

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 3
FUNGSI**



Disusun Oleh :

ANDIKA NEVIANTORO / 2311102167

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

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.

Deklarasi Parameter Bertipe Data Sama

Khusus untuk fungsi yang tipe data parameternya sama, bisa ditulis dengan gaya yang unik. Tipe datanya dituliskan cukup sekali saja di akhir. Contohnya bisa dilihat pada kode berikut.

```
func nameOfFunc(paramA type, paramB type, paramC type) returnType  
func nameOfFunc(paramA, paramB, paramC type) returnType  
  
func randomWithRange(min int, max int) int  
func randomWithRange(min, max int) int
```

Penerapan Fungsi Sebagai Parameter

Cara membuat parameter fungsi adalah dengan langsung menuliskan skema fungsi nya sebagai tipe data. Contohnya bisa dilihat pada kode berikut.

```
package main  
  
import "fmt"  
import "strings"  
  
func filter(data []string, callback func(string) bool) []string {  
    var result []string  
    for _, each := range data {  
        if filtered := callback(each); filtered {  
            result = append(result, each)  
        }  
    }  
    return result  
}
```

Parameter callback merupakan sebuah closure yang dideklarasikan bertipe `func(string) bool`. Closure tersebut dipanggil di tiap perulangan dalam fungsi `filter()`.

Fungsi `filter()` sendiri kita buat untuk filtering data array (yang datanya didapat dari parameter pertama), dengan kondisi filter bisa ditentukan sendiri. Di bawah ini adalah contoh pemanfaatan fungsi tersebut.

```
func main() {
    var data = []string{"wick", "jason", "ethan"}
    var dataContainsO = filter(data, func(each string) bool
    {
        return strings.Contains(each, "o")
    })
    var dataLength5 = filter(data, func(each string) bool
    {
        return len(each) == 5
    })

    fmt.Println("data asli \t\t:", data)
    // data asli : [wick jason ethan]

    fmt.Println("filter ada huruf \"o\" \t:", dataContainsO)
    // filter ada huruf "o" : [jason]

    fmt.Println("filter      jumlah      huruf      \"5\" \t:",
dataLength5)
    // filter jumlah huruf "5" : [jason ethan]
}
```

Ada cukup banyak hal yang terjadi di dalam tiap pemanggilan fungsi `filter()` di atas. Berikut adalah penjelasannya:

1. Data array (yang didapat dari parameter pertama) akan di-looping.
2. Di tiap perulangannya, closure callback dipanggil, dengan disisipkan data tiap elemen perulangan sebagai parameter.
3. Closure callback berisikan kondisi filtering, dengan hasil bertipe `bool` yang kemudian dijadikan nilai balik dikembalikan.
4. Di dalam fungsi `filter()` sendiri, ada proses seleksi kondisi (yang nilainya didapat dari hasil eksekusi closure callback). Ketika kondisinya bernilai `true`, maka data elemen yang sedang diulang dinyatakan lolos proses filtering.

5. Data yang lolos ditampung variabel result. Variabel tersebut dijadikan sebagai nilai balik fungsi filter().

```
[novalagung:belajar-golang $ go run bab21.go  
data asli      : [wick jason ethan]  
filter ada huruf "o" : [jason]  
filter jumlah huruf "5" : [jason ethan]  
novalagung:belajar-golang $
```

Pada dataContainsO, parameter kedua fungsi filter() berisikan statement untuk deteksi apakah terdapat substring "o" di dalam nilai variabel each (yang merupakan data tiap elemen), jika iya, maka kondisi filter bernilai true, dan sebaliknya.

Pada contoh ke-2 (dataLength5), closure callback berisikan statement untuk deteksi jumlah karakter tiap elemen. Jika ada elemen yang jumlah karakternya adalah 5, berarti elemen tersebut lolos filter.

Memang butuh usaha ekstra untuk memahami pemanfaatan closure sebagai parameter fungsi. Tapi setelah paham, penerapan teknik ini pada kondisi yang tepat akan sangat berguna.

Pemanggilan Function atau Fungsi

Fungsi dipanggil dengan menyebutkan namanya diikuti oleh argumen dalam tanda kurung. Hasil dari fungsi dapat disimpan dalam variabel atau bisa digunakan secara langsung dalam pernyataan.

