrograman Go (Golang), fungsi adalah sekumpulan pernyataan yang dikelompokkan untuk menjalankan tugas tertentu.

Penggunaan fungsi mempermudah pengorganisasian kode, memungkinkan penggunaan ulang, dan meningkatkan modularitas program.

- **Komponen Fungsi dalam Go:**

- Nama Fungsi: Identifikasi unik untuk setiap fungsi. Nama fungsi yang diawali dengan huruf kapital dapat diekspor dan diakses di luar paket, sedangkan yang diawali dengan huruf kecil bersifat privat dalam paket.
- Parameter: Daftar variabel input yang diterima fungsi. Setiap parameter harus memiliki tipe data yang ditentukan. Jika beberapa parameter bertipe sama, tipe tersebut dapat dituliskan sekali setelah parameter terakhir.
- Tipe Kembalian: Menentukan jenis data yang dikembalikan oleh fungsi. Fungsi dapat mengembalikan satu atau beberapa nilai. Jika tidak ada nilai yang dikembalikan, bagian ini dapat dihilangkan.
- Badan Fungsi: Blok kode yang berisi pernyataan-pernyataan yang akan dijalankan saat fungsi dipanggil.

- **Fitur Khusus Fungsi dalam Go:**

- Fungsi Variadik: Fungsi yang dapat menerima sejumlah argumen yang tidak ditentukan sebagai parameter terakhir. Ini memungkinkan fungsi menerima nol atau lebih argumen dari tipe yang sama.
- Fungsi Anonim dan Closure: Go mendukung fungsi tanpa nama yang dapat disimpan dalam variabel atau dipanggil langsung. Fungsi ini dapat mengakses variabel di lingkup sekitarnya, membentuk closure.
- Fungsi Sebagai Nilai dan Parameter: Fungsi di Go adalah tipe data first-class, artinya fungsi dapat disimpan dalam variabel, diteruskan sebagai parameter, atau dikembalikan sebagai nilai dari fungsi lain.

II. GUIDED

Source Code Guided 1

```
//ICHYA ULUMIDDIIN
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)
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIIN\OneDrive\Do
Praktikum_week2\Guided1\Guided1.go"
5 6
720
```

Penjelasan

Program ini bertujuan untuk menghitung permutasi dari dua bilangan yang dimasukkan pengguna. Fungsi `main()` membaca dua bilangan `a` dan `b`, lalu memastikan nilai terbesar digunakan sebagai `n` dalam perhitungan `P(n, r)`. Fungsi `faktorial(n)` menghitung faktorial menggunakan perulangan, sedangkan `permutasi(n, r)` menghitung permutasi dengan rumus $P(n, r) = n! / (n-r)!$, mengembalikan `0` jika `r > n`. Program memastikan bahwa perhitungan permutasi selalu valid sebelum menampilkan hasilnya.

Source Code Guided 2

```
//ICHYA ULUMIDDIIN
package main

import "fmt"

// Fungsi untuk mengonversi Celcius ke Fahrenheit
func celciusToFahrenheit(celcius float64) float64 {
    return (9.0/5.0)*celcius + 32
}

func main() {
    var N int
    fmt.Print("Masukkan jumlah data: ")
    _, err := fmt.Scan(&N)

    // Validasi input jumlah data
    if err != nil || N <= 0 {
        fmt.Println("Input tidak valid, pastikan masukkan
angka positif")
        return
    }

    temperatures := make([]float64, N) // Array untuk menyimpan
suhu dalam Celcius

    // 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
        }
    }

    // Konversi dan output hasil
    fmt.Println("Suhu dalam Fahrenheit:")
    for _, temp := range temperatures {
        fmt.Printf("%.2f\n", celciusToFahrenheit(temp)) //
Perbaikan format
    }
}
```

Output

```
Masukkan jumlah data: 3
Masukkan suhu dalam Celcius:
30 87 65
Suhu dalam Fahrenheit:
86.00
188.60
149.00
```

Penjelasan

Program ini bertujuan untuk mengonversi suhu dari Celsius ke Fahrenheit berdasarkan sejumlah data yang dimasukkan oleh pengguna. Program dimulai dengan meminta pengguna memasukkan jumlah data `N`, lalu memvalidasi apakah input tersebut adalah angka positif. Selanjutnya, program membaca `N` suhu dalam Celsius dan menyimpannya dalam array. Setelah semua data dikumpulkan, program menggunakan fungsi `celciusToFahrenheit()` untuk mengonversi setiap suhu ke Fahrenheit dan mencetak hasilnya dengan format dua desimal. Jika terjadi kesalahan input, program akan menampilkan pesan kesalahan dan berhenti.

