

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3



**DISUSUN OLEH:
RIZKINA AZIZAH
103112400082
S1 IF-12-01**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025/2026**

DASAR TEORI

Dalam konteks pemrograman, fungsi adalah sekumpulan blok kode yang dibungkus dengan nama tertentu. Penerapan fungsi yang tepat akan menjadikan kode lebih modular dan juga *dry* (singkatan dari *don't repeat yourself*) yang artinya kita tidak perlu menuliskan banyak kode untuk kegunaan yang sama berulang kali. Cukup deklarasikan sekali saja blok kode sebagai suatu fungsi, lalu panggil sesuai kebutuhan.

Pada chapter ini kita akan belajar tentang penerapannya di Go.

Penerapan Fungsi

Mungkin pembaca sadar, bahwa sebenarnya kita sudah mengimplementasikan fungsi pada banyak praktek sebelumnya, yaitu fungsi `main()`. Fungsi `main()` sendiri merupakan fungsi utama pada program Go, yang akan dieksekusi ketika program dijalankan.

Selain fungsi `main()`, kita juga bisa membuat fungsi lainnya. Dan caranya cukup mudah, yaitu dengan menuliskan keyword `func` kemudian diikuti nama fungsi, lalu kurung `()` (yang bisa diisi parameter), dan diakhiri dengan kurung kurawal untuk membungkus blok kode.

Parameter merupakan variabel yang menempel di fungsi yang nilainya ditentukan saat pemanggilan fungsi tersebut. Parameter sifatnya opsional, suatu fungsi bisa tidak memiliki parameter, atau bisa saja memiliki satu atau banyak parameter (tergantung kebutuhan).

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 `randomWithRange()` didesain untuk *generate* angka acak sesuai dengan range yang ditentukan lewat parameter, yang kemudian angka tersebut dijadikan nilai balik fungsi.

Penggunaan Fungsi `rand.New()`

Fungsi `rand.New()` digunakan untuk membuat object randomizer, yang dari object tersebut kita bisa mendapatkan nilai random/acak hasil generator. Dalam penerapannya, fungsi `rand.New()` membutuhkan argument yaitu random source seed, yang bisa kita buat lewat statement `rand.NewSource(time.Now().Unix())`.

Import Banyak Package

Penulisan keyword import untuk banyak package bisa dilakukan dengan dua cara, dengan menuliskannya di tiap package, atau cukup sekali saja, bebas silakan pilih sesuai selera.

Penggunaan Keyword return Untuk Menghentikan Proses Dalam Fungsi

Selain sebagai penanda nilai balik, keyword return juga bisa dimanfaatkan untuk menghentikan proses dalam blok fungsi di mana ia ditulis. Fungsi divideNumber() dirancang tidak memiliki nilai balik. Fungsi ini dibuat untuk membungkus proses pembagian 2 bilangan, lalu menampilkan hasilnya.

Di dalamnya terdapat proses validasi nilai variabel pembagi, jika nilainya adalah 0, maka akan ditampilkan pesan bahwa pembagian tidak bisa dilakukan, lalu proses dihentikan pada saat itu juga (dengan memanfaatkan keyword return).

GUIDED

1. Guided1

Source Code:

```
//RizkinaAzizah
//perhitungan manual permutasi  $5!/(5-3)! = 5 \times 4 \times 3 \times 2 \times 1 / 2 \times 1 = 60$ 
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{
    hasil := 1
    for i := 1 ; i <= n;i++){
        hasil *= i
    }
    return hasil
}

func permutasi(n, r int)int{
    if r > n{
```

```
    return 0
}
return faktorial(n)/faktorial(n-r)
}
```

Deskripsi Program:

- Mendeklarasikan dua variabel yaitu a dan b dengan tipe data integer
- Instruksi Scan digunakan untuk menerima input nilai a dan b
- Fungsi utama digunakan untuk menentukan nilai n dan r berdasarkan besar kecilnya nilai a dan b
- Fungsi faktorial(n) menghitung faktorial dari n dengan perulangan
- Fungsi permutasi(n, r) menghitung permutasi menggunakan rumus $\text{faktorial}(n)/\text{faktorial}(n-r)$
- Jika $r > n$, permutasi dianggap tidak valid dan fungsi mengembalikan 0.

