

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III
FUNGSI**



Disusun Oleh :

SYAHRUL ROMADHONI / 2311102261

S1 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

Definisi Fungsi dalam Golang

Fungsi dalam Golang adalah blok kode yang dapat dipanggil ulang untuk melakukan tugas tertentu. Setiap fungsi memiliki nama, parameter, dan return value. Fungsi sangat penting dalam struktur logika program karena mereka memperbolehkan modifikasi dan reuse kode, sehingga meningkatkan efisiensi dan skalabilitas suatu proyek.

Keuntungan Menggunakan Fungsi di Golang:

- **Reusability (Dapat digunakan ulang):** Fungsi membantu mengurangi duplikasi kode.
- **Abstraksi:** Fungsi memungkinkan abstraksi yang memudahkan developer fokus pada tugas-tugas spesifik.
- **Modularitas:** Fungsi membantu memecah program besar menjadi unit-unit kecil yang lebih mudah dikelola.
- **Testing:** Fungsi yang terisolasi lebih mudah diuji dan divalidasi.

II. GUIDED

Soal Studi Case

1.

SOURCECODE

```
package main

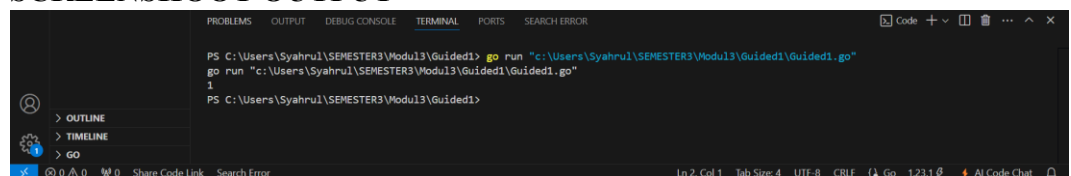
import "fmt"

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



```
PS C:\Users\Syahrul\SEMESTER3\Modul3\Guided1> go run "c:\Users\Syahrul\SEMESTER3\Modul3\Guided1\Guided1.go"
go run "c:\Users\Syahrul\SEMESTER3\Modul3\Guided1\Guided1.go"
1
PS C:\Users\Syahrul\SEMESTER3\Modul3\Guided1>
```

DESKRIPSI PROGRAM

Program di atas adalah implementasi sederhana untuk menghitung permutasi dari dua bilangan yang dimasukkan oleh pengguna. Permutasi adalah cara untuk menghitung banyaknya urutan berbeda yang bisa dibuat dari sekumpulan objek. Program dimulai dengan mendeklarasikan dua variabel a dan b yang bertipe integer, dan kemudian meminta pengguna untuk memasukkan nilai-nilai ini melalui fungsi `fmt.Scan`. Selanjutnya, program memeriksa apakah nilai a lebih besar atau sama dengan b. Jika iya, fungsi `permutasi(a, b)` dipanggil; jika tidak, `permutasi(b, a)` yang dipanggil. Fungsi `faktorial(n int)` digunakan untuk menghitung faktorial dari sebuah bilangan n. Faktorial dihitung dengan mengalikan semua bilangan dari 1

hingga n. Fungsi permutasi(n, r) kemudian memanfaatkan hasil dari fungsi faktorial untuk menghitung nilai permutasi menggunakan rumus $n! / (n-r)!$. Hasil dari perhitungan ini kemudian ditampilkan ke layar menggunakan `fmt.Println`.

2.

SOURCECODE

```
package main

import "fmt"

func hitungLuas(sisi float64) float64 {
    return sisi * sisi
}

func hitungKeliling(sisi float64) float64 {
    return 4 * sisi
}

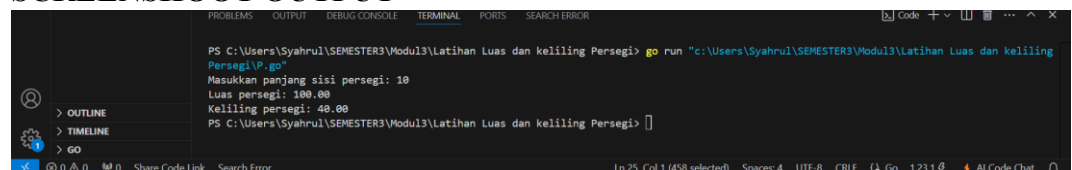
func main() {
    var sisi float64

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

    luas := hitungLuas(sisi)
    keliling := hitungKeliling(sisi)

    fmt.Printf("Luas persegi: %.2f\n", luas)
    fmt.Printf("Keliling persegi: %.2f\n", keliling)
}
```

SCREENSHOOT OUTPUT



