

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 2

FUNGSI



Disusun Oleh :

Afif Rijal Azzami / 2311102235

IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

A. Definisi Fungsi

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, nilai Terbesar, ketemu, dan selesai.

B. Deklarasi Function

yaitu dengan menuliskan keyword func kemudian diikuti nama fungsi, lalu kurung () (yang bisa diisi parameter), dan diakhiri dengan kurung kurawal untuk membungkus blok kode.

	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	}

C. Pemanggilan Function

Dengan memanggil parameter pada func main, parameter merupakan variabel yang menempel di fungsi yang nilainya ditentukan saat pemanggilan fungsi tersebut. Parameter sifatnya opsional, suatu fungsi bisa tidak memiliki parameter, atau bisa saja memiliki satu atau banyak parameter (tergantung kebutuhan).

	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	

II. GUIDED 1

Sourcecode

```
package main

import "fmt"

func main() {
    var a, b int

    fmt.Print("Masukkan bilangan a = ")

    fmt.Scan(&a)

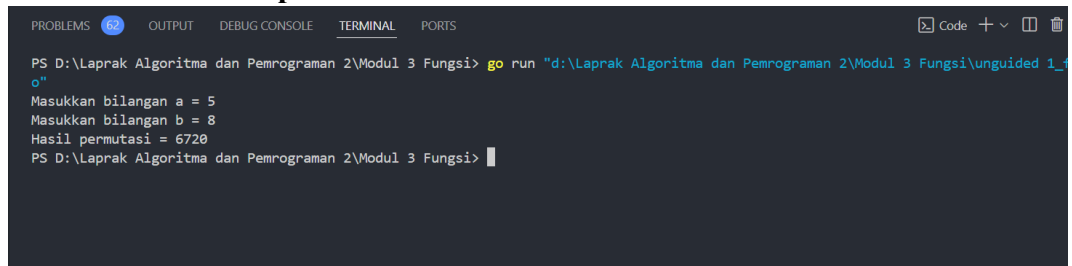
    fmt.Print("Masukkan bilangan b = ")

    fmt.Scan(&b)

    if a >= b {
        fmt.Println("Hasil permutasi =", permutasi(a, b))
    } else {
        fmt.Println("Hasil permutasi =", permutasi(b, a))
    }
}
```

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

Screenshot Output

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the command 'go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi\unguided 1_o"' and its output: 'Masukkan bilangan a = 5', 'Masukkan bilangan b = 8', and 'Hasil permutasi = 6720'. The prompt 'PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi>' is visible at the bottom.

```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi\unguided 1_o"  
Masukkan bilangan a = 5  
Masukkan bilangan b = 8  
Hasil permutasi = 6720  
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi>
```

Deskripsi Program

Sintaks tersebut adalah program untuk mencari hasil dari permutasi menggunakan fungsi, untuk menghitungnya pertama kita menggunakan pengondisian if, karna untuk mencari permutasi nilai $n \geq r$, kemudian kita membuat fungsi yang diberi nama faktorial, fungsi tersebut untuk mencari hasil dari faktorial dengan perulangan dan mengalikan setiap bilangan yang diulang, kemudian kita membuat fungsi yang diberi nama permutasi, fungsi tersebut berisikan rumus dari permutasi yaitu faktorial dari nilai n dibagi dengan faktorial dari hasil pengurangan n dan r ($n-r$).

III. GUIDED 2

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung luas persegi
func hitungLuasPersegi(sisi int) int {
    return sisi * sisi
}

// Fungsi untuk menghitung keliling persegi
func hitungKelilingPersegi(sisi int) int {
    return 4 * sisi
}

func main() {
    // Meminta input panjang sisi persegi
    var panjangSisi int

    fmt.Print("Masukkan panjang sisi persegi: ")

    fmt.Scan(&panjangSisi)
```

```
// Menghitung luas dan keliling persegi menggunakan fungsi

luas := hitungLuasPersegi(panjangSisi)

keliling := hitungKelilingPersegi(panjangSisi)


// Menampilkan hasil

fmt.Printf("Luas persegi: %d\n", luas)

fmt.Printf("Keliling persegi: %d\n", keliling)

