

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III
FUNGSI**



Disusun Oleh :

Wafiq Nur Azizah / 2311102270

S1IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dalam pemrograman, fungsi merupakan sekumpulan blok kode yang diberi nama tertentu. Penggunaan fungsi dengan baik dapat membuat kode menjadi lebih modular dan mengikuti prinsip DRY (Don't Repeat Yourself), yang berarti kita tidak perlu menulis kode yang sama berulang kali. Cukup dengan mendeklarasikan blok kode sebagai fungsi satu kali, kita dapat memanggilnya sesuai kebutuhan.

A. Penerapan Fungsi

Pengguna mungkin menyadari bahwa mereka sebenarnya telah mengimplementasikan fungsi dalam banyak praktik sebelumnya, yaitu fungsi `main()`. Fungsi `main()` merupakan fungsi utama dalam program Go yang akan dijalankan saat program diaktifkan.

Selain fungsi `main()`, kita juga dapat membuat fungsi lainnya dengan mudah. Caranya adalah dengan menuliskan kata kunci `func`, diikuti dengan nama fungsi, kemudian tanda kurung `()` (yang bisa diisi dengan parameter), dan diakhiri dengan tanda kurung kurawal yang membungkus blok kode. Parameter adalah variabel yang terikat pada fungsi dan nilainya ditentukan saat fungsi tersebut dipanggil. Parameter bersifat opsional; suatu fungsi dapat memiliki nol, satu, atau lebih parameter sesuai kebutuhan. Data yang digunakan sebagai nilai parameter saat pemanggilan fungsi disebut sebagai argumen parameter.

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    // Initialize the slice of names
    var names = []string{"John", "Wick"}
    // Call the printMessage function with a message and
    the names slice
    printMessage("halo", names)
}

// Function to print a message concatenated with names
func printMessage(message string, arr []string) {
    // Join the names with a space
    var nameString = strings.Join(arr, " ")
    // Print the message followed by the joined names
    fmt.Println(message, nameString)
}
```

```
PS C:\Users\HP> go run "d:\Modul 3\coba.go"
halo John Wick
PS C:\Users\HP>
```

Pada kode di atas, sebuah fungsi baru dibuat dengan nama 'printMessage()' memiliki 2 buah parameter yaitu string message dan slice string 'arr'.

Fungsi tersebut dipanggil dalam 'main()', dalam pemanggilannya disisipkan dua buah argument parameter.

- Argument parameter pertama adalah string 'halo' yang ditampung parameter message.
- Argument parameter ke-2 adalah 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()' (berada di dalam package strings).

B. Fungsi Dengan Return Value / Nilai Balik

Selain parameter, fungsi bisa memiliki attribute return value atau nilai balik. Fungsi yang memiliki return value, saat deklarasinya harus ditentukan terlebih dahulu tipe data dari nilai baliknya. Fungsi yang tidak mengembalikan nilai apapun (contohnya seperti fungsi 'main()' dan 'printMessage()') biasa disebut dengan void function. Program berikut merupakan contoh penerapan fungsi yang memiliki 'return' value.

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()'. Sedangkan cara untuk mengembalikan nilai itu sendiri adalah dengan menggunakan keyword return diikuti data yang 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 hal yang belum pernah kita lakukan pada pembahasan-pembahasan sebelumnya, kita akan bahas satu-persatu.

II. GUIDED

1. Soal Studi Case 1

Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilai faktorial dan permutasi.

Masukan terdiri dari dua buah bilangan positif a dan b.

Keluaran berupa sebuah bilangan bulat yang menyatakan nilai a permutasi b apabila $a \geq b$ atau b permutasi a untuk kemungkinan yang lain.

Sourcecode

