

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III
FUNGSI**



Disusun Oleh :

Marsep Trianto Pakondo / 2311102251

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 merupakan aspek penting dalam pemrograman. Definisi fungsi sendiri adalah sekumpulan blok kode yang dibungkus dengan nama tertentu. Penerapan fungsi yang tepat akan menjadikan kode lebih modular dan juga dry (kependekan dari don't repeat yourself), karena tak perlu menuliskan banyak proses berkali-kali, cukup sekali saja dan tinggal panggil jika dibutuhkan.

- **Penerapan Fungsi**

Sebenarnya tanpa sadar, kita sudah menerapkan fungsi di bab-bab sebelum ini, yaitu pada fungsi main . Fungsi main merupakan fungsi yang paling utama pada program Golang. Cara membuat fungsi cukup mudah, yaitu dengan menuliskan keyword func , diikuti setelahnya nama fungsi, kurung yang berisikan parameter, dan kurung kurawal untuk membungkus blok kode. Parameter sendiri adalah variabel yang disisipkan pada saat pemanggilan fungsi. Berikut adalah contoh penerapan fungsi.

```
package main

import "fmt"

import "strings"

func main() {

    var names = []string{"John", "Wick"}

    printMessage("halo", names)

}

func printMessage(message string, arr []string) {

    var nameString = strings.Join(arr, " ")
```

```
fmt.Println(message, nameString)

}
```

Pada kode di atas, fungsi baru dibuat dengan nama `printMessage` memiliki 2 buah parameter yaitu string `message` dan slice string `arr` . Fungsi tersebut dipanggil dalam `main` , dengan disisipkan 2 buah data sebagai parameter, data pertama adalah string "hallo" yang ditampung parameter `message` , dan slice string `names` yang nilainya ditampung oleh parameter `arr` . Di dalam `printMessage` , nilai `arr` yang merupakan slice string digabungkan menjadi sebuah string dengan pembatas adalah karakter spasi. Penggabungan slice dapat dilakukan dengan memanfaatkan fungsi `strings.Join()` . Fungsi ini berada di dalam package `strings` .

- **Fungsi Dengan Return Value / Nilai Balik**

Sebuah fungsi bisa didesain tidak mengembalikan apa-apa (`void`), atau bisa mengembalikan suatu nilai. Fungsi yang memiliki nilai kembalian, harus ditentukan tipe data nilai baliknya pada saat deklarasi. Program berikut merupakan contoh penerapan fungsi yang memiliki return value.

```
package main

import (

    "fmt"

    "math/rand"

    "time"

)

func main() {

    rand.Seed(time.Now().Unix())
```

```

var randomValue int

randomValue = randomWithRange(2, 10)

fmt.Println("random number:", randomValue)

randomValue = randomWithRange(2, 10)

fmt.Println("random number:", randomValue)

randomValue = randomWithRange(2, 10)

fmt.Println("random number:", randomValue)

}

func randomWithRange(min, max int) int {

var value = rand.Int() % (max - min + 1) + min

return value

}

```

Di dalam fungsi `randomWithRange` terdapat proses generate angka acak, yang angka tersebut kemudian digunakan sebagai nilai kembalian. Cara menentukan tipe data nilai balik fungsi adalah dengan menuliskan tipe data yang diinginkan setelah kurung parameter. Bisa dilihat pada kode di atas, bahwa `int` merupakan tipe data nilai balik fungsi `randomWithRange`. `func randomWithRange(min, max int) int` Sedangkan cara untuk mengembalikan nilainya adalah dengan menggunakan keyword `return` diikuti data yang ingin dikembalikan. Pada contoh di atas, `return value` artinya nilai variabel `value` dijadikan nilai kembalian fungsi. Eksekusi keyword `return` akan menjadikan proses dalam blok fungsi berhenti pada saat itu juga. Semua statement setelah keyword tersebut tidak akan dieksekusi. Dari kode di atas mungkin ada beberapa statement yang masih terasa asing, berikut merupakan pembahasannya.

