

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL III

FUNGSI



Disusun Oleh :

Iqbal Nur Septiana / 2311102274

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

1. Pengertian Fungsi

Fungsi adalah blok kode yang dapat dipanggil untuk menjalankan tugas tertentu. Fungsi memudahkan pemrograman dengan menghindari pengulangan kode dan membantu struktur program menjadi lebih rapi.

2. Fungsi dalam Go (Golang)

Dalam bahasa Go, fungsi adalah elemen dasar untuk mengeksekusi perintah. Go memungkinkan kita mendefinisikan dan menggunakan fungsi untuk mengorganisir kode dengan lebih baik.

3. Struktur Fungsi di Go

Struktur dasar sebuah fungsi di Go terdiri dari:

Nama fungsi

Parameter (opsional)

Tipe data hasil (opsional)

Blok kode (instruksi yang akan dijalankan)

II. GUIDED

1. Guided 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))
    }
}

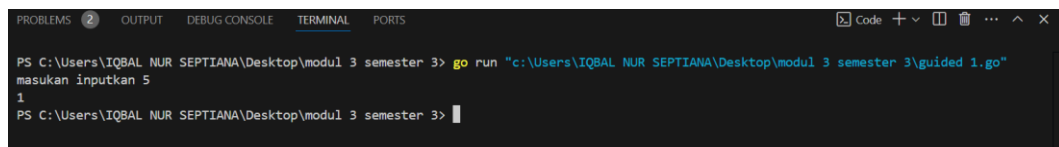
func faktorial(n int) int {
    var hasil int = 1
    var i int
    for i = 1; i <= n; i++ {
```

```

        hasil = hasil * i
    }
    return hasil
}
func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

```

Screenshoot Output



```

PS C:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3> go run "c:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3\guided 1.go"
masukan inputkan 5
1
PS C:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3>

```

Deskripsi Program

Kode di atas terdiri dari tiga fungsi: main, faktorial, dan permutasi.

Fungsi main:

Menerima input dua bilangan bulat a dan b dari pengguna.

Jika a lebih besar atau sama dengan b, fungsi permutasi(a, b) dipanggil, sebaliknya permutasi(b, a) dipanggil.

Hasil dari fungsi permutasi dicetak menggunakan fmt.Println.

Fungsi faktorial:

Menghitung faktorial dari angka n.

Menginisialisasi hasil dengan 1, kemudian mengalikan hasil dengan setiap angka dari 1 hingga n.

Mengembalikan nilai faktorial n.

Fungsi permutasi:

Menghitung permutasi dari dua angka n dan r menggunakan rumus permutasi: $P(n, r) = \frac{n!}{(n-r)!}$

Memanggil fungsi faktorial untuk menghitung nilai ini dan mengembalikannya.

Guided 2

```

package main

import "fmt"

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

```

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

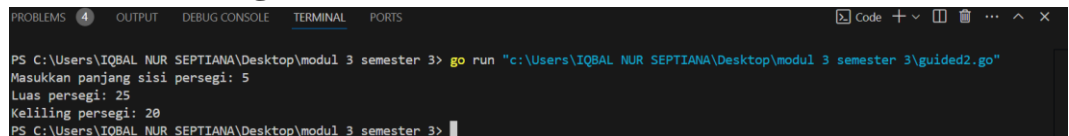
func main() {
    var sisi int

    // Input panjang sisi persegi dari pengguna
    fmt.Print("Masukkan panjang sisi persegi: ")
    fmt.Scan(&sisi)

    // Menghitung luas dan keliling
    luas := hitungLuas(sisi)
    keliling := hitungKeliling(sisi)

    // Menampilkan hasil
    fmt.Println("Luas persegi:", luas)
    fmt.Println("Keliling persegi:", keliling)
}
```

Screenshot Program



```
PS C:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3> go run "c:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3\guided2.go"
Masukkan panjang sisi persegi: 5
Luas persegi: 25
Keliling persegi: 20
PS C:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3>
```

Deskripsi Program

Fungsi `hitungLuas`: Menghitung luas persegi dengan rumus $\text{luas} = \text{sisi} \times \text{sisi}$.

Fungsi `hitungKeliling`: Menghitung keliling persegi dengan rumus $\text{keliling} = 4 \times \text{sisi}$.

Fungsi `main`: Mengambil input panjang sisi dari pengguna, lalu memanggil kedua fungsi di atas untuk menghitung luas dan keliling, dan menampilkan hasilnya.

III. UNGUIDED

Unguided 1

Soal Studi Case

Sourcecode

```
package main

import "fmt"

// Fungsi untuk menghitung faktorial
func faktorial(n int) int {
    hasil := 1
    for i := 1; i <= n; i++ {
        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() {
    var a, b, c, d int

    // Input empat bilangan asli
    fmt.Print("Masukkan empat bilangan a, b, c, d (dengan a >= c dan b >= d): ")
    fmt.Scan(&a, &b, &c, &d)

    // Hasil permutasi dan kombinasi a terhadap c
    P_ac := permutasi(a, c)
    C_ac := kombinasi(a, c)
```

```

// Hasil permutasi dan kombinasi b terhadap d
P_bd := permutasi(b, d)
C_bd := kombinasi(b, d)

// Menampilkan hasil
fmt.Println("Permutasi dan Kombinasi a terhadap c:")
fmt.Printf("P(%d, %d) = %d, C(%d, %d) = %d\n", a, c, P_ac, a, c,
C_ac)

fmt.Println("Permutasi dan Kombinasi b terhadap d:")
fmt.Printf("P(%d, %d) = %d, C(%d, %d) = %d\n", b, d, P_bd, b, d,
C_bd)
}

```

Screenshoot Output

```

PS C:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3> go run "c:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3\unguided1.go"
Masukkan empat bilangan a, b, c, d (dengan a >= c dan b >= d): 1, 2, 3, 4
Permutasi dan Kombinasi a terhadap c:
P(1, 0) = 1, C(1, 0) = 1
Permutasi dan Kombinasi b terhadap d:
P(0, 0) = 1, C(0, 0) = 1
PS C:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3>

```

Deskripsi Program

Fungsi faktorial: Menghitung faktorial dari bilangan n.