- Fungsi Variadic (Variadic Function) : Bahasa Pemrograman Golang mendukung parameter variadic, yang memungkinkan seorang Backend Developer untuk mengirim jumlah argumen yang tidak terbatas ke fungsi. Parameter variadic ini ditandai dengan `...`.
- Fungsi Anonim (Anonym Function) : Seorang Backend Developer dapat membuat fungsi tanpa nama atau yang lebih dikenal sebagai fungsi anonim (Anonym Function), yang digunakan dalam ekspresi lambda. Fungsi Anonim (Anonym Function) ini sangat berguna dalam kasus penggunaan singkat.
- Fungsi Metode (Method Function): Dalam Bahasa Pemrograman Golang, seorang Backend Developer dapat memiliki metode yang terkait dengan tipe data kustom sendiri. Metode adalah fungsi yang “terikat” ke tipe data tertentu dan memungkinkan seorang Backend Developer melakukan operasi pada instance dari tipe data tersebut.

- Fungsi Rekursif (Recursif Function) : Function atau Fungsi dapat memanggil dirinya sendiri, yang dikenal sebagai rekursi. Fungsi rekursif (Recursif Function) berguna untuk menyelesaikan masalah berulang atau terpecahkan secara berulang.
- Fungsi Closure :Bahasa Pemrograman Golang mendukung closure, di mana fungsi dalam fungsi lain dapat mengakses variabel yang dideklarasikan di luar lingkup fungsi tersebut.
- Error Handling : Function atau Fungsi sering mengembalikan nilai kesalahan (error) untuk mengindikasikan jika operasi fungsi gagal. Hal ini membantu dalam penanganan kesalahan atau error yang baik.
- Penggunaan Goroutine : Function atau Fungsi dapat digunakan dalam pembuatan goroutine, yang merupakan fitur konkurensi dari bahasa pemrograman Golang untuk menjalankan tugas secara bersamaan.

II. GUIDED

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

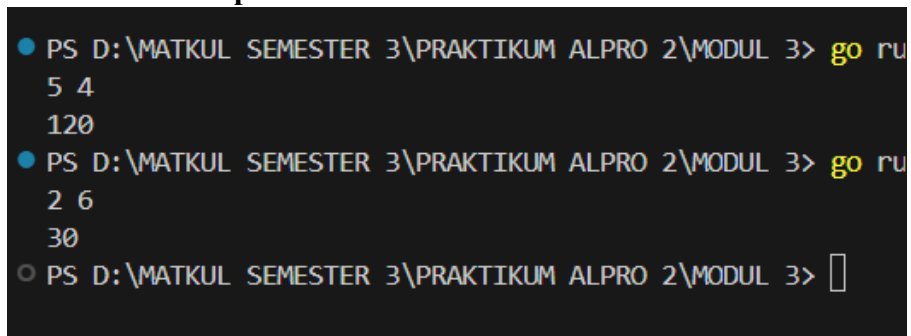
import "fmt"

func main() {
    var a, b int
    fmt.Scan(&a, &b)
    if a >= b {
        fmt.Println(permutasi(a, b))
    } 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)
}
```

Screenshot Output :



```
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go run main.go
5 4
120
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go run main.go
2 6
30
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> 
```

Deskripsi Program :

Program di atas bertujuan untuk menghitung permutasi dari dua bilangan input. Alur program dimulai dengan membaca dua bilangan bulat `a` dan `b` dari input pengguna. Program kemudian memeriksa apakah `a` lebih besar atau sama dengan `b`. Jika benar, program memanggil fungsi `permutasi(a, b)`; jika tidak, program menukar nilai `a` dan `b` dan memanggil `permutasi(b, a)`.

Fungsi `faktorial(n)` digunakan untuk menghitung faktorial dari bilangan `n`, dengan cara mengalikan semua bilangan dari 1 hingga `n`. Fungsi `permutasi(n, r)` menggunakan fungsi faktorial untuk menghitung permutasi dengan rumus $P(n, r) = \frac{n!}{(n-r)!}$. Hasil permutasi ini kemudian ditampilkan sebagai output. Tujuan utama dari program ini adalah menghitung permutasi dua bilangan sesuai dengan aturan matematika diskrit.

2. Program beserta fungsi yang digunakan untuk menghitung luas dan keliling persegi. Masukan terdiri dari sisi persegi. Keluaran berupa hasil luas dan keliling persegi.

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung luas persegi
func luasPersegi(sisi float64) float64 {
    return sisi * sisi
}

// Fungsi untuk menghitung keliling persegi
func kelilingPersegi(sisi float64) float64 {
    return 4 * sisi
}

func main() {
    var sisi float64

    // Meminta input dari user
    fmt.Print("Masukkan panjang sisi persegi: ")
    fmt.Scan(&sisi)

    // Menghitung luas dan keliling
    luas := luasPersegi(sisi)
    keliling := kelilingPersegi(sisi)

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

Screenshoot Output :