II. GUIDED

Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

Sourcecode

```
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

    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

```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\guided\guided1.go"
5 10
30240
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\guided\guided1.go"
5 3
60
○ PS E:\alpro 2\src> █
```

Deskripsi Program

Program diatas digunakan untuk untuk menghitung permutasi dari dua bilangan bulat yang diberikan oleh pengguna. Seperti biasa membuat struktur golang terlebih dahulu. Kemudian didalam fungsi utama dimulai dengan mendeklarasikan dua variabel integer a dan b , yang kemudian diisi melalui input pengguna menggunakan `fmt.Scan`. Setelah itu, program memeriksa apakah nilai a lebih besar atau sama dengan b . Jika benar, maka fungsi `permutasi(a, b)` dipanggil, namun jika tidak, fungsi `permutasi(b, a)` yang dipanggil, dan hasilnya dicetak. Fungsi faktorial bertugas menghitung faktorial dari bilangan bulat n dengan memulai nilai awal 1 dan mengalikan hasil tersebut dengan setiap bilangan hingga n dalam sebuah loop. Fungsi ini kemudian mengembalikan hasil faktorial. Fungsi `permutasi(n, r)` menghitung permutasi dari n elemen yang diambil r elemen dengan menggunakan rumus faktorial, yakni faktorial n dibagi dengan faktorial dari $n-r$, dan hasilnya dikembalikan sebagai nilai akhir. Program ini secara keseluruhan bertujuan untuk menghitung permutasi dua bilangan yang diberikan oleh pengguna.

Sourcecode

```
package main

import "fmt"

var p, l, t int

func luasPermukaanBalok(p, l, t int) int {
    luas := 2 * (p*l + l*t + p*t)
    return luas
}

func volumeBalok(p, l, t int) int {
    volume := p * l * t
```

```

        return volume
    }

    func input() {
        fmt.Print("Masukkan panjang balok : ")
        fmt.Scan(&p)
        fmt.Print("Masukkan lebar balok : ")
        fmt.Scan(&l)
        fmt.Print("Masukkan tinggi balok : ")
        fmt.Scan(&t)
    }

    func main() {

        input()

        lausBalok := luasPermukaanBalok(p, l, t)
        volume := volumeBalok(p, l, t)

        fmt.Printf("Luas permukaan baloknya adalah %v dan volumenya adalah %v", lausBalok, volume)
    }

```

Screenshoot Output

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\guided\guided2.go"
Masukkan panjang balok : 10
Masukkan lebar balok : 3
Masukkan tinggi balok : 3
Luas permukaan baloknya adalah 138 dan volumenya adalah 90
PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk untuk menghitung luas permukaan dan volume sebuah balok dengan meminta input dari pengguna berupa panjang, lebar, dan tinggi balok. Seperti biasa buat struktur dasar golang terlebih dahulu, setelah itu buat variabel global *p*, *l*, dan *t* untuk menyimpan panjang, lebar, dan tinggi balok. Fungsi *luasPermukaanBalok* menghitung luas permukaan balok menggunakan rumus luas permukaan balok, sementara fungsi *volumeBalok* menghitung

volume menggunakan rumus volume balok . Fungsi input meminta pengguna memasukkan nilai panjang, lebar, dan tinggi yang kemudian disimpan dalam variabel global tersebut. Di dalam fungsi main, program memanggil fungsi input untuk menerima masukan dari pengguna, kemudian memanggil fungsi luasPermukaanBalok dan volumeBalok untuk menghitung luas permukaan dan volume balok, dan hasilnya akan ditampilkan ke layar.

III. UNGUIDED

Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

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))
}

func main() {
```



```

        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

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\guided\guided3.go"
Masukkan input = 5 10 3 10
60, 10
3628800, 1
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\guided\guided3.go"
Masukkan input = 8 0 2 0
56, 28
1, 1
○ PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk menghitung faktorial, permutasi, dan kombinasi dari dua pasang bilangan bulat yang dimasukkan oleh pengguna. Setelah buat struktur dasar golang, buat empat variabel global *a*, *b*, *c*, dan *d* dideklarasikan sebagai integer untuk menyimpan nilai-nilai yang akan digunakan dalam perhitungan. Fungsi faktorial menerima satu parameter integer *n* dan menghitung faktorialnya dengan menggunakan perulangan, kemudian mengembalikan hasilnya. Fungsi permutasi menerima dua parameter integer *n* dan *r*, lalu menghitung permutasi dari *n* elemen yang diambil *r* elemen dengan memanggil fungsi faktorial untuk menghitung faktorial dari *n* dan *n-r*, dan mengembalikan hasil

perhitungan permutasi. Fungsi kombinasi bekerja serupa, namun menggunakan rumus kombinasi, dengan memanggil fungsi faktorial untuk menghitung faktorial dari n , r , dan $n-r$, lalu mengembalikan hasil kombinasi. Dalam fungsi main, program meminta input dari pengguna berupa empat bilangan bulat a , b , c , dan d . Setelah itu, program memeriksa apakah nilai a lebih besar atau sama dengan c serta b lebih besar atau sama dengan d . Jika kondisi ini terpenuhi, program akan menghitung dan mencetak hasil permutasi dan kombinasi untuk a terhadap c , serta b terhadap d . Jika kondisi tidak terpenuhi, program menampilkan pesan kesalahan.

Sourcecode

