

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3
PENGENALAN FUNGSI



Oleh:

ADITHANA DHARMA PUTRA

2311102207

IF – 11- 02

S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024

I. DASAR TEORI

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 variable
- 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**

Deklarasi Function

Deklarasi fungsi sama dengan prosedur yaitu berada pada blok yang terpisah dengan program utama.

INFORMATIKA

	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	}

Pada bagian deklarasi terlihat setelah parameter terdapat tipe data dari nilai yang dikembalikan, sedangkan pada bagian badan fungsi terdapat return dari nilai yang dikembalikan.

Berikut adalah contoh fungsi untuk menghitung volume dari tabung apabila jari-jari alas dan tinggi tabung diketahui.

	Notasi Algoritma
1	function volumeTabung(jari_jari, tinggi integer) -> real
2	kamus
3	luasAlas, volume: real
4	algoritma
5	luasAlas <- 3.14 * (jari_jari * jari_jari)
6	volume <- luasAlas * tinggi
7	return volume
8	endfunction
	Notasi dalam bahasa Go
10	func volumeTabung(jari_jari, tinggi int) float64 {
11	var luasAlas, volume float64
12	luasAlas = 3.14 * float64(jari_jari * jari_jari)
13	volume = luasAlas * tinggi
14	return volume
15	}

Cara Pemanggilan Function

Sama halnya dengan prosedur pemanggilan fungsi cukup dilakukan dengan penulisan nama fungsi beserta argumen yang diminta oleh parameter dari fungsi. Perbedaannya dengan prosedur adalah fungsi bisa di-assign ke suatu variabel, menjadi bagian dari ekspresi, dan argumen dari suatu subprogram.

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
18	v2 = volumeTabung(r, t) + volumeTabung(15, t) // cara pemanggilan #2
19	fmt.Println(volumeTabung(14, 100)) // cara pemanggilan #3
20	}

Pada contoh pemanggilan fungsi di atas terlihat tidak ada perbedaan pada saat pemanggilan fungsi pada pseudocode ataupun Golang. Di sini terlihat fungsi bisa di-assign ke suatu variabel pada saat pemanggilan, bisa dioperasikan sesuai dengan tipe data yang dikembalikan, dan juga bisa langsung ditampilkan dengan perintah output ataupun print.

II. GUIDED

1. Guided

Source Code

```
package main

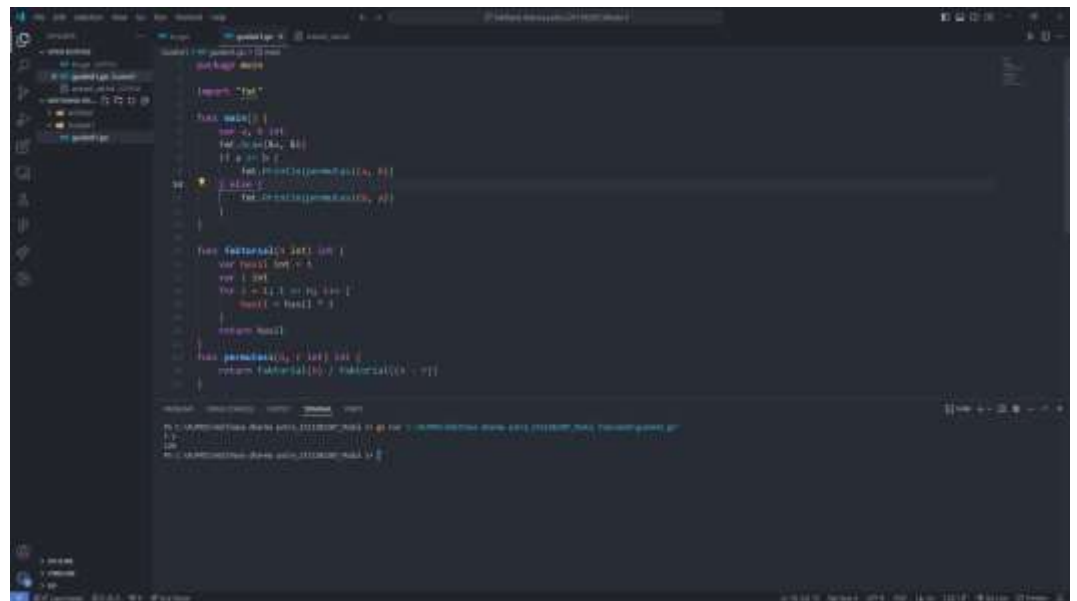
import "fmt"

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 {
    var hasil int = 1
    var i int
    for i = 1; i <= n; i++ {
        hasil = hasil * i
    }
    return hasil
}

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

Screenshot