```
PS C:\Users\Syahrul\SEMESTER3\Modul3\Latihan Luas dan keliling Persegi> go run "c:\Users\Syahrul\SEMESTER3\Modul3\Latihan Luas dan keliling Persegi\p.go"
Masukkan panjang sisi persegi: 10
Luas persegi: 100.00
Keliling persegi: 40.00
PS C:\Users\Syahrul\SEMESTER3\Modul3\Latihan Luas dan keliling Persegi> []
```

DESKRIPSI PROGRAM

Program di atas bertujuan untuk menghitung luas dan keliling dari sebuah persegi berdasarkan panjang sisi yang dimasukkan oleh pengguna. Pada awalnya, program meminta pengguna untuk memasukkan nilai panjang sisi persegi melalui input keyboard. Setelah itu, terdapat dua fungsi, yaitu `hitungLuas` dan `hitungKeliling`, yang masing-masing berfungsi untuk

menghitung luas dan keliling persegi. Fungsi `hitungLuas` menerima satu parameter berupa panjang sisi dan mengembalikan hasil perkalian sisi dengan dirinya sendiri ($sisi^2$). Sementara itu, fungsi `hitungKeliling` menghitung keliling dengan mengalikan panjang sisi dengan empat. Nilai hasil perhitungan luas dan keliling ini kemudian ditampilkan dalam format dua angka di belakang koma menggunakan fungsi `fmt.Printf`.

III. UNGUIDED

Soal Studi Case

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 $a < b$ dan $b \geq d$. 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! $P(n,r) = n! / (n-r)!$, sedangkan $C(n,r) = n! / r!(n-r)!$

Sourcecode

```
package main

import (
    "fmt"
)

func factorial(n int) int {
    if n == 0 {
        return 1
    }
    fact := 1
    for i := 1; i <= n; i++ {
        fact *= i
    }
    return fact
}

func permutation(n, r int) int {
    if n >= r {
        return factorial(n) / factorial(n-r)
    }
    return 0
}

func combination(n, r int) int {
    if n >= r {
        return factorial(n) / (factorial(r) * factorial(n-r))
    }
    return 0
}

func main() {
```

```

var a, b, c, d int

fmt.Print("Masukkan bilangan: ")
fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

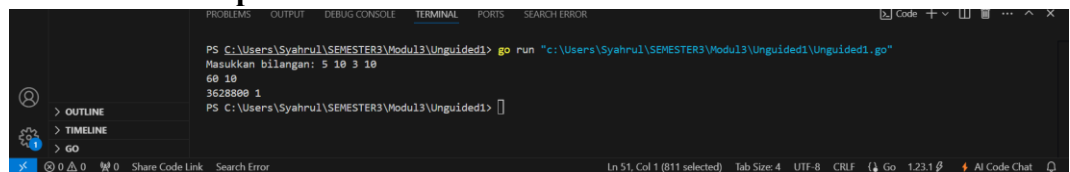
if a >= c && b >= d {
    Pac := permutation(a, c)
    Cac := combination(a, c)

    Pbd := permutation(b, d)
    Cbd := combination(b, d)

    fmt.Printf("%d %d\n", Pac, Cac)
    fmt.Printf("%d %d\n", Pbd, Cbd)
} else {
    fmt.Println("Syarat  $a \geq c$  dan  $b \geq d$  tidak terpenuhi.")
}
}

```

Screenshoot Output