```
package main

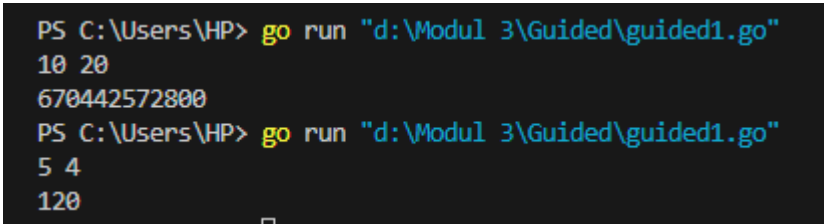
import "fmt"

func main() {
    var a, b int
    fmt.Scan(&a, &b)
    if a >= b {
        fmt.Println(permutasi(a, b)) // Mengganti
        PrintIn dengan Println
    } else {
        fmt.Println(permutasi(b, a)) // Mengganti
        PrintIn dengan Println
    }
}

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)
}
```

Output Program



```
PS C:\Users\HP> go run "d:\Modul 3\Guided\guided1.go"
10 20
670442572800
PS C:\Users\HP> go run "d:\Modul 3\Guided\guided1.go"
5 4
120
```

Deskripsi Program

Program Go di atas meminta dua angka sebagai input dari pengguna dan menghitung permutasi berdasarkan kedua angka tersebut. Permutasi adalah proses di mana urutan elemen sangat diperhitungkan, dan program ini menggunakan rumus permutasi standar. Jika angka pertama (a) lebih besar atau sama dengan angka kedua (b), maka program menghitung $P(a,b)$, tetapi jika sebaliknya permutasi akan dihitung sebagai $P(b,a)$.

Program ini memiliki dua fungsi utama, yaitu 'faktorial' dan 'permutasi'. Fungsi 'faktorial' bertugas menghitung faktorial suatu angka, yaitu hasil perkalian angka tersebut dengan semua angka di bawahnya hingga 1. Fungsi 'permutasi' akan menggunakan nilai faktorial untuk menghitung permutasi dengan membagi faktorial dari n dengan faktorial dari (n-r). Hasil perhitungan ini kemudian dicetak menggunakan 'fmt.Println'.

2. Soal Studi Case 2

Sourcecode

```
package main

import "fmt"

func main() {
    var sisi int
    sisi = inputSisi()

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

    tampilkanHasil(luas, keliling)
}

func inputSisi() int {
    var s int
    fmt.Print("Panjang sisi persegi: ")
    fmt.Scan(&s)
    return s
}

func hitungLuas(s int) int {
    return s * s
}

func hitungKeliling(s int) int {
```

```
    return 4 * s
}

func tampilkanHasil(luas, keliling int) {
    fmt.Println("Hasil luas persegi:", luas)
    fmt.Println("Hasil keliling persegi:", keliling)
}
```

Output Program

```
PS C:\Users\HP> go run "d:\Modul 3\Guided\guided2.go"
Panjang sisi persegi: 19
Hasil luas persegi: 361
Hasil keliling persegi: 76
```

Deskripsi Program

Program Go di atas merupakan sebuah program sederhana yang menghitung luas dan keliling persegi. Pertama, program meminta input panjang sisi persegi dari pengguna melalui fungsi `inputSisi()`, yang kemudian digunakan untuk menghitung luas dan keliling persegi. Fungsi `hitungLuas(s int)` mengalikan panjang sisi dengan dirinya sendiri untuk mendapatkan luas, sedangkan fungsi `hitungKeliling(s int)` mengalikan panjang sisi dengan 4 untuk menghitung keliling.

Setelah perhitungan selesai, hasil luas dan keliling ditampilkan kepada pengguna melalui fungsi `tampilkanHasil(luas, keliling int)`. Komponen penting dari program ini adalah pengambilan input (panjang sisi), perhitungan luas dan keliling, serta menampilkan hasilnya. Program ini cukup mudah dipahami karena mengikuti alur yang jelas, mulai dari input, perhitungan, hingga output.

III. UNGUIDED

1. 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 $a \leq c$ dan $b \leq d$.

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!

$$P(n, r) = \frac{n!}{(n-r)!}, \text{ sedangkan } C(n, r) = \frac{n!}{r!(n-r)!}$$

Contoh:

No	Masukan	Keluaran	Penjelasan
1	5 10 3 10	60 10 3628800 1	$P(5, 3) = 5! / 2! = 120 / 2 = 60$ $C(5, 3) = 5! / (3! \times 2!) = 120 / 12 = 10$ $P(10, 10) = 10! / 0! = 3628800 / 1 = 3628800$ $C(10, 10) = 10! / (10! \times 0!) = 10! / 10! = 1$
2	8 0 2 0	56 28 1 1	

Selesaikan program tersebut dengan memanfaatkan subprogram yang diberikan berikut ini!

