

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL III

FUNGSI



Disusun Oleh :

Haposan Felix Marcel Siregar / 2311102210

IF_11_06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

FUNGSI

1.1 Definisi Fungsi

Fungsi pemrograman Go adalah sekumpulan instruksi yang disusun untuk melakukan tugas tertentu, menjadikannya alat penting dalam pengembangan perangkat lunak. Kode menjadi lebih mudah dibaca dan dikelola jika programmer menggunakan fungsi untuk memecahnya menjadi bagian yang lebih kecil. Kata kunci `func` digunakan untuk mendefinisikan fungsi, diikuti oleh nama fungsi, parameter, dan tipe data yang dikembalikan. Jika diperlukan, fungsi dapat mengembalikan nilai menggunakan kata kunci `return`; namun, mereka juga dapat tidak mengembalikan nilai sama sekali.

Selain itu, Go mendukung berbagai fitur penggunaan fungsi yang canggih, seperti fungsi variabel, juga dikenal sebagai `varargs`, yang memungkinkan pengguna untuk memberikan sejumlah argumen dengan tipe yang sama. Selain itu, fungsi dapat bersifat rekursif, yang berarti bahwa fungsi dapat memanggil dirinya sendiri untuk menyelesaikan tugas. Pengolahan data sering menggunakan fungsi anonim yang tidak memiliki nama. Programmer dapat menulis kode yang lebih terstruktur dan efisien dengan memahami dan menerapkan konsep fungsi, yang memudahkan pengembangan aplikasi yang kompleks.

1.2 Deklarasi Fungsi

Dalam bahasa pemrograman Go, deklarasi fungsi adalah proses mendefinisikan fungsi yang dapat digunakan untuk melakukan operasi tertentu.

	Notasi Algoritma
1	function <nama function> (<params>) -> <type>
2	kamus
3	{deklarasi variabel lokal dari fungsi}
4	...
5	algoritma
6	{badan algoritma fungsi}
7	...
8	return <value/variabel>
9	endfunction
	Notasi dalam bahasa Go
10	func <nama function> (<params>) <type> {
11	/* deklarasi variabel lokal dari fungsi */
12	...
13	/* badan algoritma fungsi*/
14	...
15	return <value/variabel>
16	
17	}

1.3 Cara Pemanggilan Fungsi.

Pemanggilan fungsi di Go sangat mudah dan intuitif. Anda dapat memanggil fungsi dengan atau tanpa parameter, dan menggunakan berbagai teknik seperti fungsi variabel dan rekursi. Dengan memahami cara pemanggilan fungsi, Anda dapat mengorganisir kode Anda dengan lebih baik dan mengimplementasikan logika yang kompleks

dengan lebih efisien. Berikut adalah contohnya:

	Notasi Algoritma
1	program ContohProsedur
2	kamus
3	r,t : integer
4	v1,v2 : real
5	algoritma
6	r <- 5;
7	t <- 10
8	v1 <- volumeTabung(r,t) {cara pemanggilan #1}
9	v2 <- volumeTabung(r,t) + volumeTabung(15,t) {cara pemanggilan #2}
10	output(volumeTabung(14,100)) {cara pemanggilan #3}
11	endprogram
	Notasi dalam bahasa Go
12	func main() {
13	var r,t int
14	var v1,v2 float64
15	r = 5
16	t = 10
17	v1 = volumeTabung(r,t) // cara pemanggilan #1

II. GUIDED

1. Soal Studi Case

Studi kasus ini menunjukkan bagaimana program dapat digunakan untuk menghitung permutasi dengan menggunakan konsep dasar faktorial dan permutasi.

