

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL III

FUNGSI



Disusun Oleh :

Rasyid Nafsyarie / 2311102011

IF-11-06

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

I. DASAR TEORI

Fungsi (function) dalam bahasa pemrograman Go atau Golang merupakan blok kode yang dapat digunakan kembali dan dirancang untuk melakukan tugas tertentu. Fungsi membantu memecah program besar menjadi bagian-bagian kecil yang lebih mudah dikelola dan dipahami. Dalam Golang, fungsi adalah objek first-class, yang berarti fungsi dapat disimpan dalam variabel, dikirim sebagai argumen ke fungsi lain, atau dikembalikan dari fungsi. Setiap program Go memiliki setidaknya satu fungsi, yaitu `main()`, yang menjadi titik awal eksekusi program.

Golang mendukung berbagai fitur fungsi seperti parameter dan nilai kembalian (return value). Fungsi dapat menerima parameter input dan mengembalikan output. Parameter dapat berupa tipe data sederhana (seperti integer dan string) hingga tipe data kompleks (seperti struktur dan slice). Fungsi dapat mengembalikan lebih dari satu nilai, yang merupakan salah satu kelebihan Go. Misalnya, ketika kita memanggil suatu fungsi, hasilnya bisa berupa nilai hasil operasi dan kesalahan (error), yang membuat penanganan kesalahan menjadi lebih mudah dan terstruktur.

Fungsi di Golang juga mendukung fitur anonim (anonymous functions) dan fungsi sebagai nilai (function as value). Fungsi anonim adalah fungsi yang tidak memiliki nama, dan biasanya digunakan sebagai fungsi sementara atau inline. Dengan kemampuan untuk menjadi objek first-class, fungsi anonim sering digunakan sebagai argumen fungsi lain, terutama dalam kasus seperti concurrency atau event handling. Selain itu, Golang mendukung penggunaan fungsi closures, yang memungkinkan fungsi menyimpan dan mengakses variabel dari lingkup di mana fungsi tersebut didefinisikan, bukan hanya dari lingkup eksekusinya.

Terakhir, fungsi dalam Golang juga memainkan peran penting dalam concurrency melalui goroutines. Goroutines adalah fungsi yang berjalan secara bersamaan dengan fungsi lainnya, memungkinkan program untuk menangani beberapa

pekerjaan secara paralel. Penggunaan goroutines dalam fungsi memungkinkan pengembangan aplikasi yang lebih efisien, terutama untuk program yang memerlukan pemrosesan cepat dan asinkron. Golang, dengan dukungan bawaannya terhadap concurrency, menjadikan penggunaan fungsi sebagai dasar untuk membangun program yang skalabel dan responsif.

II. GUIDED

1. Guided 1

Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

Sourcecode

```
package main

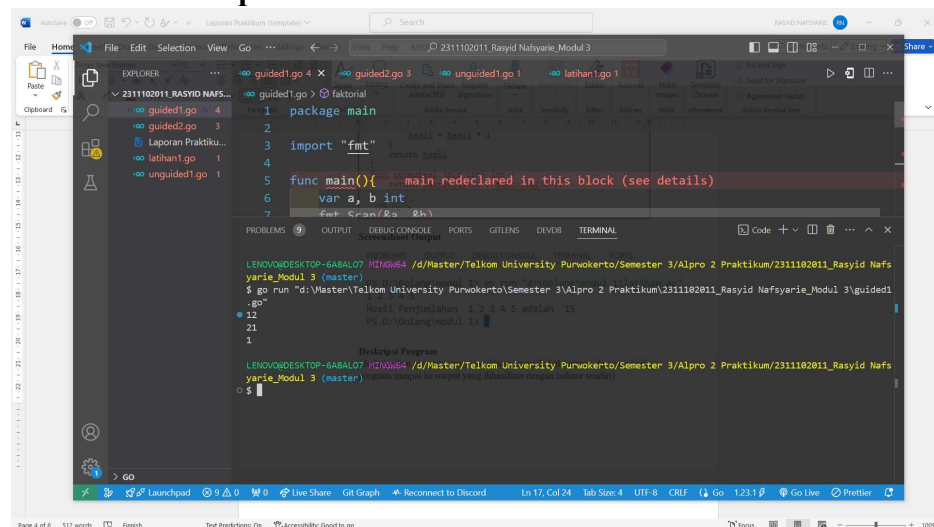
import "fmt"

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
    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)
}
```

Screenshoot Output



Deskripsi Program

Fungsi permutasi yaitu menghitung permutasi dengan menggunakan rumus yang telah dijelaskan sebelumnya, yaitu membagi faktorial dari n dengan faktorial dari n-r.

