

LAPORAN PRAKTIKUM
Algoritma Pemrograman
MODUL III
FUNGSI



Oleh:

Ilhan Sahal Mansiz

2311102029

IF-11-02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

A. Definisi Fungsi

Dalam bahasa Go, definisi fungsi menggabungkan deklarasi dan implementasi. Setiap fungsi didefinisikan dengan keyword `func` diikuti dengan nama fungsi, parameter (jika ada), tipe data dari nilai yang dikembalikan (jika ada), dan blok kode yang berisi logika atau proses yang dilakukan oleh fungsi.

```
package main

import "fmt"

// Definisi fungsi
func luasPersegi(sisi float64) float64 {
    return sisi * sisi
}

func main() {
    hasil := luasPersegi(4.0)
    fmt.Println("Luas persegi adalah:", hasil)
}
```

Penjelasan:

- `func` adalah keyword untuk mendefinisikan fungsi.
- `luasPersegi` adalah nama fungsi.
- `sisi float64` adalah parameter yang diterima fungsi dengan tipe data `float64`.
- `float64` setelah tanda kurung adalah tipe data dari nilai yang dikembalikan oleh fungsi.
- `return` digunakan untuk mengembalikan hasil dari `sisi * sisi`.

B. Deklarasi Fungsi

Dalam bahasa Go, deklarasi fungsi langsung disertai dengan definisi. Tidak seperti bahasa C atau C++, Go tidak memisahkan deklarasi dan definisi fungsi. Jadi, saat Anda menulis fungsi di Go, Anda sekaligus melakukan deklarasi dan definisi.

Penjelasan Deklarasi dan Definisi dalam Go

- Nama fungsi diletakkan setelah keyword func.
- Parameter fungsi (jika ada) ditulis dalam tanda kurung dengan tipe datanya. Jika tidak ada parameter, kurung kosong tetap diperlukan.
- Tipe data dari nilai yang dikembalikan (jika ada) ditulis setelah parameter.
- Blok kode fungsi ditulis dalam kurung kurawal { }.

Contoh Fungsi tanpa Nilai Balik:

```
package main

import "fmt"

// Fungsi tanpa nilai balik
func tampilkanPesan() {
    fmt.Println("Halo, ini adalah fungsi tanpa return value.")
}

func main() {
    tampilkanPesan()
}
```

Pada contoh ini, fungsi tampilkanPesan tidak memiliki parameter dan tidak mengembalikan nilai apa pun. Ini dikenal sebagai fungsi dengan return type void di bahasa lain.

Contoh Fungsi dengan Beberapa Parameter dan Nilai Balik:

```
package main

import "fmt"

// Fungsi dengan dua parameter dan nilai balik
func tambah(a int, b int) int {
    return a + b
}

func main() {
    hasil := tambah(3, 5)
    fmt.Println("Hasil penjumlahan adalah:", hasil)
}
```

Penjelasan:

- tambah adalah nama fungsi.
- Fungsi ini menerima dua parameter a dan b, keduanya bertipe int.
- Fungsi ini mengembalikan nilai bertipe int, yaitu hasil penjumlahan dari a + b.

C. Pemanggilan Fungsi

Dalam bahasa Go, pemanggilan fungsi sangat sederhana. Fungsi dipanggil dengan menuliskan nama fungsi dan memberikan nilai argumen yang sesuai dengan parameter yang diterima oleh fungsi tersebut. Fungsi dapat dipanggil dari fungsi lainnya, termasuk dari fungsi main yang merupakan entry point dari program Go.

Contoh Pemanggilan Fungsi:

```
package main

import "fmt"

// Definisi fungsi dengan parameter dan return value
func tambah(a int, b int) int {
    return a + b
}

func main() {
    // Pemanggilan fungsi
    hasil := tambah(3, 4) // Memanggil fungsi tambah dengan nilai 3
    dan 4
    fmt.Println("Hasil penjumlahan:", hasil) // Menampilkan hasil
}
```

Penjelasan:

- Fungsi tambah menerima dua parameter a dan b, lalu mengembalikan hasil penjumlahan keduanya.
- Pada fungsi main, fungsi tambah dipanggil dengan argumen 3 dan 4, lalu hasilnya disimpan ke dalam variabel hasil.
- Nilai hasil kemudian dicetak menggunakan fmt.Println.