```
package main

import (
    "fmt"
)

func fungsiF (x int) int {
    return (x * x)
}
func fungsiG (x int) int {
    return (x - 2)
}
func fungsiH (x int) int {
    return (x + 1)
}

func komposisiA (x int) int {
    return fungsiF(fungsiG(fungsiH(x)))
}
func komposisiB (x int) int {
    return fungsiG(fungsiH(fungsiF(x)))
}
func komposisiC (x int) int {
    return fungsiH(fungsiF(fungsiG(x)))
}

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

```

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

        A := komposisiA(a)
        B := komposisiB(b)
        C := komposisiC(c)

        fmt.Print("\n", A)
        fmt.Print("\n", B)
        fmt.Print("\n", C)
    }

```

Screenshoot Output

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\unguided\unguided1.go"
Masukkan input = 7 2 10

36
3
65
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\unguided\unguided1.go"
Masukkan input = 5 5 5

16
24
10
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\unguided\unguided1.go"
Masukkan input = 3 8 4

4
63
5
PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk menerapkan konsep komposisi fungsi, yaitu proses di mana hasil dari satu fungsi digunakan sebagai input untuk fungsi lainnya. Untuk fungsiF, mengembalikan hasil kuadrat dari parameter x. Fungsi kedua, fungsiG, mengembalikan hasil pengurangan x dengan 2, dan fungsi ketiga, fungsiH, menambahkan 1 pada nilai x. Kemudian, terdapat tiga fungsi komposisi. Fungsi komposisiA menerapkan komposisi bertingkat dengan memanggil fungsiF(fungsiG(fungsiH(x))), yaitu fungsi fungsiH dijalankan terlebih dahulu, diikuti oleh fungsiG, lalu hasilnya diproses oleh fungsiF. Fungsi komposisiB melakukan komposisi dengan urutan fungsiG(fungsiH(fungsiF(x))), sedangkan komposisiC menerapkan urutan fungsiH(fungsiF(fungsiG(x))).