Source Code Guided 3

```
//ICHYA ULUMIDDIIN
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("=====")
}
```

Output

```
Masukkan jari-jari tabung: 8
Masukkan tinggi tabung: 16
=====
Luas Permukaan Tabung: 1206.37 satuan2
Volume Tabung: 3216.99 satuan3
=====
```

Penjelasan

Program ini bertujuan untuk menghitung luas permukaan dan volume tabung berdasarkan jari-jari (r) dan tinggi (t) yang dimasukkan oleh pengguna. Program pertama-tama meminta input jari-jari dan tinggi tabung, lalu memvalidasi apakah input yang diberikan adalah angka dan bernilai positif. Jika input valid, program menghitung luas permukaan menggunakan rumus $2 * \pi * r * (r + t)$ dan volume menggunakan rumus $\pi * r^2 * t$. Hasil perhitungan kemudian ditampilkan dalam format dua desimal dengan satuan yang sesuai. Jika input tidak valid atau bernilai negatif, program akan menampilkan pesan kesalahan dan menghentikan eksekusi.

III. UNGUIDED

Source Code Unguided 1

```
//ICHYA ULUMIDDIIN
package main

import (
    "fmt"
)

func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    result := 1
    for i := 2; 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)
    if a >= c && b >= d {
        fmt.Printf("P(%d,%d) = %d, C(%d,%d) = %d\n", a,
c, permutasi(a, c), a, c, kombinasi(a, c))
        fmt.Printf("P(%d,%d) = %d, C(%d,%d) = %d\n", b,
d, permutasi(b, d), b, d, kombinasi(b, d))
    } else {
        fmt.Println("Input tidak valid, pastikan a >= c
dan b >= d")
    }
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\
Praktikum_week2\Unguided1\Unguided1.go"
5 10 3 10
P(5,3) = 60, C(5,3) = 10
P(10,10) = 3628800, C(10,10) = 1
```

Penjelasan

Program ini bertujuan untuk menghitung permutasi (P) dan kombinasi (C) dari dua pasang bilangan yang dimasukkan oleh pengguna. Program meminta empat bilangan bulat a, b, c, d , kemudian memastikan bahwa $a \geq c$ dan $b \geq d$ agar perhitungan valid. Fungsi `faktorial(n)` digunakan untuk menghitung faktorial suatu bilangan, yang kemudian digunakan dalam fungsi `permutasi(n, r)` dan `kombinasi(n, r)`, masing-masing untuk menghitung nilai $P(n, r) = n! / (n-r)!$ dan $C(n, r) = n! / (r!(n-r)!)$. Jika input memenuhi syarat, program akan mencetak hasil permutasi dan kombinasi untuk kedua pasangan (a, c) dan (b, d) . Jika tidak, program menampilkan pesan bahwa input tidak valid.

Source Code Unguided 2

```
//ICHYA ULUMIDDIIN
package main

import (
    "fmt"
)

func kuadrat(x int) int {
    return x * x
}

func kurangDua(x int) int {
    return x - 2
}

func tambahSatu(x int) int {
    return x + 1
}

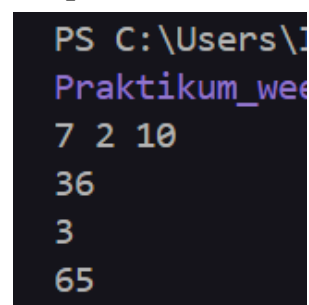
func komposisiFoGoH(x int) int {
    return kuadrat(kurangDua(tambahSatu(x)))
}

func komposisiGoHoF(x int) int {
    return kurangDua(tambahSatu(kuadrat(x)))
}

func komposisiHoFoG(x int) int {
    return tambahSatu(kuadrat(kurangDua(x)))
}

func main() {
    var a, b, c int
    fmt.Scan(&a, &b, &c)
    fmt.Println(komposisiFoGoH(a))
    fmt.Println(komposisiGoHoF(b))
    fmt.Println(komposisiHoFoG(c))
}
```