2. Guided2

Source Code:

```
package main

import (
    "fmt"
)

// Fungsi untuk mengonversi suhu dari Celsius ke Fahrenheit
func celsiusToFahrenheit(celsius float64) float64 {
    return (9.0/5.0)*celsius + 32
}

func main() {
    var N int
    fmt.Print("Masukkan jumlah data: ")
    _, err := fmt.Scan(&N)
    if err != nil || N <= 0 {
```

```

    fmt.Println("Input tidak valid, pastikan memasukkan angka positif.")
    return
}

temperatures := make([]float64, N)

// Membaca suhu dalam Celsius
fmt.Println("Masukkan suhu dalam Celsius:")
for i := 0; i < N; i++ {
    _, err := fmt.Scan(&temperatures[i])
    if err != nil {
        fmt.Println("Input tidak valid, pastikan memasukkan angka.")
        return
    }
}

// Mengonversi ke Fahrenheit dan mencetak hasil
fmt.Println("Suhu dalam Fahrenheit:")
for _, temp := range temperatures {
    fmt.Printf("%.2f\n", celsiusToFahrenheit(temp))
}
}

```

Deskripsi Program:

- Fungsi pada awal digunakan untuk mengonversi suhu celcius ke fahrenheit
- Fungsi utama digunakan untuk :
 - Meminta jumlah data suhu (N).
 - Membaca N suhu dalam Celsius.
 - Mengonversi suhu ke Fahrenheit dan mencetak hasilnya.

3. Guided3

Source Code:

```
package main
```

```

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung luas permukaan tabung
func luasPermukaanTabung(r, t float64) float64 {
    return 2 * math.Pi * r * (r + t)
}

// Fungsi untuk menghitung volume tabung
func volumeTabung(r, t float64) float64 {
    return math.Pi * math.Pow(r, 2) * t
}

func main() {
    var r, t float64

    // Input jari-jari dan tinggi tabung dengan validasi
    fmt.Print("Masukkan jari-jari tabung: ")
    _, errR := fmt.Scan(&r)
    fmt.Print("Masukkan tinggi tabung: ")
    _, errT := fmt.Scan(&t)

    // Memeriksa apakah input valid
    if errR != nil || errT != nil {
        fmt.Println("Input tidak valid! Harap masukkan angka yang benar.")
        return
    }

    // Memeriksa apakah jari-jari dan tinggi bernilai positif
    if r <= 0 || t <= 0 {

```

```

        fmt.Println("Jari-jari dan tinggi tabung harus lebih dari nol.")
        return
    }

    // Menghitung luas permukaan dan volume
    luas := luasPermukaanTabung(r, t)
    volume := volumeTabung(r, t)

    // Menampilkan hasil
    fmt.Println("=====")
    fmt.Printf("Luas Permukaan Tabung: %.2f satuan²\n", luas)
    fmt.Printf("Volume Tabung: %.2f satuan³\n", volume)
    fmt.Println("=====")
}

```

Deskripsi Program:

- Program meminta pengguna untuk memasukkan jari-jari (r) dan tinggi (t) tabung.
- *luasPermukaanTabung(r, t float64) float64* Fungsi ini digunakan untuk menghitung luas permukaan tabung menggunakan rumus $2 \cdot \pi \cdot r \cdot (r + t)$
- *volumeTabung(r, t float64) float64* Fungsi ini digunakan untuk menghitung volume tabung dengan rumus $\pi \cdot r^2 \cdot t$
- Fungsi utama digunakan untuk :
 - Input jari-jari dan tinggi
 - Melakukan validasi input
 - Menghitung luas permukaan tabung
 - Menampilkan output

UNGUIDED

1. Latihan 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 $b \geq d$.

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.

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	

Source Code:

```
//RizkinaAzizah
package main
import "fmt"
func main(){
    var a,b,c,d int
    fmt.Scan(&a, &b, &c, &d)
    if a >= c && b >= d{
        fmt.Println(permutasi(a,c),kombinasi(a,c))
        fmt.Println(permutasi(b,d),kombinasi(b,d))
    }else{
        fmt.Println(permutasi(c,a),kombinasi(c,a))
        fmt.Println(permutasi(d,b),kombinasi(d,b))
    }
}
func faktorial (n int) int{
```

```

    hasil := 1
    for i := 1 ; i <= n; i++{
        hasil *= i
    }
    return hasil
}

func permutasi(n, r int)int{
    if r > n{
        return 0
    }
    return faktorial(n)/faktorial(n-r)
}

func kombinasi(n,r int)int{
    if r > n{
        return 0
    }

    return faktorial(n)/faktorial(r)*faktorial(n-r)
}

```

Output:

```

PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODUL3.go\103112400082_Unguided1.go"
5 10 3 10
60 40
3628800 1
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODUL3.go\103112400082_Unguided1.go"
8 0 2 0
56 14515200
1 1
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2>

```

Deskripsi Program:

- Program ini digunakan untuk menentukan kombinasi dan permutasi
- *Fungsi Utama* digunakan untuk:
 - Memasukkan input
 - Menentukan nilai yang lebih besar
 - Memanggil fungsi
 - Menampilkan output
- *faktorial (n int) int* fungsi ini digunakan untuk menghitung faktorial dari bilangan n menggunakan paradigma perulangan
- *permutasi(n, r int)int* fungsi ini digunakan untuk menghitung permutasi menggunakan rumus $\text{faktorial}(n)/\text{faktorial}(n-r)$
- *func kombinasi(n,r int)int* fungsi ini digunakan untuk menghitung permutasi menggunakan rumus $\text{faktorial}(n)/\text{faktorial}(r)*\text{faktorial}(n-r)$

