

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
MODUL 2  
FUNGSI**



**DISUSUN OLEH:  
Keishin Naufa Alfaridzhi  
103112400061  
S1 IF-12-01**

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## I. DASAR TEORI

### A. Bahasa Yang Digunakan

Pada praktikum ini bahasa pemrograman yang digunakan adalah bahasa pemrograman Go, sesuai dengan modul yang menjadi acuan praktikum. Golang (atau) Go adalah bahasa pemrograman baru, yang mulai dilirik oleh para developer karena kelebihan-kelebihan yang dimilikinya. Sudah banyak Perusahaan besar yang menggunakan bahasa ini untuk produk-produk mereka hingga di level production.

### B. Komentar

Komentar biasa dimanfaatkan untuk menyisipkan catatan pada kode program, menulis penjelasan atau deskripsi mengenai suatu blok kode, atau bisa juga digunakan untuk me-remark kode (men-non-aktifkan kode yang tidak digunakan). Komentar akan diabaikan Ketika kompilasi maupun eksekusi program.

Ada 2 jenis komentar di Golang, yaitu inline dan multiline.

#### 1. Komentar Inline

Penulisan komentar jenis ini diawali dengan tanda *double slash* (*//*) lalu diikuti pesan komentarnya. Komentar inline hanya berlaku untuk satu baris pesan saja. Jika pesan komentar lebih dari satu baris, maka tanda *double slash* harus ditulis lagi di baris selanjutnya.

#### 2. Komentar Multiline

Komentar yang cukup panjang akan lebih rapi jika ditulis menggunakan teknik komentar multiline. Ciri dari komentar jenis ini adalah penulisannya diawali dengan tanda *(/\**) dan diakhiri *(\*/)*.

### C. Variabel

Golang mengadopsi 2 jenis penulisan variabel, yang dituliskan tipe data-nya dan yang tidak. Kedua cara tersebut intinya adalah sama, pembedanya hanyalah cara penulisannya saja. Untuk penulisan variabel dengan tipe data, keyword *var* digunakan untuk deklarasi variabel kemudian diakhiri dengan tipe data misalnya *string*. Kemudian untuk penulisan variabel tanpa tipe data, variabel dideklarasikan dengan menggunakan metode type inference. Penandanya tipe data tidak dituliskan pada saat deklarasi. Pada penggunaan metode ini, operand (*=*) harus diganti dengan (*:=*) dan keyword *var* dihilangkan.

Golang memiliki aturan unik yang tidak dimiliki bahasa lain, yaitu tidak boleh ada satupun variabel yang menganggur. Artinya, semua variabel yang dideklarasikan harus digunakan. Jika terdapat variabel yang tidak digunakan tapi dideklarasikan, program akan gagal dikompilasi. Untuk mengatasi itu, Golang memiliki variabel yaitu underscore. Underscore ( `_` ) adalah predefined variabel yang bisa dimanfaatkan untuk menampung nilai yang tidak dipakai.

#### **D. Tipe Data**

Golang mengenal beberapa jenis tipe data, diantaranya adalah tipe data numerik (decimal dan non-desimal), string, dan boolean.

1. Tipe Data Numerik Non-Desimal (uint, int)
2. Tipe Data Numerik Desimal (float64, float32)
3. Tipe Data Bool (true, false)
4. Tipe Data String (string, “ “)

#### **E. Operator Aritmatika**

Operator aritmatika merupakan operator yang digunakan untuk operasi yang sifatnya perhitungan. Golang mendukung beberapa operator aritmatika standar, yaitu:

1. Penjumlahan (+)
2. Pengurangan (-)
3. Perkalian (\*)
4. Pembagian (/)
5. Modulus atau sisa hasil pembagian (%)

#### **F. Seleksi Kondisi**

Seleksi kondisi pada program berguna untuk mengontrol sebuah blok kode yang akan dieksekusi. Yang dijadikan acuan oleh seleksi kondisi adalah nilai bertipe bool, bisa berasal dari variabel, ataupun hasil operasi perbandingan. Nilai tersebut menentukan blok kode mana yang akan dieksekusi. Go memiliki 2 macam keyword untuk seleksi kondisi, yaitu if else dan switch.

##### **1. If Expression**

If adalah salah satu kata kunci yang digunakan dalam percabangan. Percabangan artinya kita bisa mengeksekusi kode program tertentu ketika suatu kondisi terpenuhi. Hampir semua bahasa pemrograman mendukung if expression.

##### **2. Else if expression**

Terkadang kita butuh membuat beberapa kondisi. Kasus seperti ini dapat menggunakan else if expression. If mendukung short statement sebelum kondisi.