Dalam fungsi utama program meminta pengguna untuk memasukkan tiga bilangan bulat sebagai input. Setelah menerima input dari pengguna, fungsi-fungsi komposisi (komposisiA, komposisiB, dan komposisiC) dijalankan menggunakan bilangan-bilangan tersebut sebagai parameter, dan hasil dari komposisi fungsi disimpan dalam variabel A, B, dan C. Kemudian hasil komposisinya dicetak ke layar.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func jarak(x1, y1, x2, y2 float64) float64 {
    hasil := math.Sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2))
    return hasil
}

func dalamLingkaran(x, y, cx, cy, r float64) bool {
    var hasil bool
    jarak := jarak(x, y, cx, cy)
    if jarak <= r {
        hasil = true
    } else {
        hasil = false
    }
    return hasil
}

func main() {

    var a, b, c, d, e, f float64
    var x, y float64

    fmt.Print("Masukkan koordinat pusat lingkaran 1 dan radius : ")
    fmt.Scan(&a, &b, &c)

    fmt.Print("Masukkan koordinat pusat lingkaran 2 dan radius : ")
    fmt.Scan(&d, &e, &f)

    fmt.Print("Masukkan koordinat titik sembarang : ")
    fmt.Scan(&x, &y)

    dalamLingkaran1 := dalamLingkaran(x, y, a, b, c)
```

```

        dalamLingkaran2 := dalamLingkaran(x, y, d, e, f)

        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")
        }
    }
}

```

Screenshoot Output

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\unguided\unguided2.go"
Masukkan koordinat pusat lingkaran 1 dan radius : 1 2 3
Masukkan koordinat pusat lingkaran 2 dan radius : 4 5 6
Masukkan koordinat titik sembarang : 7 8
Titik di dalam lingkaran 2
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\unguided\unguided2.go"
Masukkan koordinat pusat lingkaran 1 dan radius : 1 1 5
Masukkan koordinat pusat lingkaran 2 dan radius : 8 8 4
Masukkan koordinat titik sembarang : 15 20
Titik di luar lingkaran 1 dan 2
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\unguided\unguided2.go"
Masukkan koordinat pusat lingkaran 1 dan radius : 5 10 15
Masukkan koordinat pusat lingkaran 2 dan radius : -15 4 20
Masukkan koordinat titik sembarang : 0 0
Titik di dalam lingkaran 1 dan 2
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul3\unguided\unguided2.go"
Masukkan koordinat pusat lingkaran 1 dan radius : 1 1 5
Masukkan koordinat pusat lingkaran 2 dan radius : 8 8 4
Masukkan koordinat titik sembarang : 2 2
Titik di dalam lingkaran 1
○ PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk menentukan apakah sebuah titik terletak di dalam, di luar, atau pada salah satu atau kedua lingkaran yang diberikan, menggunakan konsep jarak antara dua titik dan membandingkan jarak tersebut dengan jari-jari lingkaran. Disini mengimport `math` yang akan digunakan untuk mengambil fungsi matematika yaitu fungsi `math.Sqrt` untuk menghitung akar kuadrat. Fungsi jarak digunakan untuk menghitung jarak antara dua titik dalam koordinat Cartesian menggunakan rumus Euclidean. Fungsi ini menerima koordinat x dan y dari dua titik dan mengembalikan nilai jaraknya. Fungsi kedua, `dalamLingkaran`, memeriksa apakah titik tertentu berada di dalam atau pada lingkaran dengan memanfaatkan fungsi jarak. Fungsi ini menerima koordinat titik, koordinat pusat lingkaran, dan jari-jari lingkaran sebagai input. Setelah menghitung jarak antara titik tersebut dan pusat lingkaran,

hasilnya dibandingkan dengan jari-jari lingkaran untuk menentukan apakah titik berada di dalam atau pada lingkaran. Fungsi ini mengembalikan true jika kondisi tersebut terpenuhi dan false jika titik berada di luar lingkaran.

Di dalam fungsi utama, program meminta input dari pengguna berupa koordinat pusat dua lingkaran, jari-jari kedua lingkaran, serta koordinat sebuah titik. Setelah menerima input, program memanggil fungsi dalamLingkaran dua kali untuk memeriksa apakah titik tersebut berada di dalam atau pada lingkaran pertama dan kedua. Berdasarkan hasil pemeriksaan, program akan mencetak ke layar apakah titik berada di dalam salah satu lingkaran, keduanya, atau di luar kedua lingkaran.