}
```

Screenshoot Output



```
PROBLEMS 100 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi\unguided 2.go"
Masukkan panjang sisi persegi: 10
Luas persegi: 100
Keliling persegi: 40
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi>
```

Deskripsi Program

Sintaks tersebut adalah program untuk menghitung luas dan panjang pada persegi menggunakan fungsi. Pertama kita membuat fungsi yang dinamai `hitungLuasPersegi` dan memasukan rumus yaitu sisi dikali sisi, dan fungsi `hitung keliling persegi` yang di dalamnya terdapat rumus yaitu sisi dikali 4. kemudian pada func main kita panggil fungsi yg sudah dibuat dan nantinya ketika user meng-input sisi persegi, hasil dari operasi luas dan keliling persegi akan ditampilkan.

IV. UNGUIDED 1

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung faktorial
func faktorial(n int) int {
    if n == 0 {
        return 1
    }
    hasil := 1
    for i := 1; i <= n; i++ {
        hasil *= i
    }
    return hasil
}

// Fungsi untuk menghitung permutasi
func permutasi(n, r int) int {
    if n < r {
        return 0 // Tidak valid
    }
    return faktorial(n) / faktorial(n-r)
}
```



```

// Fungsi untuk menghitung kombinasi

func kombinasi(n, r int) int {

    if n < r {

        return 0 // Tidak valid

    }

    return faktorial(n) / (faktorial(r) * faktorial(n-r))

}

func main() {

    var a, b, c, d int

    fmt.Print("Masukkan nilai a dan b (dengan a >= c dan b >=
d): ")

    fmt.Scan(&a, &b)

    fmt.Print("Masukkan nilai c dan d: ")

    fmt.Scan(&c, &d)

    if a >= c && b >= d {

        // Baris pertama: Hasil permutasi dan kombinasi a
        terhadap c

        fmt.Printf("Permutasi P(%d, %d) = %d\n", a, c,
permutasi(a, c))

        fmt.Printf("Kombinasi C(%d, %d) = %d\n", a, c,
kombinasi(a, c))

```

```

/ Baris kedua: Hasil permutasi dan kombinasi b terhadap d

    fmt.Printf("Permutasi P(%d, %d) = %d\n", b, d,
permutasi(b, d))

    fmt.Printf("Kombinasi C(%d, %d) = %d\n", b, d,
kombinasi(b, d))

} else {

    fmt.Println("Pastikan bahwa a >= c dan b >= d.")

}

}

```

Screenshot Output

```

PROBLEMS 243 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi\unguided 1.go"
Masukkan nilai a dan b (dengan a >= c dan b >= d): 15 10
Masukkan nilai c dan d: 9 6
Permutasi P(15, 9) = 1816214400
Kombinasi C(15, 9) = 5005
Permutasi P(10, 6) = 151200
Kombinasi C(10, 6) = 210
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi>

```

```

PROBLEMS 243 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi\unguided 1.go"
Masukkan nilai a dan b (dengan a >= c dan b >= d): 15 10
Masukkan nilai c dan d: 7 10
Pastikan bahwa a >= c dan b >= d.
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi>

```

Deskripsi Program

Sintaks tersebut adalah program untuk menghitung permutasi dan kombinasi menggunakan fungsi dengan syarat $a \geq c$ dan $b \geq d$. Pertama kita buat fungsi yang dinamai faktorial untuk menghitung hasil faktorial dari bilangan yang diinputkan dengan mengalikan setiap bilangan pada perulangan, kemudian kita membuat fungsi yang dinamai permutasi untuk menghitung permutasi dengan rumus faktorial dari n dibagi dengan hasil faktorial dari n dikurangi r , jika $n < r$ maka akan mengembalikan 0 karna tidak valid, kemudian kita buat fungsi yang dinamai kombinasi untuk menghitung kombinasi dengan rumus faktorial dari n dibagi dengan hasil faktorial r dikalikan dengan faktorial dari $n - r$

V. UNGUIDED 2

Sourcecode

```
package main

import (
    "fmt"
)

// fungsi f(x), g(x), h(x)
func f(x int) int {
    return x * x
}

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

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

// Fungsi komposisi untuk fogoh(x), gohof(x), dan hofog(x)
func fogoh(x int) int {
    return f(g(h(x)))
}

func gohof(x int) int {
```

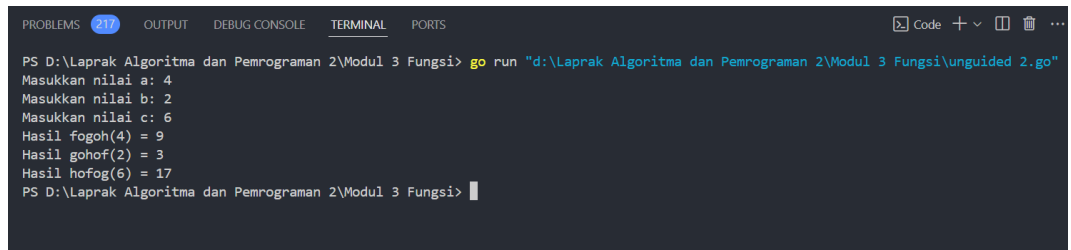
```
return h(f(g(x)))
}

func main() {
    var a, b, c int

    // Input bilangan bulat a, b, c
    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)
    fmt.Print("Masukkan nilai c: ")
    fmt.Scan(&c)

    // Output dengan keterangan
    fmt.Printf("Hasil fogoh(%d) = %d\n", a, fogoh(a)) //
fogoh(a)
    fmt.Printf("Hasil gohof(%d) = %d\n", b, gohof(b)) //
gohof(b)
    fmt.Printf("Hasil hofog(%d) = %d\n", c, hofog(c)) //
hofog(c)
}
```

Screenshoot Output



```
PROBLEMS 217 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi\unguided 2.go"
Masukkan nilai a: 4
Masukkan nilai b: 2
Masukkan nilai c: 6
Hasil fogoh(4) = 9
Hasil gohof(2) = 3
Hasil hofog(6) = 17
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi> |
```

Deskripsi Program

Sintaks tersebut adalah program untuk menghitung fungsi matematika dari $f(x)$, $g(x)$, dan $h(x)$. Pertama kita membuat fungsi f, g , dan h yang di dalamnya masing-masing terdapat operasi terhadap nilai x , kemudian kita membuat fungsi $fogoh, gohof$, dan $hofog$, fungsi tersebut untuk menghitung komposisi fungsi dengan rumus $f(g(h(x)))$ untuk mencari nilai f dan begitu juga yang lainnya, cara hitungnya dengan mencari hasil fungsi yang paling dalam $h(x)$ dan hasil dari $h(x)$ sebagai input ke fungsi berikutnya $g(hasil\ h)$ dan seterusnya.