Sourcecode

```
package main

import "fmt"

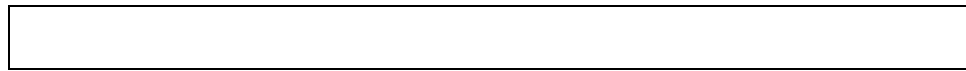
func watermark() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}

func main() {
    watermark() // Menampilkan watermark

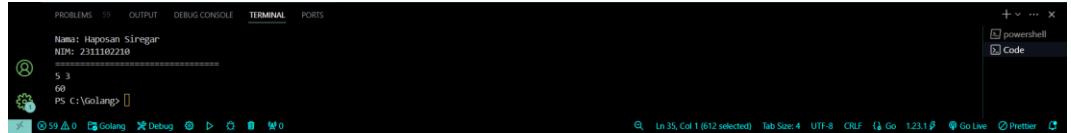
    var a, b int
    fmt.Scan(&a, &b)
    if a >= b {
        fmt.Println(permutasi(a, b))
    } else {
        fmt.Println(permutasi(b, a))
    }
}

func faktorial(n int) int {
    var hasil int = 1
    for i := 1; i <= n; i++ {
        hasil = hasil * i
    }
    return hasil
}

func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n - r)
}
```



Screenshoot Output



Deskripsi Program

- **Input yang Diberikan:**

- Pengguna memasukkan dua bilangan bulat: 5 dan 3.

- **Proses:**

- Program memeriksa apakah a (5) lebih besar atau sama dengan b (3). Karena $5 \geq 3$ adalah benar, program memanggil fungsi `permutasi(5, 3)`.
-
- Di dalam fungsi `permutasi`, program menghitung permutasi $P(n,r)P(n,r)P(n,r)$ dengan rumus:

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

- Pertama, program menghitung `faktorial(5)`:
 - $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$
- Kemudian, program menghitung `faktorial(5 - 3)` yaitu `faktorial(2)`:
 - $2! = 2 \times 1 = 2$
- Akhirnya, program menghitung permutasi: $P(5,3) = \frac{120}{2} = 60$

- **Output:**

- Program mencetak hasil perhitungan, yaitu 60.

2. Soal Studi Case

Program ini menunjukkan bagaimana menghitung Luas Permukaan dan Volume Balok. Dimana inputan berasal dari user (pengguna).

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung luas permukaan balok
func luasPermukaanBalok(panjang, lebar, tinggi float64) float64 {
    return 2 * (panjang*lebar + panjang*tinggi + lebar*tinggi)
}

// Fungsi untuk menghitung volume balok
func volumeBalok(panjang, lebar, tinggi float64) float64 {
    return panjang * lebar * tinggi
}

func watermark() {
    fmt.Println("=====")
    fmt.Println("Nama : Haposan Felix Marcel Siregar")
    fmt.Println("NIM : 2311102210")
    fmt.Println("=====")
}

func main() {
    watermark()
    var panjang, lebar, tinggi float64

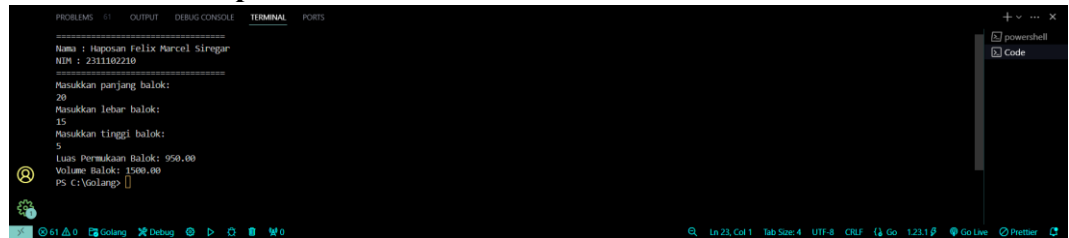
    // Input dari pengguna
    fmt.Println("Masukkan panjang balok:")
    fmt.Scan(&panjang)
    fmt.Println("Masukkan lebar balok:")
    fmt.Scan(&lebar)
    fmt.Println("Masukkan tinggi balok:")
    fmt.Scan(&tinggi)

    // Menghitung luas permukaan dan volume
    luas := luasPermukaanBalok(panjang, lebar, tinggi)
    volume := volumeBalok(panjang, lebar, tinggi)

