

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

FUNGSI



Oleh:

SAVILA NUR FADILLA

103112400031

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

3.1 Definisi Function

Fungsi merupakan satu kesatuan rangkaian instruksi yang memberikan atau menghasilkan suatu nilai dan biasanya memetakan input ke suatu nilai yang lain. Oleh karena itu, fungsi selalu menghasilkan/mengembalikan nilai. Suatu subprogram dikatakan fungsi apabila:

1. Ada deklarasi tipe nilai yang dikembalikan, dan
2. Terdapat kata kunci return dalam badan subprogram.

Maka fungsi digunakan jika suatu nilai biasanya diperlukan, seperti:

- Assignment nilai ke suatu variabel
- Bagian dari ekspresi
- Bagian dari argumen suatu subprogram, dsb.

Karena itu selalu pilih nama fungsi yang menggambarkan nilai, seperti kata benda dan kata sifat. Contoh nama-nama fungsi: median, rerata, nilaiTerbesar, ketemu, selesai, ...

3.2 Deklarasi Function

```
func <nama function> (<params>) <type> {  
    /* deklarasi variabel lokal dari fungsi */  
    ...  
    /* badan algoritma fungsi */  
    ...  
    return <value/variabel>  
}
```

3.3 Cara Pemanggilan Function

```
func main() {  
    var r,t int  
    var v1,v2 float64  
    r = 5  
    t = 10  
    v1 = volumeTabung(r,t) // cara pemanggilan #1  
    v2 = volumeTabung(r,t) + volumeTabung(15,t) // cara pemanggilan #2  
    fmt.Println(volumeTabung(14,100)) // cara pemanggilan #3  
}
```

II. GUIDED

1.) Source Code

```
package main

import "fmt"

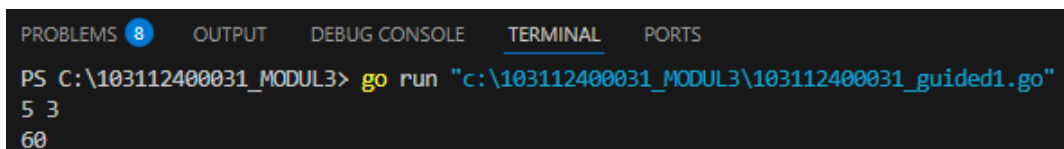
func main() {
    var a, b int
    fmt.Scan(&a, &b)

    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)
}
```

Output :



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_guided1.go"
5 3
60
```

Penjelasan : Program ini bertujuan untuk menghitung permutasi.

- ❖ func main
 - var a, b int → deklarasi variable a dan b berupa integer atau bilangan bulat
 - fmt.Scan(&a, &b) → membaca dan menerima inputan a dan b
 - if a >= b → memeriksa apakah nilai a lebih besar atau sama dengan b
 - fmt.Println(permutasi(a, b)) → jika iya, akan mencetak hasil ini
 - fmt.Println(permutasi(b, a)) → jika tidak, akan mencetak hasil ini
- ❖ func faktorial(n int) int
 - hasil := 1 → dimulai dari 1
 - for i := 1; i <= n; i++ → perulangan dari 1 hingga n
 - hasil *= i → mengalikan hasil sebelumnya dengan nilai i hingga
 - return hasil → mengembalikan hasil factorial
- ❖ func permutasi(n, r int) int
 - if r > n → jika r lebih besar dari n
 - return 0 → program akan mengembalikan 0
 - return faktorial(n) / faktorial(n-r) → menghitung hasil dengan rumus permutasi

2.) Source Code

```
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))
}
}

```

Output :