Hal ini sangat cocok untuk membuat statement yang sederhana sebelum melakukan pengecekan terhadap kondisi.

### 3. Switch-Case

Switch merupakan seleksi kondisi yang sifatnya fokus pada satu variabel, lalu kemudian di-cek nilainya. Contoh sederhananya seperti penentuan apakah nilai variabel x adalah: 1, 2, 3, atau lainnya. Perlu diketahui, switch pada pemrograman Go memiliki perbedaan dibanding bahasa lain. Di Go, ketika sebuah case terpenuhi, tidak akan dilanjutkan ke pengecekan case selanjutnya, meskipun tidak ada keyword “break” di situ. Konsep ini berkebalikan dengan switch pada umumnya pemrograman lain (yang ketika sebuah case terpenuhi, maka akan tetap dilanjutkan mengecek case selanjutnya kecuali ada keyword “break”).

## G. Perulangan

Perulangan merupakan proses mengulang dan mengeksekusi blok kode tanpa henti sesuai dengan kondisi yang dijadikan acuan. Biasanya disiapkan variabel untuk iterasi atau penanda kapan perulangan akan dihentikan.

### a. For Loop

For loop merupakan statement perulangan dasar dan cukup sering ditemui. Format for loop yaitu sebagai berikut.

- *Init Statement*: bagian ini akan dieksekusi sebelum perulangan dimulai. Biasanya diisi dengan mendeklarasi variabel iterasi.
- *Condition Expression*: bagian ini akan dicek dan dieksekusi setiap perulangan yang dilakukan, jika true maka perulangan akan terus berjalan hingga kondisi bernilai false.
- *Post Statement*: statement ini akan dieksekusi pada akhir iterasi. Jika terdapat range, maka perulangan akan dieksekusi untuk setiap item pada range.

### b. While Loop

While loop merupakan perulangan yang akan terus berjalan hingga suatu kondisi terpenuhi. Penulisan while loop adalah dengan menuliskan kondisi setelah keyword for (hanya kondisi). Deklarasi dan iterasi variabel counter tidak dituliskan setelah keyword, hanya kondisi perulangan saja. Konsepnya mirip seperti while milik bahasa pemrograman lain.

### c. Repeat Until

Untuk Repeat Until ini mirip seperti for loop biasa namun hanya menggunakan inisiasi dan kondisi saja.

## H. Fungsi

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.

### 1. Penerapan Fungsi

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).

### 2. Fungsi dengan Nilai Balik / Return Value

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.

## II. GUIDED

### 1. Source Code:

```
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)
}
```

Output:

```
D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>
5 2
20
```

Penjelasan:

Program mencari permutasi dari 2 input dengan function faktorial dan permutasi.

### 2. Source Code:

```
package main
import "fmt"

func celciusToFahrenheit(celcius float64) float64 {
    return (9.0/5.0)*celcius + 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)

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

    fmt.Println("Suhu dalam Fahrenheit:")
    for _, temp := range temperatures {
        fmt.Printf("%.2f\n", celciusToFahrenheit(temp))
    }
}

```

Output:

```

D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3\guided2.go"
Masukkan jumlah data: 5
Masukkan suhu dalam Celcius:
27
32
14
2
37
Suhu dalam Fahrenheit:
80.60
89.60
57.20
35.60
98.60

```

Penjelasan:

Program untuk konversi suhu dalam celcius ke fahrenheit, dapat menginput beberapa data sekaligus dalam sekali running program.

### 3. Source Code:

```

package main
import (
    "fmt"
    "math"
)

func luasPermukaanTabung(r, t float64) float64 {

```

```

    return 2 * math.Pi * r * (r+t)
}

func volumeTabung(r, t float64) float64 {
    return math.Pi * math.Pow(r, 2) * t
}

func main() {
    var r, t float64

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

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

    if r <= 0 || t <= 0 {
        fmt.Println("Jari-jari dan tinggi tabung harus lebih dari nol.")
        return
    }

    luas := luasPermukaanTabung(r, t)
    volume := volumeTabung(r, t)

    fmt.Println("=====")
    fmt.Printf("Luas permukaan tabung: %.2f satuan² \n", luas)
    fmt.Printf("Volume tabung: %.2f satuan³ \n", volume)
    fmt.Println("=====")
}

```

Output:

```

D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODULE3>go run "d:\Tugas\SEM2\Alpro\
Masukkan jari-jari tabung: 5
Masukkan tinggi tabung: 8
=====
Luas permukaan tabung: 408.41 satuan²
Volume tabung: 628.32 satuan³
=====

```

Penjelasan:

Program untuk mencari Luas Permukaan Tabung dan Volume Tabung, input berupa bilangan riil dengan error handling.