    // Menampilkan hasil
    fmt.Printf("Luas Permukaan Balok: %.2f\n", luas)
```

```
    fmt.Printf("Volume Balok: %.2f\n", volume)
}
```

Screenshoot Output

A screenshot of a terminal window with a dark background. The terminal shows the following text: "=====", "Nama : Hapusan Felix Marcel Siregar", "NIM : 2311102210", "=====", "Masukkan panjang balok:", "20", "Masukkan lebar balok:", "15", "Masukkan tinggi balok:", "5", "Luas Permukaan Balok: 950.00", "Volume Balok: 1500.00", and "PS C:\Voolang>". The terminal has tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". On the right side, there are tabs for "powershell" and "Code". The status bar at the bottom shows "Ln 23, Col 1", "Tab Size: 4", "UTF-8", "CRLF", "Go", "1.23.1", "Go Live", and "Prettier".

Deskripsi Program

1. **Fungsi Luas Permukaan:** Fungsi luasPermukaanBalok menerima tiga parameter (panjang, lebar, tinggi) dan mengembalikan nilai luas permukaan balok menggunakan rumus:

$$2 \times (\text{panjang} \times \text{lebar} + \text{panjang} \times \text{tinggi} + \text{lebar} \times \text{tinggi})$$

2. **Fungsi Volume:** Fungsi volumeBalok juga menerima tiga parameter (panjang, lebar, tinggi) dan menghitung volume balok dengan rumus:

$$\text{panjang} \times \text{lebar} \times \text{tinggi}$$

3. **Input Pengguna:** Program meminta pengguna untuk memasukkan nilai panjang, lebar, dan tinggi balok melalui konsol, dan nilai-nilai ini disimpan dalam variabel yang sesuai.
4. **Penghitungan dan Output:** Setelah menerima input, program menghitung luas permukaan dan volume balok menggunakan fungsi yang telah didefinisikan sebelumnya dan menampilkan hasilnya ke konsol dengan format dua desimal.

III. UNGUIDED

1. Soal Studi Case

Dalam aplikasi ini, kita ingin menghitung permutasi dan kombinasi dari dua set bilangan berdasarkan input dari pengguna. Aplikasi ini berguna dalam berbagai konteks, seperti analisis statistik, penghitungan kemungkinan, dan pengelompokan data.

Sourcecode

```
package main

import "fmt"

var a,b,c,d int

// Fungsi untuk menghitung faktorial dari bilangan n
func faktorial(n int) int {
    hasil := 1
    // Loop untuk menghitung faktorial, dimulai dari 1 hingga n
    for i := 1; i <= n; i++ {
        hasil = hasil * i
    }
    return hasil
}

// Fungsi untuk menghitung permutasi  $P(n, r) = n! / (n-r)!$ 
func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

// Fungsi untuk menghitung kombinasi  $C(n, r) = n! / (r! * (n-r)!)$ 
func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}

// Fungsi untuk mencetak watermark
func watermark() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}
```

```

func main() {
    watermark() // Menampilkan watermark

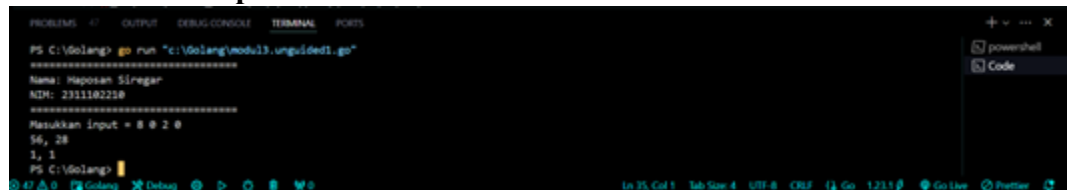
    fmt.Print("Masukkan input = ")
    fmt.Scan(&a,&b,&c,&d)

    if a >= c && b >= d {
        // Baris pertama: Permutasi dan Kombinasi a terhadap c
        fmt.Printf("%d, %d\n", permutasi(a, c), kombinasi(a, c))

