LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 3

FUNGSI

Oleh:

DWI OKTA SURYANINGRUM

103112400066

12-IF-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

1.1 Fungsi dalam Pemrograman

Fungsi adalah sekumpulan kode yang memiliki nama tertentu dan digunakan untuk menjalankan tugas tertentu. Penggunaan fungsi membuat kode lebih modular dan efisien karena dapat dipanggil berulang kali tanpa perlu menulis kode yang sama. Fungsi dapat memiliki parameter yang diteruskan saat pemanggilannya, serta nilai balik (return value) yang dikembalikan setelah fungsi selesai dieksekusi.

1.2 Penerapan Fungsi

Fungsi utama dalam Go adalah main(), yang merupakan titik awal eksekusi program. Selain main(), kita bisa membuat fungsi lainnya dengan menggunakan kata kunci func, diikuti dengan nama fungsi, parameter (opsional), dan blok kode. Parameter adalah variabel yang diteruskan ke dalam fungsi, dan argument adalah nilai yang diberikan untuk parameter tersebut saat fungsi dipanggil.

1.3 Fungsi dengan Return Value

Fungsi dapat mengembalikan nilai menggunakan return. Fungsi dengan return value memiliki tipe data yang harus ditentukan setelah parameter fungsi. Fungsi yang tidak mengembalikan nilai disebut fungsi void, seperti main().

1.4 **Penggunaan Fungsi rand.New()**

Fungsi rand.New() digunakan untuk menghasilkan angka acak. Fungsi ini membutuhkan sumber acak atau "seed", yang sering kali menggunakan waktu sistem untuk memastikan angka acak yang dihasilkan berbeda setiap kali program dijalankan.

1.5 Import Banyak Package

Kita dapat mengimpor lebih dari satu package dalam Go dengan menuliskannya satu per satu atau menggunakan sintaks import blok, yang memudahkan untuk mengelola banyak import.

1.6 Deklarasi Parameter Bertipe Data Sama

Jika parameter fungsi memiliki tipe data yang sama, kita bisa menuliskannya secara bersamaan untuk membuat kode lebih ringkas.

1.7 Penggunaan Keyword return

Selain digunakan untuk mengembalikan nilai, return juga berfungsi untuk menghentikan eksekusi dalam fungsi. Dengan menggunakan return, program dapat keluar dari fungsi lebih awal jika kondisi tertentu terpenuhi.

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

a. Guided 1

```go
// DWI OKTA SURYANINGRUM
package main

import "fmt"

// Fungsi utama untuk menjalankan program
func main() {
    var a, b int // mendeklarasikan dua variabel bertipe integer untuk menyimpan input dari pengguna
    fmt.Scan(&a, &b) // membaca input dua angka yang diberikan oleh pengguna dan menyimpannya dalam variabel a dan b

    // mengecek apakah nilai a lebih besar atau sama dengan b
    if a >= b {
        // jika a lebih besar atau sama dengan b, maka memanggil fungsi permutasi dengan argumen a dan b
        fmt.Println(permutasi(a, b))
    } else {
        // jika b lebih besar dari a, maka memanggil fungsi permutasi dengan argumen b dan a
        fmt.Println(permutasi(b, a))
    }
}

// Fungsi faktorial digunakan untuk menghitung faktorial dari suatu bilangan
func faktorial(n int) int {
    hasil := 1 // inisialisasi variabel hasil dengan nilai awal 1
    // perulangan untuk menghitung faktorial dari 1 hingga n
    for i := 1; i <= n; i++ {
        hasil *= i // mengalikan nilai hasil dengan i pada setiap iterasi
    }
    // mengembalikan hasil faktorial
    return hasil
}

// fungsi permutasi digunakan untuk menghitung permutasi dari dua angka n dan r
func permutasi(n, r int) int {
    // memeriksa apakah r lebih besar dari n, jika iya permutasi tidak valid
    if r > n {
        return 0 // mengembalikan 0 jika r lebih besar dari n karena permutasi tidak dapat dihitung
    }
    // jika r tidak lebih besar dari n, maka menghitung permutasi dengan rumus: n! / (n - r)!
    // pertama menghitung faktorial dari n, lalu membaginya dengan faktorial dari (n - r)
    return faktorial(n) / faktorial(n - r)
}
```