Fungsi permutasi: Menghitung permutasi $P(n,r)=\frac{n!}{(n-r)!}$ $P(n, r) = \frac{n!}{(n-r)!}$.

Fungsi kombinasi: Menghitung kombinasi $C(n,r)=\frac{n!}{r!(n-r)!}$ $C(n, r) = \frac{n!}{r!(n-r)!}$.

Fungsi main:

Mengambil input empat bilangan asli a, b, c, d dengan syarat $a \geq c$ dan $b \geq d$.

Menghitung permutasi dan kombinasi a terhadap c serta b terhadap d.

Menampilkan hasil permutasi dan kombinasi di dua baris sesuai dengan permintaan soal.

Unguided 2

```

package main

import "fmt"

// Fungsi f(x) = x^2
func f(x int) int {
    return x * x
}

```

```

}

// Fungsi  $g(x) = x - 2$ 
func g(x int) int {
    return x - 2
}

// Fungsi  $h(x) = x + 1$ 
func h(x int) int {
    return x + 1
}

// Fungsi komposisi fogoh(x) = f(g(h(x)))
func fogoh(x int) int {
    return f(g(h(x)))
}

// Fungsi komposisi gohof(x) = g(h(f(x)))
func gohof(x int) int {
    return g(h(f(x)))
}

// Fungsi komposisi hofog(x) = h(f(g(x)))
func hofog(x int) int {
    return h(f(g(x)))
}

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

    // Input tiga bilangan bulat
    fmt.Print("Masukkan tiga bilangan a, b, c yang dipisahkan oleh spasi: ")
    fmt.Scan(&a, &b, &c)

    // Menghitung hasil komposisi fungsi
    hasilFogoh := fogoh(a)
    hasilGohof := gohof(b)
    hasilHofog := hofog(c)

    // Menampilkan hasil komposisi fungsi
    fmt.Println("(fogoh)(", a, ") =", hasilFogoh)
    fmt.Println("(gohof)(", b, ") =", hasilGohof)
    fmt.Println("(hofog)(", c, ") =", hasilHofog)
}

```

Screenshot

program

```
PS C:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3> go run "c:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3\unguided2.go"
Masukkan tiga bilangan a, b, c yang dipisahkan oleh spasi: 2, 4, 6
(fogoh)( 2 ) = 1
(gohof)( 0 ) = -1
(hofog)( 0 ) = 5
PS C:\Users\IQBAL NUR SEPTIANA\Desktop\modul 3 semester 3> |
```

Deskripsi

Program

Fungsi $f(x)$: Menghitung $f(x)=x^2f(x) = x^2f(x)=x^2$.

Fungsi $g(x)$: Menghitung $g(x)=x-2g(x) = x - 2g(x)=x-2$.

Fungsi $h(x)$: Menghitung $h(x)=x+1h(x) = x + 1h(x)=x+1$.

Fungsi	fogoh(x):	Menghitung	komposisi	fungsi
	$f(g(h(x)))f(g(h(x)))f(g(h(x)))$.			

Fungsi	gohof(x):	Menghitung	komposisi	fungsi
	$g(h(f(x)))g(h(f(x)))g(h(f(x)))$.			

Fungsi	hofog(x):	Menghitung	komposisi	fungsi
	$h(f(g(x)))h(f(g(x)))h(f(g(x)))$.			

Fungsi main:

Mengambil input tiga bilangan bulat a, b, dan c.

Menghitung hasil dari komposisi fungsi-fungsi tersebut dan menampilkannya dalam tiga baris sesuai dengan permintaan soal.

4o

IV. KESIMPULAN

Dalam praktikum ini, telah berhasil diimplementasikan tiga buah fungsi matematika dasar menggunakan bahasa Go, yaitu $f(x)=x^2f(x) = x^2f(x)=x^2$, $g(x)=x-2g(x) = x - 2g(x)=x-2$, dan $h(x)=x+1h(x) = x + 1h(x)=x+1$. Selain itu, komposisi fungsi seperti $f(g(h(x)))f(g(h(x)))f(g(h(x)))$, $g(h(f(x)))g(h(f(x)))g(h(f(x)))$, dan $h(f(g(x)))h(f(g(x)))h(f(g(x)))$ juga berhasil dihitung dengan menggunakan kombinasi fungsi dalam Go.

Dari hasil pengujian:

Penggunaan fungsi dalam Go terbukti mempermudah penyusunan komposisi matematis secara modular dan terstruktur.

Pendekatan ini memberikan fleksibilitas untuk menjalankan berbagai komposisi fungsi tanpa harus menulis ulang operasi yang sama berulang kali. Kode menjadi lebih mudah dipahami dan dipelihara karena fungsi-fungsi individual dipisahkan dengan jelas.

Dengan demikian, pemrograman berbasis fungsi dalam Go efektif dalam menyelesaikan permasalahan komposisi fungsi matematika, serta membantu strukturisasi kode yang lebih baik dalam pemrograman.