Output



```
PS C:\Users\I
Praktikum_wee
7 2 10
36
3
65
```

Penjelasan

Program ini bertujuan untuk menghitung hasil dari komposisi tiga fungsi matematika, yaitu $\text{kuadrat}(x)$, $\text{kurangDua}(x)$, dan $\text{tambahSatu}(x)$, berdasarkan nilai input yang diberikan pengguna. Program menerima tiga bilangan bulat a , b , dan c , lalu menerapkan fungsi komposisi sesuai dengan urutan tertentu: $(f \circ g \circ h)(a)$, $(g \circ h \circ f)(b)$, dan $(h \circ f \circ g)(c)$. Masing-masing komposisi fungsi ini dipanggil dalam komposisi $\text{FoGoH}(x)$, komposisi $\text{GoHoF}(x)$, dan komposisi $\text{HoFoG}(x)$, kemudian hasilnya dicetak ke layar.

Source Code Unguided 3

```
//ICHYA ULUMIDDIIN
package main

import (
    "fmt"
    "math"
)

func jarak(a, b, c, d float64) float64 {
    return math.Sqrt(math.Pow(a-c, 2) + math.Pow(b-d,
2))
}

func didalam(cx, cy, r, x, y float64) bool {
    return jarak(cx, cy, x, y) <= 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)
    dalamLingkaran2 := didalam(cx2, cy2, r2, 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")
    }
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDr  
Praktikum_week2\Unguided3\Unguide  
1 1 5  
8 8 4  
2 2  
Titik di dalam lingkaran 1
```

Penjelasan

Program ini bertujuan untuk menentukan posisi suatu titik (x, y) terhadap dua lingkaran yang didefinisikan oleh pusat $(cx1, cy1)$, $(cx2, cy2)$ dan jari-jari $r1$, $r2$. Fungsi $\text{jarak}(a, b, c, d)$ menghitung jarak antara dua titik menggunakan rumus Euclidean, sedangkan fungsi $\text{didalam}(cx, cy, r, x, y)$ mengecek apakah titik (x, y) berada di dalam suatu lingkaran.

Program membaca input koordinat pusat dan jari-jari dua lingkaran serta titik yang diuji, lalu mengevaluasi posisi titik tersebut. Berdasarkan hasil pengecekan, program mencetak apakah titik berada di dalam salah satu, kedua, atau di luar kedua lingkaran.

IV. KESIMPULAN

Pemrograman dalam Go (Golang) menekankan pada penggunaan fungsi untuk meningkatkan modularitas, efisiensi, dan keterbacaan kode. Dari berbagai implementasi yang telah dibahas, terdapat beberapa poin utama yang dapat disimpulkan:

1. Fungsi dalam Go

- Fungsi digunakan untuk membagi tugas dalam program menjadi blok yang lebih terstruktur dan reusable.
- Fungsi dapat memiliki parameter dan nilai kembalian, serta bisa dikombinasikan dalam komposisi untuk menghasilkan perhitungan yang lebih kompleks.

2. Keuntungan Penggunaan Fungsi

- Dengan menggunakan fungsi, kode menjadi lebih modular, efisien, dan mudah dipelihara.
- Penggunaan fungsi juga memungkinkan kode lebih terorganisir dan dapat digunakan kembali, mengurangi duplikasi serta mempermudah debugging.

REFERENSI

- Kurniawan, R. (2022, July 8). #20: *Function - Belajar Golang Dari Dasar*. Retrieved from ruangdeveloper: <https://blog.ruangdeveloper.com/golang-function/>
- School, D. (2019, February 13). *Pengertian Functions dalam Bahasa Pemrograman Golang Part 1*. Retrieved from DUMETSchool: <https://kursuswebprogramming.com/pengertian-functions-dalam-bahasa-pemrograman-golang-part-1/>
- Wilson, B. (2024, September 26). *Tutorial Golang: Belajar Bahasa Pemrograman Go untuk Pemula*. Retrieved from guru99: <https://www.guru99.com/id/google-go-tutorial.html>