```
function factorial(n: integer) → integer
{mengembalikan nilai faktorial dari n}

function permutation(n, r : integer) → integer
{Mengembalikan hasil n permutasi r, dan n >= r}

function combination(n, r : integer) → integer
{Mengembalikan hasil n kombinasi r, dan n >= r}
```

Sourcecode

```
package main

import "fmt"

func main() {
    var a, b, c, d int
    fmt.Print("Input nilai a, b, c, d : ")
    fmt.Scan(&a, &b, &c, &d)

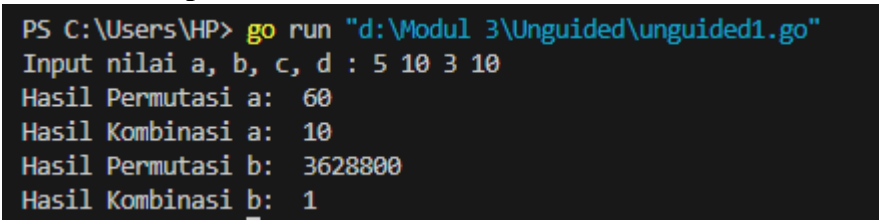
    if a >= c && b >= d {
        fmt.Println("Hasil Permutasi a: ",
hitungPermutasi(a, c))
        fmt.Println("Hasil Kombinasi a: ",
hitungKombinasi(a, c))
        fmt.Println("Hasil Permutasi b: ",
hitungPermutasi(b, d))
        fmt.Println("Hasil Kombinasi b: ",
hitungKombinasi(b, d))
    } else {
        fmt.Print("Kondisi 'a >= c && b >= d' tidak
terpenuhi: ", a, b, c, d)
    }
}

func hitungFaktorial(n int) int {
    var hasil int = 1
    for i := 1; i <= n; i++ {
        hasil *= i
    }
    return hasil
}

func hitungPermutasi(n, r int) int {
    return hitungFaktorial(n) / hitungFaktorial(n-r)
}

func hitungKombinasi(n, r int) int {
    return hitungFaktorial(n) / (hitungFaktorial(r) *
hitungFaktorial(n-r))
}
```

Screenshoot Output



```
PS C:\Users\HP> go run "d:\Modul 3\Unguided\unguided1.go"
Input nilai a, b, c, d : 5 10 3 10
Hasil Permutasi a: 60
Hasil Kombinasi a: 10
Hasil Permutasi b: 3628800
Hasil Kombinasi b: 1
```


Deskripsi Program

Program Go di atas untuk menghitung permutasi dan kombinasi dari dua pasang bilangan yang diberikan oleh pengguna. Pertama, program meminta pengguna untuk memasukkan empat bilangan bulat positif, yaitu a, b, c, dan d. Setelah menerima input, program memeriksa apakah kondisi $a \geq c$ dan $b \geq d$ terpenuhi. Jika kondisi ini terpenuhi, program akan menghitung dan menampilkan hasil permutasi dan kombinasi untuk pasangan (a, c) dan (b, d). Jika tidak, program akan memberikan pesan bahwa kondisi tidak terpenuhi beserta nilai-nilai yang dimasukkan.

Untuk melakukan perhitungan, program memiliki tiga fungsi utama, yaitu 'hitungFaktorial', 'hitungPermutasi', dan 'hitungKombinasi'. Fungsi 'hitungFaktorial' menghitung faktorial dari sebuah angka yang merupakan dasar untuk perhitungan permutasi dan kombinasi. Fungsi 'hitungPermutasi' menggunakan rumus permutasi biasanya dan fungsi 'hitungKombinasi' menggunakan rumus kombinasi pada umumnya. Program ini memudahkan pengguna untuk memahami konsep dasar permutasi dan kombinasi dalam matematika.

2. Soal Studi Case 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 $(h \circ f \circ g)(c)$!

Contoh

No	Masukan	Keluaran	Penjelasan
1	7 2 10	36 3 65	(fogog)(7) = 36 (gohof)(2) = 3 (hofog)(10) = 65
2	5 5 5	16 24 10	(fogog)(5) = 16 (gohof)(5) = 24 (hofog)(5) = 10
3	3 8 4	4 63 5	(fogog)(5) = 4 (gohof)(5) = 63 (hofog)(5) = 5

Sourcecode