```
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(a int) int {
    var hasil int = 1
    for i := 1; i <= a; i++ {
        hasil = hasil * i
    }
    return hasil
}

func permutasi(a, b int) int {
    return faktorial(a) / faktorial(a - b)
}
```

Deskripsi

Program ini meminta dua angka dari pengguna, kemudian menghitung dan mencetak permutasi dari angka-angka tersebut. Jika angka pertama lebih besar atau sama dengan angka kedua, permutasi dihitung dengan urutan tersebut, jika tidak, urutannya dibalik.

III. UNGUIDED

1. Unguided 1

Source Code

```
package main

import "fmt"

// Fungsi buat menghitung faktorial
func factorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
    return result
}

// Fungsi buat menghitung permutasi
func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

// Fungsi buat menghitung kombinasi
func combination(n, r int) int {
    return factorial(n) / (factorial(r) *
factorial(n-r))
}

func main() {
    // Input 4 bilangan
```

```

var a, b, c, d int
fmt.Println("Masukkan bilangan a, b, c, d
(dengan spasi): ")
fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

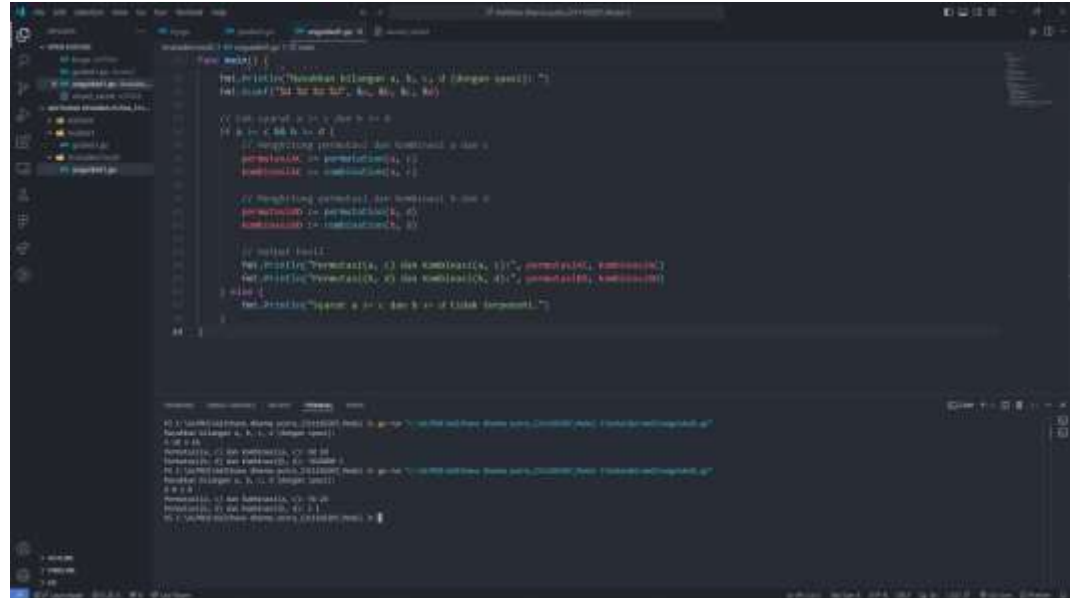
// Cek syarat a >= c dan b >= d
if a >= c && b >= d {
    // Menghitung permutasi dan kombinasi
a dan c
    permutasiAC := permutation(a, c)
    kombinasiAC := combination(a, c)

    // Menghitung permutasi dan kombinasi
b dan d
    permutasiBD := permutation(b, d)
    kombinasiBD := combination(b, d)

    // Output hasil
    fmt.Println("Permutasi(a, c) dan
Kombinasi(a, c):", permutasiAC, kombinasiAC)
    fmt.Println("Permutasi(b, d) dan
Kombinasi(b, d):", permutasiBD, kombinasiBD)
} else {
    fmt.Println("Syarat a >= c dan b >= d
tidak terpenuhi.")
}
}

```


Screenshot