        // Baris kedua: Permutasi dan Kombinasi b terhadap d
        fmt.Printf("%d, %d\n", permutasi(b, d), kombinasi(b, d))
    } else {
        fmt.Println("Syarat tidak terpenuhi: a harus >= c dan b harus >=
d")
    }
}

```

Screenshoot Output



```

PS C:\delang> go run "c:\delang\modul3\unguddef1.go"
*****
Nama: Nopisan Siregar
NIM: 2311182258
*****
Masukkan input = 8 2 0
56, 28
1, 1
PS C:\delang>

```

Deskripsi Program

- **Fungsi Utama:** Program menghitung permutasi dan kombinasi dari dua set bilangan dengan input dari pengguna.
- **Import Paket:** Menggunakan paket fmt untuk operasi input dan output.
- **Variabel Global:** Menyimpan empat bilangan bulat (a, b, c, d) yang diinput pengguna.
- **Fungsi Faktorial:** Menghitung faktorial dari bilangan bulat nnn dengan iterasi.
- **Fungsi Permutasi dan Kombinasi:** Menghitung permutasi ($P(n,r)$ $P(n,r)$ $P(n,r)$) dan kombinasi ($C(n,r)$ $C(n,r)$ $C(n,r)$) menggunakan rumus matematika yang sesuai.

- **Validasi Input:** Memeriksa syarat input agar $a \geq c$ dan $b \geq d$; menampilkan hasil perhitungan atau pesan kesalahan jika syarat tidak terpenuhi.

2. Soal Studi Case

Program ini ditulis dalam bahasa Go dan dirancang untuk mendemonstrasikan penggunaan fungsi dan komposisi fungsi. Dengan memanfaatkan tiga fungsi dasar, program ini memperlihatkan bagaimana fungsi-fungsi tersebut dapat digabungkan untuk menghasilkan fungsi baru yang lebih kompleks. Selain itu, program ini juga mencetak informasi pemilik di awal eksekusi.

Source code

```
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 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)))
}

// Fungsi untuk mencetak watermark
func watermark() {
```

```

    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}

func main() {
    watermark() // Menampilkan watermark
    var a, b, c int
    fmt.Println("Input User:")
    fmt.Scanln(&a, &b, &c)

    fmt.Println("Keluaran:")
    fmt.Println("(fogoh)(", a, ") =", fogoh(a))
    fmt.Println("(gohof)(", b, ") =", gohof(b))
    fmt.Println("(hofog)(", c, ") =", hofog(c))
}

```

Screenshoot Output