2. Guided 2

Soal Studi Case

XXXXXXXXXXXXXXXXXX

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
    }
}
```

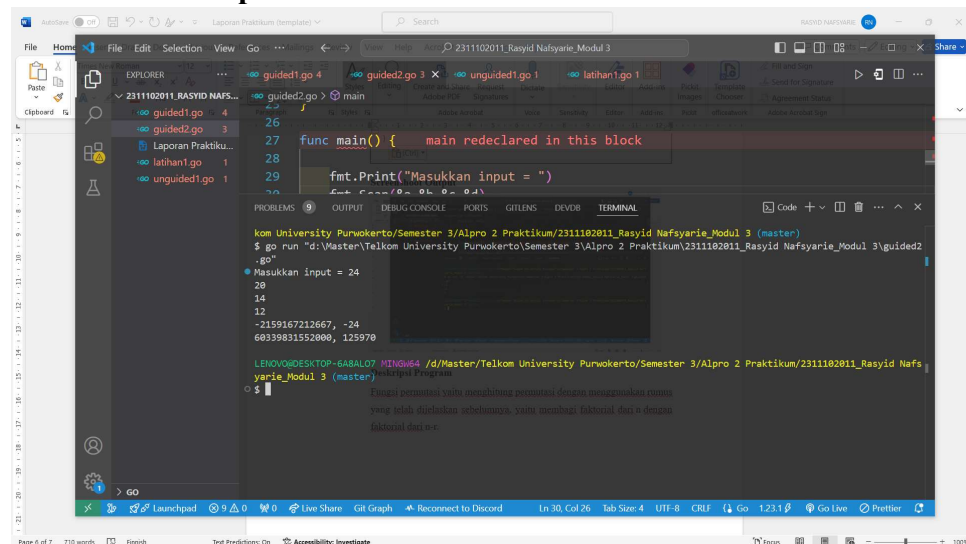
```

        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



Deskripsi Program

Program ini meminta input dari pengguna untuk variabel a, b, c, dan d. lalu memeriksa apakah syarat $a \geq c$ dan $b \geq d$ terpenuhi. Jika ya, menghitung dan menampilkan permutasi dan kombinasi. Jika tidak, menampilkan pesan kesalahan.

III. UNGUIDED

Unguided 1

Minggu Ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalian membantu Jonas? (tidak tentunya ya :p)

Masukan terdiri dari empat buah bilangan asli a , b , c , dan d yang dipisahkan oleh spasi, dengan syarat $a \geq c$ dan $b \geq$

Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi a terhadap c , sedangkan baris kedua adalah hasil permutasi dan kombinasi b terhadap d .

Catatan: permutasi (P) dan kombinasi (C) dari n terhadap r ($n \geq r$) dapat dihitung dengan menggunakan persamaan berikut!

Sourcecode

```
package main

import "fmt"

func main() {
    var a, b, c, d int
    fmt.Print("Masukkan nilai: ")
    fmt.Scan(&a, &b, &c, &d)
    if a >= c && b >= d {
        fmt.Println("Permutasi:", permutasi(a, c))
        fmt.Println("Kombinasi:", kombinasi(a, c))
        fmt.Println("Permutasi:", permutasi(b, d))
        fmt.Println("Kombinasi:", kombinasi(b, d))
    } else {
        fmt.Println("Tidak memenuhi kondisi")
    }
}

func faktorial(n int) int {
    hasil_2311102011 := 1
    for i := 1; i <= n; i++ {
        hasil_2311102011 *= i
    }
    return hasil_2311102011
}
```

```

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

```

Screenshoot Output

The screenshot shows a Go IDE with the following code in the editor:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b, c, d int
7     fmt.Println("Masukkan nilai: ")
8     fmt.Scan(&a, &b, &c, &d)
9 }

```

The terminal output shows the execution of the program:

```

$ go run "d:\Master\Telkom University Purwokerto\Semester 3\Alpro 2 Praktikum\2311102011_Rasyid Nafsyarie_Modul 3\unguided1.go"
Masukkan nilai: 10 20 21 22
Tidak memenuhi kondisi

$ go run "d:\Master\Telkom University Purwokerto\Semester 3\Alpro 2 Praktikum\2311102011_Rasyid Nafsyarie_Modul 3\unguided1.go"
Masukkan nilai: 6 12 4 8
Permutasi: 360
Kombinasi: 15
Permutasi: 19958400
Kombinasi: 495

```

Deskripsi Program

Program memeriksa apakah nilai a lebih besar atau sama dengan c dan b lebih besar atau sama dengan d. Jika tidak, program akan mencetak "Tidak memenuhi kondisi".

Unguided 2

Diberikan tiga buah fungsi matematika yaitu $f(x) = x^2$, $g(x) = x - 2$ dan $h(x) = x + 1$. Fungsi komposisi $(f \circ g \circ h)(x)$ artinya adalah $f(g(h(x)))$. Tuliskan $f(x)$, $g(x)$ dan $h(x)$ dalam bentuk function.

Masukan terdiri dari sebuah bilangan bulat a , b dan c yang dipisahkan oleh spasi.

Keluaran terdiri dari tiga baris. Baris pertama adalah $(f \circ g \circ h)(a)$, baris kedua $(g \circ h \circ f)(b)$, dan baris ketiga adalah $(h \circ f \circ g)(c)$!

Sourcecode

```
package main

import "fmt"

func main() {
    var a_2311102011, b, c int

    fmt.Scan(&a_2311102011, &b, &c)

    fmt.Println(fogoh(a_2311102011))
    fmt.Println(gohof(b))
    fmt.Println(hofog(c))
}

func f(x int) int {
    return x * x
}

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

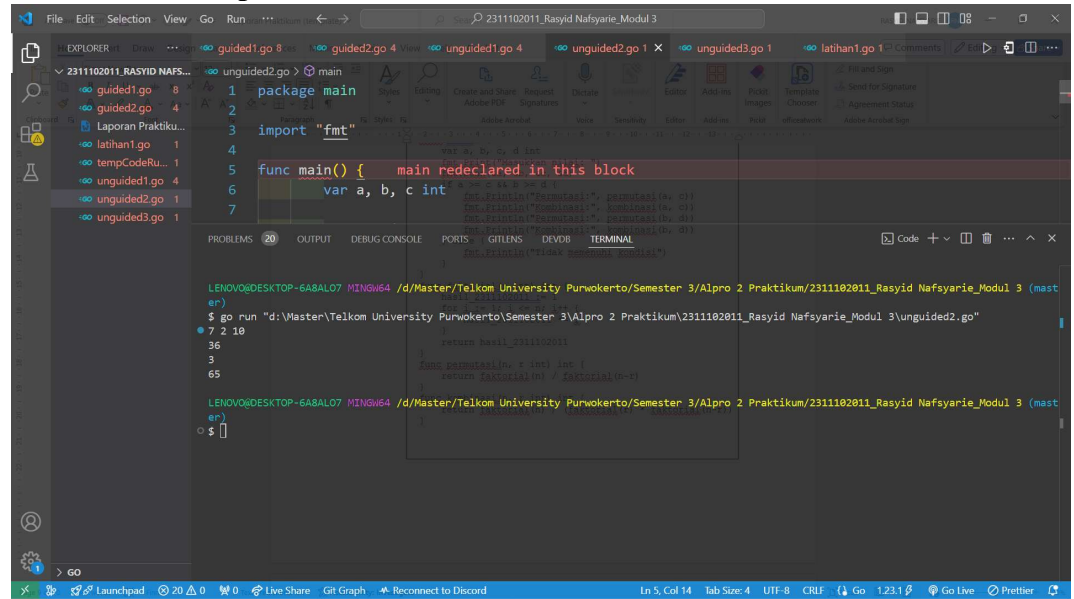
func h(x int) int {
    return x + 1
}

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)))
}
```

Screenshoot Output



Deskripsi Program

Fungsi fogoh, gohof, dan hofog dipanggil dengan argumen a, b, dan c masing-masing. Hasil perhitungan langsung dicetak ke layar.

Unguided 3

[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".

Sourcecode