Output :

```
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/103112400066_Guided1.go"
5 10
30240
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/103112400066_Guided1.go"
1 2
2
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/103112400066_Guided1.go"
10 5
30240
```

b. Guided 2

```go
// DWI OKTA SURYANINGRUM
package main

import "fmt"

// Fungsi untuk mengonversi suhu dari Celsius ke Fahrenheit
func celsiusToFahrenheit(celsius float64) float64 {
    // Fungsi ini menerima suhu dalam Celsius dan mengonversinya ke Fahrenheit menggunakan rumus
    return (9.0/5.0)*celsius + 32
}

func main() {
    // Deklarasi variabel N untuk jumlah suhu yang akan dikonversi
    var N int
    fmt.Print("Masukkan jumlah data: ")

    // Membaca input dari pengguna untuk jumlah data suhu yang akan dimasukkan
    _, err := fmt.Scan(&N)
    // Melakukan validasi input, jika N kurang dari atau sama dengan 0 atau terdapat error, maka tampilkan pesan error
    if err != nil || N <= 0 {
        fmt.Println("Input tidak valid, pastikan memasukkan angka positif.")
        return
    }

    // Membuat slice temperatures untuk menyimpan suhu dalam Celsius sebanyak N data
    temperatures := make([]float64, N)

    // Membaca N buah suhu dalam Celsius dari pengguna
    fmt.Println("Masukkan suhu dalam Celsius:")
    for i := 0; i < N; i++ {
        // Melakukan perulangan sebanyak N kali untuk membaca setiap suhu
        _, err := fmt.Scan(&temperatures[i])
        // Jika input bukan angka, maka tampilkan pesan error dan hentikan program
        if err != nil {
            fmt.Println("Input tidak valid, pastikan memasukkan angka.")
            return
        }
    }

    // Mengonversi suhu ke Fahrenheit dan mencetak hasilnya
    fmt.Println("Suhu dalam Fahrenheit:")
    // Perulangan untuk membaca setiap elemen suhu dalam slice temperatures
    for _, temp := range temperatures {
        // Mencetak hasil konversi suhu dengan dua angka di belakang koma
        fmt.Printf("%.2f\n", celsiusToFahrenheit(temp))
    }
}
```

Output :

```
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/103112400066_Guided2.go"
Masukkan jumlah data: 3
Masukkan suhu dalam Celsius:
3
2
1
Suhu dalam Fahrenheit:
37.40
35.60
33.80
```

c. Guided 3

```go
// DWI OKTA SURYANINGRUM
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("================================")
```

Output :

```
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/103112400066_Guided3.go"
Masukkan jari-jari tabung: 10
Masukkan tinggi tabung: 15
==================================
Luas Permukaan Tabung: 1570.80 satuan²
Volume Tabung: 4712.39 satuan³
==================================
```

III. UNGUIDED

a. Unguided 1

```go
// DWI OKTA SURYANINGRUM
package main

import "fmt"

func main() {
    // deklarasi variabel a, b, c, dan d bertipe integer
    var a, b, c, d int

    // menerima input untuk variabel a, b, c, dan d secara bersamaan
    fmt.Scan(&a, &b, &c, &d)

    // memeriksa apakah a >= c dan b >= d
    if a >= c && b >= d {
        // jika kondisi terpenuhi, menampilkan hasil permutasi dan kombinasi
        // dari (a, c) dan (b, d)
        fmt.Println(permutasi(a,c), kombinasi(a,c))
        fmt.Println(permutasi(b,d), kombinasi(b,d))
    } else {
        // Jika kondisi tidak terpenuhi, menampilkan pesan kesalahan
        fmt.Println("input tidak sesuai")
    }
}

// fungsi untuk menghitung faktorial dari n
func faktorial(n int) int {
    hasil := 1
    // menghitung hasil faktorial dengan perkalian berulang dari 1 hingga n
    for i := 1; i <= n; i++ {
        hasil *= i
    }
    // mengembalikan hasil faktorial
    return hasil
}

// fungsi untuk menghitung permutasi, rumus: n! / (n-r)!
func permutasi(n, r int) int {
    // jika r lebih besar dari n, permutasi tidak valid, mengembalikan 0
    if r > n {
        return 0
    }
    // menghitung permutasi dengan menggunakan fungsi faktorial
    return faktorial(n) / faktorial(n-r)
}

// fungsi untuk menghitung kombinasi, rumus: n! / (r! * (n-r)!)
func kombinasi(n, r int) int {
    // jika r lebih besar atau sama dengan n, kombinasi tidak valid
    if r >= n {
        return 0
    }
    // menghitung kombinasi dengan menggunakan fungsi faktorial
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}
```