```
package main

import "fmt"

func main() {
    var a, b, c int
    fmt.Print("Input a, b, c : ")
    fmt.Scanln(&a, &b, &c)

    hasil1 := komposisiFogoh(a)
    hasil2 := komposisiGohof(b)
    hasil3 := komposisiHofog(c)
    fmt.Printf("(fogoh) (%d) = %d\n", a, hasil1)
    fmt.Printf("(gohof) (%d) = %d\n", b, hasil2)
    fmt.Printf("(hofog) (%d) = %d\n", c, hasil3)
}

// Definisi fungsi matematika
// f(x) := x * x
// g(x) := x - 2
// h(x) := x + 1

func komposisiFogoh(x int) int {
    // f(g(h(x)))
    hx := h(x)
    ghx := g(hx)
    result := f(ghx)

    return result
}

func komposisiGohof(x int) int {
    // g(h(f(x)))
```

```

        fx := f(x)
        hfx := h(fx)
        result := g(hfx)

        return result
    }

    func komposisiHofog(x int) int {
        // h(f(g(x)))
        gx := g(x)
        fgx := f(gx)
        result := h(fgx)

        return result
    }

    // Definisi fungsi f, g, dan h
    func f(x int) int {
        return x * x
    }

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

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

```

Screenshoot Output

```

PS C:\Users\HP> go run "d:\Modul 3\Unguided\unguided2.go"
Input a, b, c : 7 2 10
(fogoh)(7) = 36
(gohof)(2) = 3
(hofog)(10) = 65
PS C:\Users\HP> go run "d:\Modul 3\Unguided\unguided2.go"
Input a, b, c : 5 5 5
(fogoh)(5) = 16
(gohof)(5) = 24
(hofog)(5) = 10
PS C:\Users\HP> go run "d:\Modul 3\Unguided\unguided2.go"
Input a, b, c : 3 8 4
(fogoh)(3) = 4
(gohof)(8) = 63
(hofog)(4) = 5

```

Deskripsi Program

Program Go di atas untuk menghitung komposisi dari tiga fungsi matematika, yaitu f , g , dan h berdasarkan input yang diberikan oleh pengguna. Fungsi $f(x)$ mengalikan input dengan dirinya sendiri (kuadrat), fungsi $g(x)$ mengurangi input dengan 2, dan fungsi $h(x)$ menambahkan 1 pada input. Program akan meminta pengguna untuk memasukkan tiga nilai a , b , dan c yang kemudian dihitung tiga komposisi dari fungsi-fungsi ini, yaitu $f(g(h(a)))$, $g(h(f(b)))$, dan $h(f(g(c)))$.

Setelah melakukan perhitungan, program menampilkan hasil masing-masing komposisi ke layar dengan format yang jelas. Bagian penting dari program ini adalah definisi fungsi matematika dan penerapan komposisi fungsi melalui fungsi 'komposisiFogoh', 'komposisiGohof', dan 'komposisiHofog'. Dengan cara ini, pengguna dapat melihat bagaimana hasil dari fungsi-fungsi tersebut berinteraksi satu sama lain sehingga dapat memberikan pemahaman yang lebih baik tentang konsep komposisi dalam matematika.

3. Soal Studi Case 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".

Contoh

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3	Titik di dalam lingkaran 2

	4 5 6 7 8	
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

Fungsi untuk menghitung Jarak titik (a, b) dan (c, d) dimana rumus jarak adalah:

$$jarak = \sqrt{(a - c)^2 + (b - d)^2}$$

dan juga fungsi untuk menentukan posisi sebuah titik sembarang berada di dalam suatu lingkaran atau tidak.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, x, y float64

    fmt.Println("Lingkaran 1")
    fmt.Print("(cx) (cy) (r) : ")
    fmt.Scanln(&cx1, &cy1, &r1)

    fmt.Println("Lingkaran 2")
    fmt.Print("(cx) (cy) (r) : ")
    fmt.Scanln(&cx2, &cy2, &r2)

    fmt.Print("Input koordinat sembarang (x) (y) : ")
    fmt.Scanln(&x, &y)

    // Cek posisi terhadap Lingkaran 1 dan Lingkaran 2
    isInsideCircle1 := isPointInCircle(cx1, cy1, x, y,
r1)
    isInsideCircle2 := isPointInCircle(cx2, cy2, x, y,
r2)

    if isInsideCircle1 && isInsideCircle2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    }
}
```

```

    } else if isInsideCircle1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if isInsideCircle2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

func calculateDistance(a, b, c, d float64) float64 {
    // Menghitung jarak antara dua titik
    dx := (a - c) * (a - c)
    dy := (b - d) * (b - d)
    return math.Sqrt(dx + dy)
}

func isPointInCircle(cx, cy, x, y, r float64) bool {
    // Mengecek apakah titik (x, y) berada dalam
    lingkaran
    return calculateDistance(cx, cy, x, y) <= r
}