2. Latihan 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 $(h \circ f \circ g)(c)$!

No	Masukan	Keluaran	Penjelasan
1	7 2 10	36 3 65	$(f \circ g \circ h)(7) = 36$ $(g \circ h \circ f)(2) = 3$ $(h \circ f \circ g)(10) = 65$
2	5 5 5	16 24 10	$(f \circ g \circ h)(5) = 16$ $(g \circ h \circ f)(5) = 24$ $(h \circ f \circ g)(5) = 10$
3	3 8 4	4 63 5	$(f \circ g \circ h)(5) = 4$ $(g \circ h \circ f)(5) = 63$ $(h \circ f \circ g)(5) = 5$

Source Code:

```
package main

import "fmt"

func main(){
    var a, b, c int
    fmt.Scan(&a, &b, &c)

    fmt.Println(fx(gx(hx(a))))
    fmt.Println(gx(hx(fx(b))))
    fmt.Println(hx(fx(gx(c))))
}

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

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

func hx(x int)int{
    return x+1
}
```

Output:

```
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODULE3.go\103112400082_Unguided2.go"
7 2 10
36
3
65
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODULE3.go\103112400082_Unguided2.go"
5 5 5
16
24
10
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODULE3.go\103112400082_Unguided2.go"
3 8 4
4
63
5
```

Deskripsi Program:

- $fx(x \text{ int})int$ fungsi digunakan untuk mengembalikan hasil kuadrat dari x
- $gx(x \text{ int})int$ fungsi digunakan untuk mengurangi x dengan 2
- $hx(x \text{ int})int$ fungsi digunakan untuk menambah x dengan 1
- *Fungsi utama* digunakan untuk membaca input (a, b, c), menghitung dan mencetak hasil operasi

3. Latihan 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".

Source Code:

```
package main

import (
    "fmt"
    "math"
)

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

    fmt.Scan(&cx1, &cy1, &r1)
    fmt.Scan(&cx2, &cy2, &r2)
    fmt.Scan(&x, &y)

    fmt.Println(posisi(cx1, cy1, r1, cx2, cy2, r2, x, y))
}

func jarak(a, b, c, d int) float64 {
    return math.Sqrt(float64((a-c)*(a-c) + (b-d)*(b-d)))
}

func didalam(cx, cy, r, x, y int) bool {
    return jarak(cx, cy, x, y) <= float64(r)
}

func posisi(cx1, cy1, r1, cx2, cy2, r2, x, y int) string {
    square1 := didalam(cx1, cy1, r1, x, y)
    square2 := didalam(cx2, cy2, r2, x, y)

    if square1 && square2 {
        return "Titik didalam lingkaran 1 dan 2"
    }
}
```

```

} else if square1 {
    return "Titik didalam lingkaran 1"
} else if square2 {
    return "Titik didalam lingkaran 2"
} else {
    return "Titik diluar lingkaran 1 dan 2"
}
}
}

```

Output:

```

PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODUL3.go\tempCodeRunnerFile.go"
1 1 5
8 8 4
2 2
Titik didalam lingkaran 1
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODUL3.go\tempCodeRunnerFile.go"
1 2 3
4 5 6
7 8
Titik didalam lingkaran 2
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODUL3.go\tempCodeRunnerFile.go"
5 10 15
-15 4 20
0 0
Titik didalam lingkaran 1 dan 2
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
"c:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2\1031124000
82_MODUL3.go\tempCodeRunnerFile.go"
1 1 5
8 8 4
15 20
Titik diluar lingkaran 1 dan 2

```

Deskripsi Program:

- *Fungsi utama* digunakan untuk membaca input dan memanggil fungsi posisi() untuk letak titik dan mencetak hasil
- *jarak(a, b, c, d int) float64* fungsi ini digunakan untuk menghitung jarak Euclidean antara dua titik (a, b) dan (c, d)
- *didalam(cx, cy, r, x, y int) bool* fungsi ini digunakan untuk menentukan apakah titik (x, y) berada di dalam lingkaran dengan pusat (cx, cy) dan jari-jari r
- *posisi(cx1, cy1, r1, cx2, cy2, r2, x, y int) string* fungsi ini digunakan untuk menentukan posisi titik (x, y) terhadap dua lingkaran, apakah berada di dalam satu, kedua, atau di luar keduanya.

DAFTAR PUSTAKA

Novalagung(2024), A.18.Fungsi – Dasar Pemrograman Golang
(<https://dasarpemrogramangolang.novalagung.com/CHANGELOG.html> ,2024)