```
package main

//__Rasyid Nafsyarie_2311102011_IF1106__
import (
    "fmt"
    "math"
)

type Circle struct {
    cx, cy, r float64
}

func isPointInside(c Circle, x, y float64) bool {
    distance := math.Sqrt(math.Pow(x-c.cx, 2) +
    math.Pow(y-c.cy, 2))
    return distance <= c.r
}

func determinePosition(c1, c2 Circle, x, y float64) string
{
    in1 := isPointInside(c1, x, y)
    in2 := isPointInside(c2, x, y)

    if in1 && in2 {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if in1 {
        return "Titik di dalam lingkaran 1"
    } else if in2 {
        return "Titik di dalam lingkaran 2"
    }
}
```

```

        return "Titik di luar lingkaran 1 dan 2"
    }

func main() {
    var c1, c2 Circle
    var x, y float64

    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r):")
    fmt.Scan(&c1.cx, &c1.cy, &c1.r)

    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r):")
    fmt.Scan(&c2.cx, &c2.cy, &c2.r)

    fmt.Println("Masukkan koordinat titik yang akan dicek (x y):")
    fmt.Scan(&x, &y)

    result := determinePosition(c1, c2, x, y)
    fmt.Println(result)
}

```

Screenshoot Output

```

package main

import (
    "fmt"
    "math"
)

type Circle struct {
    cx, cy float64
    r      float64
}

func determinePosition(c1, c2 Circle, x, y float64) string {
    // Implementasi logika untuk menentukan posisi titik relatif terhadap dua lingkaran
    // 1. Hitung jarak antara pusat lingkaran 1 dan 2
    d12 := math.Sqrt((c2.cx - c1.cx) * (c2.cx - c1.cx) + (c2.cy - c1.cy) * (c2.cy - c1.cy))

    // 2. Hitung jarak antara pusat lingkaran 1 dan titik (x, y)
    d1p := math.Sqrt((x - c1.cx) * (x - c1.cx) + (y - c1.cy) * (y - c1.cy))

    // 3. Hitung jarak antara pusat lingkaran 2 dan titik (x, y)
    d2p := math.Sqrt((x - c2.cx) * (x - c2.cx) + (y - c2.cy) * (y - c2.cy))

    // 4. Tentukan posisi titik berdasarkan jarak
    if d12 > d1p + d2p {
        return "Titik di luar lingkaran 1 dan 2"
    } else if d12 <= d1p + d2p {
        // Titik berada di dalam atau pada batas lingkaran 1
        if d1p <= c1.r {
            return "Titik di dalam lingkaran 1"
        } else {
            // Titik berada di dalam atau pada batas lingkaran 2
            if d2p <= c2.r {
                return "Titik di dalam lingkaran 2"
            } else {
                return "Titik di luar lingkaran 1 dan 2"
            }
        }
    }
}

func main() {
    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r):")
    var c1 Circle
    fmt.Scan(&c1.cx, &c1.cy, &c1.r)

    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r):")
    var c2 Circle
    fmt.Scan(&c2.cx, &c2.cy, &c2.r)

    fmt.Println("Masukkan koordinat titik yang akan dicek (x y):")
    var x, y float64
    fmt.Scan(&x, &y)

    result := determinePosition(c1, c2, x, y)
    fmt.Println(result)
}

```

```

LENOVO@DESKTOP-6A8AL07 MINGW64 /d:/Master/Telkom University Purwokerto/Semester 3/Aipro 2 Praktikum/2311102011_Rasyid Nafsyarie_Modul 3 (master)
$ go run "d:/Master/Telkom University Purwokerto/Semester 3/Aipro 2 Praktikum/2311102011_Rasyid Nafsyarie_Modul 3/unguided2.go"
Desktop Program
Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r):
0 0 5
Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r):
10 0 3
Masukkan koordinat titik yang akan dicek (x y):
2 2
Titik di dalam lingkaran 1

```

Deskripsi Program

Program ini memiliki fungsi `determinePosition` yaitu menentukan posisi titik relatif terhadap kedua lingkaran. Di fungsi `main`, kita meminta input dari pengguna untuk koordinat pusat dan radius kedua lingkaran, serta koordinat titik yang akan dicek.