```
1 // Program untuk menghitung permutasi dan kombinasi dari dua pasang bilangan
2 // yang dimasukkan oleh pengguna.
3 // Program ini menggunakan package fmt untuk input dan output.
4 // Fungsi factorial digunakan untuk menghitung faktorial dari sebuah bilangan.
5 // Fungsi permutation menghitung permutasi dengan menggunakan faktorial, dan
6 // fungsi combination menghitung kombinasi juga dengan menggunakan faktorial.
7
8 #include <fmt>
9 #include <iostream>
10
11 using namespace std;
12
13 // Fungsi untuk menghitung faktorial
14 int factorial(int n) {
15     if (n == 0 || n == 1) {
16         return 1;
17     }
18     return n * factorial(n - 1);
19 }
20
21 // Fungsi untuk menghitung permutasi
22 int permutation(int a, int b) {
23     return factorial(a) / factorial(a - b);
24 }
25
26 // Fungsi untuk menghitung kombinasi
27 int combination(int a, int b) {
28     return factorial(a) / (factorial(b) * factorial(a - b));
29 }
30
31 int main() {
32     int a, b, c, d;
33     fmt::print("Masukkan bilangan a, b, c, d (dengan spasi): ");
34     fmt::scanf("%d %d %d %d", &a, &b, &c, &d);
35
36     // Cek apakah a >= c dan b >= d
37     if (a < c || b < d) {
38         fmt::print("Syarat tidak terpenuhi. a harus lebih besar atau sama dengan c, dan b harus lebih besar atau sama dengan d.\n");
39         return 1;
40     }
41
42     // Menghitung permutasi dan kombinasi
43     int perm_ac = permutation(a, c);
44     int perm_bd = permutation(b, d);
45     int comb_ac = combination(a, c);
46     int comb_bd = combination(b, d);
47
48     // Output hasil
49     fmt::print("Permutasi(a, c) dan kombinasi(a, c): {}, {}\n", perm_ac, comb_ac);
50     fmt::print("Permutasi(b, d) dan kombinasi(b, d): {}, {}\n", perm_bd, comb_bd);
51
52     return 0;
53 }
```

Deskripsi

Program bertujuan untuk menghitung permutasi dan kombinasi dari dua pasang bilangan yang dimasukkan oleh pengguna. Pertama, program mengimpor package `fmt` untuk input dan output. Fungsi `factorial` digunakan untuk menghitung faktorial dari sebuah bilangan. Fungsi `permutation` menghitung permutasi dengan menggunakan faktorial, dan fungsi `combination` menghitung kombinasi juga dengan menggunakan faktorial.

Dalam fungsi `main`, program meminta pengguna untuk memasukkan empat bilangan (`a`, `b`, `c`, `d`). Kemudian, program memeriksa apakah `a` lebih besar atau sama dengan `c` dan `b` lebih besar atau sama dengan `d`. Jika kondisi ini terpenuhi, program menghitung permutasi dan kombinasi dari pasangan bilangan `a` dan `c`, serta `b` dan `d`, dan mencetak hasilnya. Jika kondisi tidak terpenuhi, program mencetak pesan bahwa syarat tidak terpenuhi. Program ini membantu pengguna untuk memahami dan

menghitung permutasi dan kombinasi dari bilangan yang diberikan dengan mudah.

2. Unguided 2

Source Code

```
package main

import "fmt"
import "math"

func cek(titikX, titikY, cx1, cy1, r1, cx2, cy2,
r2 float64) string {
    jarakLingkaran1 := math.Sqrt(math.Pow(titikX-
cx1, 2) + math.Pow(titikY-cy1, 2))
    JarakLingkaran2 := math.Sqrt(math.Pow(titikX-
cx2, 2) + math.Pow(titikY-cy2, 2))

    if jarakLingkaran1 <= r1 && JarakLingkaran2 <=
r2 {
        return "titik didalam lingkaran 1 dan 2"
    } else if jarakLingkaran1 <= r1 {
        return "titik didalam lingkaran 1"
    } else if JarakLingkaran2 <= r2 {
        return "titik didalam lingkaran 2"
    } else {
        return "titik diluar lingkaraan 1 dan 2"
    }
}

func main() {
    var titikX, titikY, cx1, cy1, r1, cx2, cy2, r2
float64
```

```

        fmt.Print("Masukkan pusat koordinat dan
radius lingkaran 1 (x y radius): ")

        fmt.Scan(&cx1, &cy1, &r1)

        fmt.Print("Masukkan pusat koordinat dan radius
lingkaran 2(x y radius): ")
        fmt.Scan(&cx2, &cy2, &r2)

        fmt.Print("masukkan titik koordinat (x y): ")
        fmt.Scan(&titikX, &titikY)

        Hasil := cek(titikX, titikY, cx1, cy1, r1,
cx2, cy2, r2)

        fmt.Println(Hasil)
    }

```

Screenshot

```

$ go run main.go
Masukkan pusat koordinat dan radius lingkaran 1 (x y radius):
1 1 2
Masukkan pusat koordinat dan radius lingkaran 2(x y radius):
3 3 2
masukkan titik koordinat (x y):
1 1
Masukkan Hasil:
titik berada diluar lingkaran

```

Deskripsi

Program bertujuan untuk menentukan apakah sebuah titik berada di dalam salah satu atau kedua lingkaran yang diberikan. Pertama, program

mengimpor package `fmt` untuk input dan output serta `math` untuk operasi matematika. Fungsi `cek` menerima koordinat titik dan parameter dua lingkaran (pusat dan radius), lalu menghitung jarak titik dari pusat masing-masing lingkaran menggunakan rumus jarak Euclidean. Berdasarkan jarak yang dihitung dan radius lingkaran, fungsi ini menentukan apakah titik berada di dalam salah satu atau kedua lingkaran, dan mengembalikan string yang sesuai. Fungsi `main` meminta pengguna untuk memasukkan koordinat pusat dan radius dua lingkaran serta koordinat titik, kemudian memanggil fungsi `cek` dengan parameter yang diberikan oleh pengguna, dan mencetak hasilnya. Program ini membantu pengguna memahami posisi relatif sebuah titik terhadap dua lingkaran yang diberikan.