```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_guided2.go"
Masukkan jumlah data: 2
Masukkan suhu dalam Celsius:
30
40
Suhu dalam Fahrenheit:
86.00
104.00
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_guided2.go"
Masukkan jumlah data: xy
Input tidak valid, pastikan memasukkan angka positif.
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_guided2.go"
Masukkan jumlah data: -2
Input tidak valid, pastikan memasukkan angka positif.
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_guided2.go"
Masukkan jumlah data: 2
Masukkan suhu dalam Celsius:
35
zr
Input tidak valid, pastikan memasukkan angka.

```

Penjelasan : Program ini bertujuan untuk mengkonversi suhu dari Celsius ke Fahrenheit.

- ❖ `func celsiusToFahrenheit(celsius float64) float64`
 - `return (9.0/5.0)*celsius + 32` → rumus untuk mengkonversi suhu dari Celsius ke Fahrenheit
- ❖ `func main`
 - `var N int` → deklarasi variabel N berupa integer atau bilangan bulat
 - `fmt.Print("Masukkan jumlah data: ")` → mencetak pesan untuk inputan
 - `_, err := fmt.Scan(&N)` → membaca dan menerima inputan N
 - `if err != nil || N <= 0` → jika input bukan angka atau kurang dari 1
 - `fmt.Println("Input tidak valid, pastikan memasukkan angka.")`
return → program menampilkan pesan error dan program berhenti
 - `temperatures := make([]float64, N)` → membuat array slice berisi N elemen dengan tipe float64 untuk menyimpan suhu Celsius
 - `for i := 0; i < N; i++` → perulangan berjalan sampai nilai i kurang dari N
 - `_, err := fmt.Scan(&temperatures[i])` → membaca dan menerima inputan suhu
 - `if err != nil` → jika errornya bukan angka
 - `fmt.Println("Input tidak valid, pastikan memasukkan angka.")`
return → program menampilkan pesan error dan berhenti
 - `for _, temp := range temperatures` → perulangan dengan untuk mengambil semua nilai dalam suhu
 - `fmt.Printf("%.2f\n", celsiusToFahrenheit(temp))` → mencetak hasil dengan 2 angka di belakang koma

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 { // err untuk pengecekan inputan user sesuai
kemampuan program
        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 :

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_guided3.go"
Masukkan jari-jari tabung: 7
Masukkan tinggi tabung: 10
=====
Luas Permukaan Tabung: 747.70 satuan2
Volume Tabung: 1539.38 satuan3
=====
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_guided3.go"
Masukkan jari-jari tabung: x
Masukkan tinggi tabung: 5
Input tidak valid! Harap masukkan angka yang benar.
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_guided3.go"
Masukkan jari-jari tabung: -3
Masukkan tinggi tabung: 10
Jari-jari dan tinggi tabung harus lebih dari nol.
```

Penjelasan : Program ini bertujuan untuk menghitung luas permukaan dan volume tabung.

- ❖ func luasPermukaanTabung(r, t float64) float64
 - return 2 * math.Pi * r * (r + t) → menghitung luas permukaan tabung
- ❖ func volumeTabung(r, t float64) float64
 - return math.Pi * math.Pow(r, 2) * t → menghitung volume tabung
- ❖ func main
 - var r, t float64 → deklarasi variable r dan t berupa integer atau bilangan bulat, r adalah jari jari tabung dan t adalah tinggi tabung
 - _, errR := fmt.Scan(&r) → membaca dan menerima inputan r atau jari jari
 - _, errT := fmt.Scan(&t) → membaca dan menerima inputan t atau tinggi
 - if errR != nil || errT != nil → jika input tidak valid (bukan angka)
 - fmt.Println("Input tidak valid! Harap masukkan angka yang benar.")
 - return → program mencetak pesan error dan program berhenti
 - if r <= 0 || t <= 0 → jika r atau t kurang dari atau sama dengan 0
 - fmt.Println("Jari-jari dan tinggi tabung harus lebih dari nol.")
 - return → program mencetak pesan error dan program berhenti
 - luas := luasPermukaanTabung(r, t) → menghitung luas permukaan
 - volume := volumeTabung(r, t) → menghitung volume
 - mencetak hasil dengan 2 angka di belakang koma
 - fmt.Printf("Luas Permukaan Tabung: %.2f satuan²\n", luas)
 - fmt.Printf("Volume Tabung: %.2f satuan³\n", volume)

III. UNGUIDED

1.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

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))
}

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))
}
```

Output :

```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_unguided1.go"
5 10 3 10
60 10
3628800 1
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_unguided1.go"
8 0 2 0
56 28
1 1
```

Penjelasan : Program ini bertujuan untuk menghitung hasil permutasi dan kombinasi.

- ❖ func faktorial(n int) int
 - hasil := 1 → hasil dimulai dari 1
 - for i := 1; i <= n; i++ → perulangan berjalan selama nilai i itu 1 sampai dengan n
 - hasil *= i → hasil dikali dengan i
 - return hasil → mengembalikan hasil factorial
- ❖ func permutasi(n, r int) int
 - if r > n → jika r lebih dari n
 - return 0 → mengembalikan 0
 - return faktorial(n) / faktorial(n-r) → mengembalikan hasil perhitungan
- ❖ func kombinasi(n, r int) int
 - if r > n → jika r lebih dari n
 - return 0 → mengembalikan 0
 - return faktorial(n) / (faktorial(r) * faktorial(n-r)) → mengembalikan hasil perhitungan
- ❖ func main
 - var a, b, c, d int → deklarasi variable berupa integer atau bilangan bulat
 - fmt.Scan(&a, &b, &c, &d) → membaca dan menerima inputan
 - terdapat beberapa syarat, nilai a harus >= c dan b harus >= d, inputan harus berupa angka, dan inputan tidak boleh negative
 - mencetak hasil
 - fmt.Println(permutasi(a, c), kombinasi(a, c))
 - fmt.Println(permutasi(b, d), kombinasi(b, d))

2.) Source code

```
// Savila Nur Fadilla
// 103112400031

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)

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

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

Output :

```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_unguided2.go"
7 2 10
36
3
65
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_unguided2.go"
5 5 5
16
24
10
PS C:\103112400031_MODUL3> go run "c:\103112400031_MODUL3\103112400031_unguided2.go"
3 8 4
4
63
5
```

Penjelasan : Program ini bertujuan untuk menghitung fungsi komposisi.