```

NIM: 2311102210
=====
Input User:
3 8 4
Keluaran:
(fogoh)( 3 ) = 4
(gohof)( 8 ) = 63
(hofog)( 4 ) = 5
PS C:\GoLang>

```

Deskripsi Program

• Fungsi Dasar:

- Terdapat tiga fungsi dasar:
 - $f(x \text{ int}) \text{ int}$: Menghitung kuadrat dari xxx (return $x^2x^2x^2$).
 - $g(x \text{ int}) \text{ int}$: Mengurangi xxx dengan 2 (return $x-2x-2x-2$).
 - $h(x \text{ int}) \text{ int}$: Menambahkan 1 pada xxx (return $x+1x+1x+1$).

• Fungsi Komposisi:

- Terdapat tiga fungsi komposisi:
 - $\text{fogoh}(x \text{ int}) \text{ int}$: Menghitung $f(g(h(x)))f(g(h(x)))f(g(h(x)))$.
 - $\text{gohof}(x \text{ int}) \text{ int}$: Menghitung $g(h(f(x)))g(h(f(x)))g(h(f(x)))$.
 - $\text{hofog}(x \text{ int}) \text{ int}$: Menghitung $h(f(g(x)))h(f(g(x)))h(f(g(x)))$.

- **Input Pengguna:**

- Dalam fungsi `main()`, program meminta pengguna untuk memasukkan tiga bilangan bulat yang akan digunakan sebagai input untuk fungsi komposisi.

- **Pemanggilan Fungsi Komposisi:**

- Program memanggil fungsi komposisi dengan input dari pengguna dan mencetak hasilnya:
 - Hasil dari `fogoh(a)`, `gohof(b)`, dan `hofog(c)` ditampilkan di konsol.

- **Alur Eksekusi:**

- Program dimulai dengan menampilkan watermark, kemudian menerima input dari pengguna, dan terakhir mencetak hasil dari fungsi komposisi berdasarkan input tersebut.

- **Output:**

- Program menghasilkan output berupa nilai dari masing-masing fungsi komposisi yang dihitung berdasarkan input pengguna, menampilkan hasil di konsol.

3. Soal Studi Case

Sebuah program ditulis dalam bahasa Go untuk menentukan posisi sebuah titik relatif terhadap dua lingkaran. Program ini menerima input dari pengguna mengenai koordinat pusat dan radius dari dua lingkaran serta koordinat titik yang akan diuji. Dengan menggunakan rumus jarak, program akan mengecek apakah titik tersebut berada di dalam, di luar, atau pada lingkaran-lingkaran tersebut.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Function to calculate the distance between two points
func distance(x1, y1, x2, y2 float64) float64 {
```

```

    return math.Sqrt(math.Pow(x1-x2, 2) + math.Pow(y1-y2, 2))
}

// Function to check if a point is inside a circle
func isInsideCircle(x, y, cx, cy, r float64) bool {
    return distance(x, y, cx, cy) <= r
}

// Fungsi untuk mencetak watermark
func watermark() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}

func main() {
    watermark() // Menampilkan watermark

    var cx1, cy1, r1 float64
    var cx2, cy2, r2 float64
    var x, y float64

    // Input for Circle 1
    fmt.Println("Masukkan koordinat pusat lingkaran 1 (cx1 cy1) dan radius (r1):")
    fmt.Scan(&cx1, &cy1, &r1)

    // Input for Circle 2
    fmt.Println("Masukkan koordinat pusat lingkaran 2 (cx2 cy2) dan radius (r2):")
    fmt.Scan(&cx2, &cy2, &r2)

    // Input for the point
    fmt.Println("Masukkan koordinat titik sembarang (x y):")
    fmt.Scan(&x, &y)

    // Check position of the point relative to the circles
    inside1 := isInsideCircle(x, y, cx1, cy1, r1)
    inside2 := isInsideCircle(x, y, cx2, cy2, r2)

    // Determine output based on the position
    if inside1 && inside2 {

```

```

        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if inside1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if inside2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

Screenshoot Output

```

PS C:\Golang> go run "c:\Golang\Unquided3\modul3.go"
=====
Nama: Haposan Siregar
NIM: 2311102210
=====
Masukkan koordinat pusat lingkaran 1 (cx1 cy1) dan radius (r1):
1 1 5
Masukkan koordinat pusat lingkaran 2 (cx2 cy2) dan radius (r2):
8 8 4
Masukkan koordinat titik sembarang (x y):
15 20
Titik di luar lingkaran 1 dan 2
PS C:\Golang>

```

Deskripsi Program

- **Menghitung Jarak:**

- Fungsi `distance` menghitung jarak antara dua titik menggunakan rumus Euclidean: $\text{jarak} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

- **Pengecekan Lingkaran:**

- Fungsi `isInsideCircle` memeriksa apakah titik (x, y) berada di dalam atau pada batas lingkaran dengan pusat (cx, cy) dan radius r . Jika jarak ke pusat lingkaran kurang dari atau sama dengan radius, fungsi mengembalikan `true`.

- **Input Data:**

- Di fungsi `main`, program meminta pengguna memasukkan koordinat pusat dan radius dari dua lingkaran, serta koordinat titik yang akan diuji.

- **Menentukan Posisi Titik:**

- Program memeriksa posisi titik relatif terhadap kedua lingkaran dan menampilkan hasilnya, apakah titik berada di dalam salah satu atau kedua lingkaran, atau di luar keduanya.