II. GUIDED

Berikut ini adalah contoh penulisan fungsi pada suatu program lengkap. Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilai faktoria dan permutasi. Masukan terdiri dari dua buah bilangan positif a dan b. Keluaran berupa sebuah bilangan bulat yang menyatakan nilai a permutasi b apabila $a > b$ atau b pemutasi a untuk kemungkinan yang lain.

```
package main

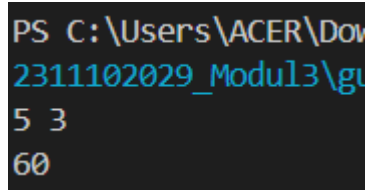
import "fmt"

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 Program :

A screenshot of a terminal window showing the output of a Go program. The prompt is 'PS C:\Users\ACER\Down\2311102029_Modul3\gu'. The user has entered '5 3' and the program has output '60'.

```
PS C:\Users\ACER\Down\2311102029_Modul3\gu
5 3
60
```

Deskripsi Program :

Program di atas merupakan program Go yang menghitung nilai permutasi dari dua bilangan integer yang dimasukkan oleh pengguna. Program pertama-tama meminta input dua angka `a` dan `b`. Kemudian, ia membandingkan nilai `a` dan `b`, memastikan bahwa nilai yang lebih besar diperlakukan sebagai jumlah total objek (`n`) dan nilai yang lebih kecil sebagai jumlah objek yang diambil (`r`). Fungsi `faktorial` digunakan untuk menghitung faktorial dari sebuah bilangan, dan fungsi `permutasi` menghitung permutasi dengan rumus $n! / (n-r)!$. Hasil perhitungan permutasi kemudian ditampilkan di layar.

III. UNGUIDED

[Lingkaran] Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx, cy) dengan radius r. Apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang (x, y) berdasarkan dua lingkaran tersebut.

Masukan terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2 sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat. Keluaran berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 2", "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2".

Source Code :

```
package main

import (
    "fmt"
    "math"
)

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

func didalam(cx, cy, r, x, y float64) bool {
    return jarak(cx, cy, x, y) <= r
}

func main() {
    var x1, y1, r1 float64
    fmt.Scan(&x1, &y1, &r1)

    var x2, y2, r2 float64
    fmt.Scan(&x2, &y2, &r2)

    var xt, yt float64
    fmt.Scan(&xt, &yt)

    dalamLingkaran1 := didalam(x1, y1, r1, xt, yt)
```



```

dalamLingkaran2 := didalam(x2, y2, r2, xt, yt)

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")
}
}

```

Screenshot Program :

```

PS C:\Users\ACER\Downloads\Semester 3\Lapr
2311102029_Modul3\unguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\Users\ACER\Downloads\Semester 3\Lapr
2311102029_Modul3\unguided1.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
PS C:\Users\ACER\Downloads\Semester 3\Lapr
2311102029_Modul3\unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Users\ACER\Downloads\Semester 3\Lapr

```

Deskripsi Program :

Program ini bertujuan untuk menentukan posisi suatu titik (x, y) relatif terhadap dua lingkaran yang didefinisikan oleh pusat dan radius masing-masing. Pertama, program meminta input koordinat pusat dan radius untuk kedua lingkaran.

Selanjutnya, program juga meminta input koordinat titik yang akan diuji. Dengan menggunakan fungsi `jarak()`, program menghitung jarak antara titik yang diuji dan pusat masing-masing lingkaran, kemudian menggunakan fungsi `didalam()` untuk menentukan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran. Akhirnya, program mencetak hasil analisis posisi titik, apakah berada di dalam lingkaran pertama, kedua, kedua lingkaran, atau di luar kedua lingkaran.