```

Screenshoot Output

```

PS C:\Users\HP> go run "d:\Modul 3\Unguided\unguided3.go"
Lingkaran 1
(cx)(cy)(r) : 1 1 5
Lingkaran 2
(cx)(cy)(r) : 8 8 4
Input koordinat sembarang (x)(y) : 2 2
Titik di dalam lingkaran 1
PS C:\Users\HP> go run "d:\Modul 3\Unguided\unguided3.go"
Lingkaran 1
(cx)(cy)(r) : 1 2 3
Lingkaran 2
(cx)(cy)(r) : 4 5 6
Input koordinat sembarang (x)(y) : 7 8
Titik di dalam lingkaran 2
PS C:\Users\HP> go run "d:\Modul 3\Unguided\unguided3.go"
Lingkaran 1
(cx)(cy)(r) : 5 10 15
Lingkaran 2
(cx)(cy)(r) : -15 4 20
Input koordinat sembarang (x)(y) : 0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\HP> go run "d:\Modul 3\Unguided\unguided3.go"
Lingkaran 1
(cx)(cy)(r) : 1 1 5
Lingkaran 2
(cx)(cy)(r) : 8 8 4
Input koordinat sembarang (x)(y) : 15 20
Titik di luar lingkaran 1 dan 2
PS C:\Users\HP> 

```

Deskripsi Program

Program Go di atas dirancang untuk mengecek posisi sebuah titik terhadap dua lingkaran yang berbeda. Pengguna diminta untuk memasukkan informasi mengenai dua lingkaran, yaitu koordinat pusat (cx , cy) dan jari-jari (r) masing-masing lingkaran. Setelah itu, pengguna juga diminta untuk memasukkan koordinat sembarang (x , y). Program kemudian memeriksa apakah titik tersebut berada di dalam, di luar, atau tepat di tepi salah satu atau kedua lingkaran.

Proses pengecekan dilakukan dengan menggunakan dua fungsi utama. Fungsi `calculateDistance` menghitung jarak antara pusat lingkaran dan titik yang dimasukkan. Fungsi `isPointInCircle` memanfaatkan jarak yang dihitung untuk menentukan apakah titik tersebut berada dalam lingkaran berdasarkan jari-jari yang diberikan. Hasil dari pengecekan ini ditampilkan kepada pengguna, memberikan informasi apakah titik tersebut berada di dalam lingkaran 1, lingkaran 2, keduanya, atau di luar keduanya. Program ini menggabungkan konsep geometri dengan pemrograman untuk memberikan analisis yang sederhana namun efektif.

IV. DAFTAR PUSTAKA

- Novalagung. (t.t.). Fungsi pada Go-Lang. Dasar Pemrograman Go-Lang. Diakses pada 13 Oktober 2024, dari <https://dasarpemrogramangolang.novalagung.com/A-fungsi.html>