```

PS C:\Users\Syahrul\SEMESTER3\Modul3\Unguided1> go run "c:\Users\Syahrul\SEMESTER3\Modul3\Unguided1\Unguided1.go"
Masukkan bilangan: 5 10 3 10
60 10
3628800 1
PS C:\Users\Syahrul\SEMESTER3\Modul3\Unguided1>

```

Deskripsi Program

Program di atas adalah implementasi untuk menghitung **permutasi** dan **kombinasi** dari empat bilangan yang dimasukkan pengguna, yaitu a, b, c, dan d. Program dimulai dengan meminta pengguna untuk memasukkan empat bilangan tersebut. Kemudian, terdapat fungsi factorial yang digunakan untuk menghitung faktorial dari bilangan yang diperlukan dalam perhitungan permutasi dan kombinasi. Fungsi permutation digunakan untuk menghitung permutasi $P(n,r)$ dengan rumus $P(n,r) = \frac{n!}{(n-r)!}$, sedangkan fungsi combination digunakan untuk menghitung kombinasi $C(n,r)$ dengan rumus $C(n,r) = \frac{n!}{r!(n-r)!}$. Program mengecek apakah syarat $a \geq c$ dan $b \geq d$ terpenuhi sebelum melakukan perhitungan. Jika syarat terpenuhi, program akan mencetak hasil permutasi dan kombinasi dari a terhadap c dan b terhadap d. Jika tidak, program akan mencetak pesan bahwa syarat tidak terpenuhi.

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 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)))
}

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

    fmt.Print("Masukkan bilangan: ")
    fmt.Scanf("%d %d %d", &a, &b, &c)

    fmt.Println(fogoh(a))
    fmt.Println(gohof(b))
    fmt.Println(hofog(c))
}
```



```
}

```

Sceenshoot Output

```

PS C:\Users\Syahrul\SEMESTER3\Modul3\Unguided2> go run "c:\Users\Syahrul\SEMESTER3\Modul3\Unguided2\Unguided2.go"
Masukkan bilangan: 7 2 10
36
3
65
PS C:\Users\Syahrul\SEMESTER3\Modul3\Unguided2>

```

Deskripsi Program

Program di atas adalah implementasi untuk menghitung komposisi tiga fungsi matematika, yaitu $f(x)=x^2$, $g(x)=x-2$, dan $h(x)=x+1$. Program ini memiliki tiga fungsi utama untuk menghitung komposisi fungsi-fungsi tersebut: $f(g(h(x)))$ untuk menghitung $f(g(h(x)))f(g(h(x)))f(g(h(x)))$, $g(h(f(x)))$ untuk menghitung $g(h(f(x)))g(h(f(x)))g(h(f(x)))$, dan $h(f(g(x)))$ untuk menghitung $h(f(g(x)))h(f(g(x)))h(f(g(x)))$. Pada bagian utama program (main), pengguna diminta untuk memasukkan tiga bilangan bulat a, b, dan c yang akan digunakan sebagai input untuk masing-masing komposisi fungsi. Program kemudian menghitung dan menampilkan hasil dari $f(g(h(a)))f(g(h(a)))f(g(h(a)))$, $g(h(f(b)))g(h(f(b)))g(h(f(b)))$, dan $h(f(g(c)))h(f(g(c)))h(f(g(c)))$ secara berurutan. Program ini berguna untuk memvisualisasikan bagaimana fungsi-fungsi komposisi bekerja dalam matematika diskrit.

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

import (
    "fmt"
)

func main() {
    var cx1, cy1, r1 int
    var cx2, cy2, r2 int
    var x, y int

    fmt.Print("Masukkan pusat lingkaran 1 (cx cy) dan radius (r): ")
    fmt.Scan(&cx1, &cy1, &r1)

    fmt.Print("Masukkan pusat lingkaran 2 (cx cy) dan radius (r): ")
    fmt.Scan(&cx2, &cy2, &r2)