VI. UNGUIDED 3

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung jarak antara dua titik
func hitungJarak(x1, y1, x2, y2 int) float64 {
    return math.Sqrt(float64((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2)))
}

// Fungsi untuk mengecek apakah suatu titik (x, y) berada di
dalam lingkaran
func cekdiDalamLingkaran(pusatX, pusatY, radius, titikX, titikY
int) bool {
```

```
return hitungJarak(pusatX, pusatY, titikX, titikY) <=
float64(radius)
}

func main() {

    // Membaca input

    var pusatX1, pusatY1, radius1 int // Koordinat dan radius
    lingkaran 1

    var pusatX2, pusatY2, radius2 int // Koordinat dan radius
    lingkaran 2

    var titikX, titikY int           // Koordinat titik sembarang

    fmt.Print("Masukkan koordinat dan radius lingkaran 1: ")
    fmt.Scanln(&pusatX1, &pusatY1, &radius1)

    fmt.Print("Masukkan koordinat dan radius lingkaran 2: ")
    fmt.Scanln(&pusatX2, &pusatY2, &radius2)

    fmt.Print("Masukkan koordinat titik: ")
    fmt.Scanln(&titikX, &titikY)

    // Mengecek posisi titik (titikX, titikY) terhadap lingkaran
    1 dan lingkaran 2

    diLingkaran1 := cekdiDalamLingkaran(pusatX1, pusatY1,
    radius1, titikX, titikY)
```

```
diLingkaran2 := cekdiDalamLingkaran(pusatX2, pusatY2, radius2,
titikX, titikY)

    // Menentukan output berdasarkan posisi titik

    if diLingkaran1 && diLingkaran2 {

        fmt.Println("\nTitik tersebut berada di dalam kedua
lingkaran")

    } else if diLingkaran1 {

        fmt.Println("\nTitik tersebut berada di dalam lingkaran
pertama saja")

    } else if diLingkaran2 {

        fmt.Println("\nTitik tersebut berada di dalam lingkaran
kedua saja")

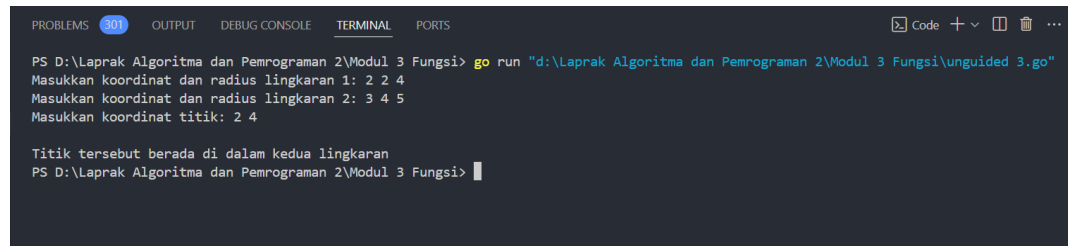
    } else {

        fmt.Println("\nTitik tersebut tidak berada di dalam kedua
lingkaran")

    }

}
```

Screenshoot Output



```
PROBLEMS 301 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi\unguided 3.go"
Masukkan koordinat dan radius lingkaran 1: 2 2 4
Masukkan koordinat dan radius lingkaran 2: 3 4 5
Masukkan koordinat titik: 2 4

Titik tersebut berada di dalam kedua lingkaran
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi>
```



```
PROBLEMS 301 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi\unguided 3.go"
Masukkan koordinat dan radius lingkaran 1: 3 4 6
Masukkan koordinat dan radius lingkaran 2: 5 6 8
Masukkan koordinat titik: 12 15

Titik tersebut tidak berada di dalam kedua lingkaran
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 3 Fungsi>
```

Deskripsi Program

Sintaks tersebut adalah program untuk memeriksa apakah suatu kordinat titik berada di dalam satu atau dua lingkaran. Untuk mengetahui apakah titik tersebut berada di suatu lingkaran pertama kita membuat fungsi yang dinamai hitungJarak dengan rumus jarak euclidean, akar dari $(x_1 - y_1)^2 + (x_2 - y_2)^2$. Kemudian kita membuat fungsi cekDalanLingkaran untuk mengecek jika jarak antara titik dan pusat lingkaran lebih kecil atau sama dengan radius lingkaran, maka titik tersebut berada di dalam lingkaran, dan jika jarak tersebut lebih besar dari radius, maka titik berada di luar lingkaran.