```
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go
Masukkan panjang sisi persegi: 5
Luas persegi: 25.00
Keliling persegi: 20.00
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> █
```

Deskripsi Program :

Program di atas bertujuan untuk menghitung luas dan keliling persegi berdasarkan panjang sisi yang diinput oleh pengguna. Alurnya dimulai dengan mendefinisikan dua fungsi: `luasPersegi(sisi)` untuk menghitung luas persegi dengan rumus ***luas = sisi x sisi***, dan `kelilingPersegi(sisi)` untuk menghitung keliling persegi dengan rumus ***keliling = 4 x sisi***. Di dalam fungsi `main()`, program meminta input panjang sisi dari pengguna, kemudian menghitung luas dan keliling persegi menggunakan fungsi yang telah dibuat. Hasil perhitungan luas dan keliling tersebut ditampilkan dengan dua angka di belakang koma. Tujuan program ini adalah mempermudah pengguna dalam menghitung dan mendapatkan hasil luas serta keliling persegi secara otomatis.

2. UNGUIDED

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 bersedialah 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 . Catatan: permutasi (P) dan kombinasi (C) dari n terhadap r ($n \geq r$) dapat dihitung dengan menggunakan persamaan berikut!

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

Sourcecode

```
package main

import (
    "fmt"
)
```



```

// Fungsi untuk menghitung faktorial
func factorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
    return result
}

// Fungsi untuk menghitung permutasi P(n, r)
func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

// Fungsi untuk menghitung kombinasi C(n, r)
func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func main() {
    // Input empat bilangan a, b, c, d
    var a, b, c, d int
    fmt.Print("Masukkan empat bilangan bulat (a >= c dan b >= d): ")
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    // Syarat a >= c dan b >= d
    if a >= c && b >= d {
        // Menghitung permutasi dan kombinasi untuk a dan c
        P_ac := permutation(a, c)
        C_ac := combination(a, c)

        // Menghitung permutasi dan kombinasi untuk b dan d
        P_bd := permutation(b, d)
        C_bd := combination(b, d)

        // Output hasil permutasi dan kombinasi
        fmt.Printf("%d %d\n", P_ac, C_ac)
        fmt.Printf("%d %d\n", P_bd, C_bd)
    } else {
        fmt.Println("Input tidak valid. Pastikan a >= c dan b >= d.")
    }
}

```

Screenshoot Output :

```
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go r
Masukkan empat bilangan bulat (a >= c dan b >= d): 5 10
60 10
3628800 1
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go r
Masukkan empat bilangan bulat (a >= c dan b >= d): 8 0
56 28
1 1
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> 
```

Deskripsi Program :

Program ini menghitung permutasi dan kombinasi dari dua set bilangan yang dimasukkan pengguna. Program meminta empat bilangan `a`, `b`, `c`, dan `d` dengan syarat `a >= c` dan `b >= d`. Jika syarat terpenuhi, program menghitung dan menampilkan hasil permutasi $P(a, c)$ dan $P(b, d)$, serta kombinasi $C(a, c)$ dan $C(b, d)$. Jika tidak, program menampilkan pesan kesalahan. Tujuannya adalah untuk menghitung permutasi dan kombinasi dua set bilangan secara otomatis.

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"
)

// 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.Println("Masukkan tiga bilangan bulat a, b, dan c:")
    fmt.Scan(&a, &b, &c)

    // Hitung komposisi fungsi
    fogohResult := fogoh(a)
    gohofResult := gohof(b)
    hofogResult := hofog(c)

    // Tampilkan hasil
    fmt.Printf("(fogoh) (%d) = %d\n", a, fogohResult)
    fmt.Printf("(gohof) (%d) = %d\n", b, gohofResult)
    fmt.Printf("(hofog) (%d) = %d\n", c, hofogResult)
}
```

Screenshoot Output :

```
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go t
Masukkan tiga bilangan bulat a, b, dan c:
7 2 10
(fogoh)(7) = 36
(gohof)(2) = 3
(hofog)(10) = 65
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go t
Masukkan tiga bilangan bulat a, b, dan c:
5 5 5
(fogoh)(5) = 16
(gohof)(5) = 24
(hofog)(5) = 10
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go t
Masukkan tiga bilangan bulat a, b, dan c:
3 8 4
(fogoh)(3) = 4
(gohof)(8) = 63
(hofog)(4) = 5
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> □
```

Deskripsi Program :

Program ini menghitung hasil komposisi tiga fungsi matematika $f(x) = x^2$, $g(x) = x - 2$, dan $h(x) = x + 1$. Pengguna diminta memasukkan tiga bilangan bulat a , b , dan c . Program kemudian menghitung dan menampilkan hasil dari tiga komposisi fungsi: $(fogoh)(a) = f(g(h(a)))$, $(gohof)(b) = g(h(f(b)))$, dan $(hofog)(c) = h(f(g(c)))$. Tujuan program adalah untuk menunjukkan cara komposisi fungsi bekerja dalam pemrograman dan matematika.