- ❖ func f(x int) int
 - return $x * x$ → mengembalikan hasil x kuadrat
- ❖ func g(x int) int
 - return $x - 2$ → mengembalikan hasil $x - 2$
- ❖ func h(x int) int
 - return $x + 1$ → mengembalikan hasil $x + 1$
- ❖ func main
 - var a, b, c int → deklarasi variable a, b, c berupa integer atau bilangan bulat
 - fmt.Scan(&a, &b, &c) → membaca dan menerima inputan
 - menghitung hasil
 - fogoh := f(g(h(a)))
 - gohof := g(h(f(b)))
 - hofog := h(f(g(c)))
 - mencetak hasil
 - fmt.Println(fogoh)
 - fmt.Println(gohof)
 - fmt.Println(hofog)

3.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import (
    "fmt"
    "math"
)

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

func inSide(cx, cy, r, x, y int) bool {
    return hitungJarak(float64(cx), float64(cy), float64(x), float64(y)) <=
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)

    inCircle1 := inSide(cx1, cy1, r1, x, y)
    inCircle2 := inSide(cx2, cy2, r2, x, y)

    if inCircle1 && inCircle2 {
        fmt.Print("Titik di dalam lingkaran 1 dan 2")
    } else if inCircle1 {
        fmt.Print("Titik di dalam lingkaran 1")
    } else if inCircle2 {
        fmt.Print("Titik di dalam lingkaran 2")
    } else {
        fmt.Print("Titik di luar lingkaran 1 dan 2")
    }
}
```

Output :

```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODULE3> go run "c:\103112400031_MODULE3\103112400031_unguided3.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\103112400031_MODULE3> go run "c:\103112400031_MODULE3\103112400031_unguided3.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\103112400031_MODULE3> go run "c:\103112400031_MODULE3\103112400031_unguided3.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\103112400031_MODULE3> go run "c:\103112400031_MODULE3\103112400031_unguided3.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
```

Penjelasan : Program ini bertujuan untuk menentukan suatu titik berada di dalam lingkaran 1, 2, atau di luar lingkaran.

- ❖ func hitungJarak(a, b, c, d float64) float64
 - return $\text{math.Sqrt}(\text{math.Pow}(a-c, 2) + \text{math.Pow}(b-d, 2))$ → rumus menghitung jarak
- ❖ func inSide(cx, cy, r, x, y int) bool
 - return $\text{hitungJarak}(\text{float64}(cx), \text{float64}(cy), \text{float64}(x), \text{float64}(y)) \leq \text{float64}(r)$ → memanggil fungsi hitung jarak dan mengecek apakah kurang dari atau sama dengan r. Jika iya(true), titik ada di dalam lingkaran. Jika tidak(false), titik ada di luar lingkaran
- ❖ func main
 - var cx1, cy1, r1, cx2, cy2, r2, x, y int → deklarasi variable berupa integer atau bilangan bulat
 - membaca dan menerima inputan
 - `fmt.Scan(&cx1, &cy1, &r1)`
 - `fmt.Scan(&cx2, &cy2, &r2)`
 - `fmt.Scan(&x, &y)`
 - menentukan titik ada di dalam lingkaran 1 atau 2
 - `inCircle1 := inSide(cx1, cy1, r1, x, y)`
 - `inCircle2 := inSide(cx2, cy2, r2, x, y)`
 - mencetak hasil dengan if else → jika inCircle1 dan inCircle2 titik di dalam lingkaran 1 dan 2, jika inCircle1 titik di dalam lingkaran 1, jika inCircle2 titik di dalam lingkaran 2, jika tidak semuanya titik di luar lingkaran

IV. KESIMPULAN

Berdasarkan laporan ini, dapat disimpulkan jika soal soal tersebut menggunakan struktur fungsi. Struktur fungsi digunakan agar program lebih terstruktur, mudah dibaca, dan mudah dipahami. Beberapa dari program juga memuat struktur perulangan for ataupun memuat struktur percabangan if else di dalam struktur fungsinya.

- a.) Program permutasi menggunakan fungsi untuk menghitung factorial dan permutasi
- b.) Program suhu menggunakan fungsi untuk mengubah suhu dari Celsius ke Fahrenheit
- c.) Program tabung menggunakan fungsi untuk menghitung luas permukaan dan volume tabung
- d.) Program permutasi dan kombinasi menggunakan fungsi untuk menghitung factorial, permutasi, dan kombinasi
- e.) Program fungsi komposisi menggunakan komposisi fungsi
- f.) Program lingkaran menggunakan fungsi untuk menghitung jarak dan mengecek apakah suatu titik berada dalam lingkaran 1, 2, atau tidak

REFERENSI

Telkom University. (2025). *Modul Praktikum Algoritma dan Pemrograman 2*.