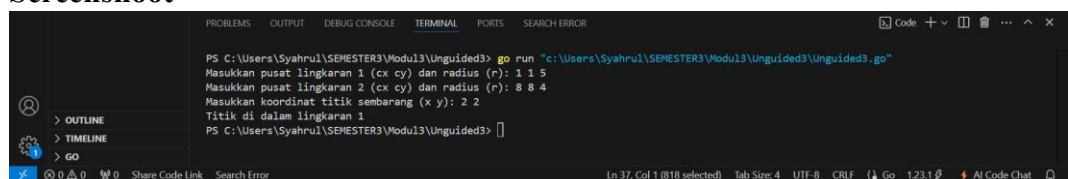
    fmt.Print("Masukkan koordinat titik sembarang (x y): ")
    fmt.Scan(&x, &y)

    d1 := (x-cx1)*(x-cx1) + (y-cy1)*(y-cy1)
    d2 := (x-cx2)*(x-cx2) + (y-cy2)*(y-cy2)

    inCircle1 := d1 < r1*r1
    inCircle2 := d2 < r2*r2

    if inCircle1 && inCircle2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if inCircle1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if inCircle2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

Screenshoot



The screenshot shows a terminal window with the following output:

```
PS C:\Users\Syahrul\SEMESTER3\Modul3\Unguided3> go run "c:\Users\Syahrul\SEMESTER3\Modul3\Unguided3\Unguided3.go"
Masukkan pusat lingkaran 1 (cx cy) dan radius (r): 1 1 5
Masukkan pusat lingkaran 2 (cx cy) dan radius (r): 8 8 4
Masukkan koordinat titik sembarang (x y): 2 2
Titik di dalam lingkaran 1
PS C:\Users\Syahrul\SEMESTER3\Modul3\Unguided3>
```

The terminal window also shows the file path: `PS C:\Users\Syahrul\SEMESTER3\Modul3\Unguided3>`. The status bar at the bottom indicates: `Ln 37, Col 1 (818 selected) Tab Size: 4 UTF-8 CRLF 1.23.1 AI Code Chat`.

Deskripsi Program

Program di atas merupakan implementasi dalam bahasa Go yang digunakan untuk menentukan posisi suatu titik relatif terhadap dua lingkaran. Program dimulai dengan mendeklarasikan variabel yang diperlukan, termasuk pusat dan radius dari dua lingkaran, serta koordinat titik yang akan diuji. Pengguna diminta untuk memasukkan pusat dan radius kedua lingkaran, diikuti oleh koordinat titik sembarang. Setelah itu, program menghitung jarak kuadrat dari titik tersebut ke masing-masing pusat lingkaran menggunakan rumus jarak Euclidean. Program kemudian membandingkan jarak tersebut dengan kuadrat radius untuk menentukan apakah titik berada di dalam, di luar, atau tepat di tepi lingkaran. Hasilnya akan ditampilkan dalam bentuk pesan yang menyatakan posisi titik tersebut, apakah berada di dalam lingkaran 1, lingkaran 2, di dalam kedua lingkaran, atau di luar keduanya. Dengan demikian, program ini memberikan cara yang interaktif dan langsung untuk mengevaluasi posisi titik relatif terhadap lingkaran yang diberikan.

KESIMPULAN

Fungsi di Golang merupakan blok kode yang dirancang untuk melakukan tugas tertentu dan dapat dipanggil berulang kali di berbagai bagian program. Penggunaan fungsi membantu dalam menyusun program dengan lebih terstruktur, meningkatkan keterbacaan, dan memudahkan pemeliharaan kode. Fungsi dapat menerima parameter untuk memproses data yang dikirimkan dan dapat mengembalikan nilai setelah eksekusi, yang membuatnya sangat fleksibel. Dalam Golang, fungsi dapat didefinisikan dengan tipe data parameter yang jelas, memungkinkan penanganan berbagai tipe data dengan cara yang aman dan efisien. Selain itu, Golang juga mendukung fungsi sebagai tipe data pertama (first-class citizens), yang berarti fungsi dapat disimpan dalam variabel, dipassing sebagai argumen ke fungsi lain, atau bahkan mengembalikan fungsi dari fungsi lain. Hal ini mendukung pengembangan paradigma pemrograman fungsional dan menjadikan Golang alat yang kuat untuk mengembangkan aplikasi yang kompleks dengan cara yang lebih modular dan terorganisir. Dengan menggunakan fungsi secara efektif, pengembang dapat meningkatkan produktivitas dan kualitas kode, menjadikannya lebih mudah untuk dikembangkan dan diuji.

DAFTAR PUSTAKA

<https://blog.myskill.id/istilah-dan-tutorial/golang-pengertian-fungsi-dan-keunggulannya/>