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

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

Sourcecode :

```
package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung jarak antara dua titik (a,b)
// dan (c,d)
func jarak(a, b, c, d float64) float64 {
    return math.Sqrt(math.Pow(a-c, 2) + math.Pow(b-d,
2))
}

// Fungsi untuk menentukan apakah titik (x, y) berada di
// dalam lingkaran dengan pusat (cx, cy) dan radius r
func didalam(cx, cy, r, x, y float64) bool {
    return jarak(cx, cy, x, y) <= r
}

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

    // Input koordinat lingkaran 1 (pusat dan radius)
    fmt.Println("Masukkan pusat (cx1, cy1) dan radius
(r1) lingkaran 1:")
    fmt.Scan(&cx1, &cy1, &r1)

    // Input koordinat lingkaran 2 (pusat dan radius)
    fmt.Println("Masukkan pusat (cx2, cy2) dan radius
(r2) lingkaran 2:")
    fmt.Scan(&cx2, &cy2, &r2)

    // Input koordinat titik sembarang (x, y)
    fmt.Println("Masukkan koordinat titik sembarang (x,
y):")
    fmt.Scan(&x, &y)

    // Cek posisi titik terhadap lingkaran 1 dan 2
    diDalamLingkaran1 := didalam(cx1, cy1, r1, x, y)
    diDalamLingkaran2 := didalam(cx2, cy2, r2, x, y)
```

```

// Tentukan output berdasarkan posisi titik
if diDalamLingkaran1 && diDalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if diDalamLingkaran1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if diDalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Screenshoot Output :

```

● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go run "
Masukkan pusat (cx1, cy1) dan radius (r1) lingkaran 1:
1 1 5
Masukkan pusat (cx2, cy2) dan radius (r2) lingkaran 2:
8 8 4
Masukkan koordinat titik sembarang (x, y):
2 2
Titik di dalam lingkaran 1
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go run "
Masukkan pusat (cx1, cy1) dan radius (r1) lingkaran 1:
1 2 3
Masukkan pusat (cx2, cy2) dan radius (r2) lingkaran 2:
4 5 6
Titik di dalam lingkaran 2
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go run "
Masukkan pusat (cx1, cy1) dan radius (r1) lingkaran 1:
5 10 15
Masukkan pusat (cx2, cy2) dan radius (r2) lingkaran 2:
-15 4 20
Masukkan koordinat titik sembarang (x, y):
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> go run "
Masukkan pusat (cx1, cy1) dan radius (r1) lingkaran 1:
1 1 5
Masukkan pusat (cx2, cy2) dan radius (r2) lingkaran 2:
8 8 4
Masukkan koordinat titik sembarang (x, y):
15 20
Titik di luar lingkaran 1 dan 2
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 3> █

```

Deskripsi Program :

Program ini bertujuan untuk menentukan posisi suatu titik sembarang terhadap dua buah lingkaran. Pertama, program meminta input berupa koordinat pusat dan radius untuk dua lingkaran serta koordinat sebuah titik sembarang. Kemudian, program menghitung jarak antara titik sembarang dan pusat masing-masing lingkaran menggunakan rumus jarak Euclidean. Berdasarkan jarak tersebut, program menentukan apakah titik berada di dalam lingkaran 1, lingkaran 2, di dalam kedua lingkaran, atau di luar keduanya, lalu menampilkan hasil tersebut.

KESIMPULAN

Kesimpulan dari materi ini adalah bahwa fungsi dalam pemrograman Go sangat penting untuk membuat kode lebih modular dan efisien. Fungsi membantu menghindari penulisan kode yang berulang (prinsip DRY) dengan mendeklarasikan logika sekali dan memanggilnya sesuai kebutuhan. Selain itu, Go mendukung berbagai jenis fungsi seperti variadic, anonim, metode, rekursif, dan closure, yang memungkinkan fleksibilitas dalam pemrograman. Fungsi juga dapat menerima fungsi lain sebagai parameter (callback), yang sangat berguna untuk kasus seperti filtering data. Golang menyediakan cara untuk menangani kesalahan (error handling) dan mendukung pemrograman konkuren melalui goroutine.

REFERENSI

[1] Agung, N. dasar pemrograman golang :

<https://dasarpemrogramangolang.novalagung.com/A-fungsi.html>

[2] Medium. Function on Golang :

<https://medium.com/@myskill.id/function-on-golang-27b6577e9cbe>