### III. UNGUIDED

#### 1. Latihan no. 1

Source Code:

```
package main
import "fmt"

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

    fmt.Scanln(&a, &b, &c, &d)

    fmt.Println(permutasi(a,c), kombinasi(a,c))
    fmt.Println(permutasi(b,d), kombinasi(b,d))
}

func permutasi(x, y int) int {
    var n, hasil int
    n = factorial(x)
    hasil = n/factorial(x-y)
    return hasil
}

func kombinasi(x, y int) int {
    var n, r, hasil int
    n = factorial(x)
    r = factorial(y)
    hasil = n/(r*(factorial(x-y)))
    return hasil
}

func factorial(n int) int{
    if n <= 1 {
        return 1
    }

    return n * factorial(n-1)
}
```

Output:

```
D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\
5 10 3 10
60 10
3628800 1

D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\
8 0 2 0
56 28
1 1
```

### Deskripsi Program:

Mencari nilai permutasi dan kombinasi dari empat bilangan yang diinput. Menghitung input pertama dengan input ketiga dan menghitung input kedua dan keempat. Terdapat 3 function yaitu permutasi, kombinasi, dan faktorial yang ketiganya membaca dan mengeluarkan output berupa bilangan integer.

### 2. Latihan no. 2

Source Code:

```
package main
import "fmt"

func main() {
    var a, b, c int
    fmt.Scanln(&a, &b, &c)
    fogoh := f(g(h(a)))
    gohof := g(h(f(b)))
    hofog := h(f(g(c)))
    fmt.Printf("%v\n%v\n%v\n", fogoh, gohof, hofog)
}

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

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

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

### Output:

```
D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_
7 2 10
36
3
65

D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_
5 5 5
16
24
10

D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_
3 8 4
4
63
5
```

Deskripsi Program:

Program mencari fungsi komposisi dari 3 fungsi yaitu  $f(x)$ ,  $g(x)$ ,  $h(x)$ . Komposisi dapat dilakukan dengan cara memasukkan suatu function yang akan digabungkan ke dalam parameter function yang menjadi induk. Contoh fog, maka penulisan dalam Go adalah  $f(g(x))$ .

### 3. Latihan no. 3

Source Code:

```
package main
import (
    "fmt"
    "math"
)

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

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

    dalam1 := didalam(cx1, cy1, r1, x, y)
    dalam2 := didalam(cx2, cy2, r2, x, y)

    if dalam1 && dalam2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalam1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dalam2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

func jarak(a, b, c, d float64) float64 {
    jarak := math.Sqrt(math.Pow(a-c, 2) + math.Pow(b-d, 2))
    return jarak
}

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

Output:

```
D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_M
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1

D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_M
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2

D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_M
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2

D:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_MODUL3>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul3\103112400061_M
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
```

Deskripsi Program:

Program untuk menentukan apakah suatu titik koordinat berada di dalam radius lingkaran. Terdapat 2 function, jarak untuk menghitung jarak antara titik (a, b) dan (c, d), dan didalam untuk menentukan apakah titik koordinat yang diinputkan berada dalam lingkaran dengan melakukan cek apakah jarak  $\leq$  radius lingkaran.

#### IV. KESIMPULAN

Pada praktikum ini telah dibahas perihal cara deklarasi dan menggunakan function dalam bahasa Go. Dalam Go, function memiliki tipe data yang dikembalikan dan harus menggunakan *return* jika ada nilai yang akan dikembalikan. Deklarasi function dapat dilakukan dengan mengetikkan 'func namaFunction()' diikuti dengan tipe data yang akan di-return. Kemudian cara pemanggilannya dapat langsung dilakukan dalam `fmt.Print()` ataupun dapat di-assign dalam variable terlebih dahulu.

## V. DAFTAR PUSTAKA

Noval Agung Prayogo. *Dasar Pemrograman Golang*. Diakses pada 01 Oktober 2024.  
<https://dasarpemrogramangolang.novalagung.com>

Annisa Nur Isnaeni. *Golang — Seleksi Kondisi*. Diakses pada 01 Oktober 2024.  
<https://medium.com/@annisaisna/golang-seleksi-kondisi-f988ead004b4>

Parvez Alam, *Golang for loop example | Golang Loops Tutorial – Phpflow.com*  
<https://medium.com/@parvez1487/golang-for-loop-example-golang-loops-tutorial-phpflow-com-f4b2b0e57944>