Output :

```
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided1.go"
5 10 3 10
60 10
3628800 1
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided1.go"
8 0 2 0
56 28
1 1
```

b. Unguided 2

```go
// DWI OKTA SURYANINGRUM
package main

import "fmt"

func main() {
    // deklarasi variabel a, b, dan c bertipe integer
    var a, b, c int

    // membaca inputan dari user dan menyimpannya di variabel a, b, dan c
    fmt.Scan(&a, &b, &c)

    // memanggil fungsi fx, gx, dan hx secara berurutan dengan parameter yang sesuai
    // melakukan perhitungan fx(gx(hx(a))) dan menampilkan hasilnya
    fmt.Println(fx(gx(hx(a))))

    // melakukan perhitungan gx(hx(fx(b))) dan menampilkan hasilnya
    fmt.Println(gx(hx(fx(b))))

    // melakukan perhitungan hx(fx(gx(c))) dan menampilkan hasilnya
    fmt.Println(hx(fx(gx(c))))
}

// fungsi untuk menghitung x^2
func fx(x int) int {
    // mengembalikan hasil perkalian x dengan dirinya sendiri (x^2)
    return x * x
}

// fungsi untuk menghitung x - 2
func gx(x int) int {
    // mengembalikan hasil pengurangan x dengan 2
    return x - 2
}

// fungsi untuk menghitung x + 1
func hx(x int) int {
    // mengembalikan hasil penjumlahan x dengan 1
    return x + 1
}
```

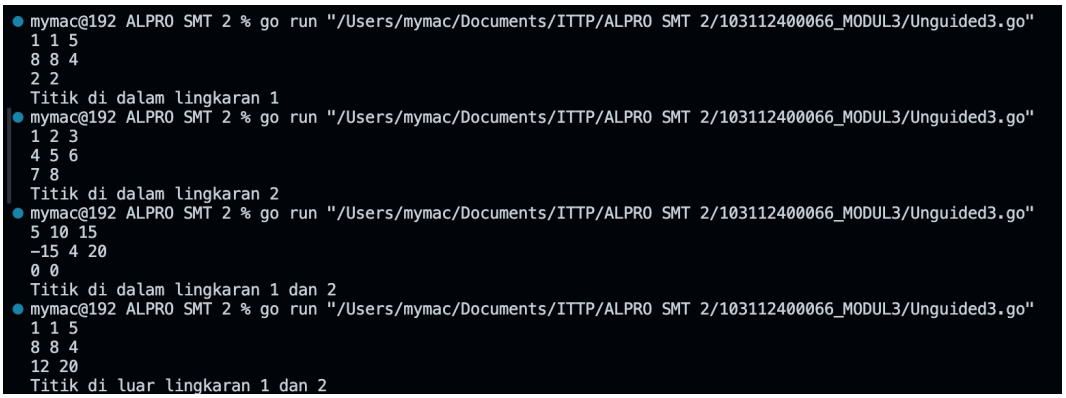
Output :

```
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided2.go"
7 2 10
36
3
65
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided2.go"
5 5 5
16
24
10
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided2.go"
3 8 4
4
63
5
```

c. Unguided 3

```go
// DWI OKTA SURYANINGRUM
package main

import (
    "fmt"
    "math"
)

func main() {
    // deklarasi variabel untuk pusat dan jari-jari lingkaran 1 dan 2, serta titik sembarang
    var cy1, cx1, r1 int
    var cy2, cx2, r2 int
    var y, x int

    // membaca input untuk lingkaran 1 (pusat dan jari-jari) dan menyimpannya
    fmt.Scan(&cy1, &cx1, &r1)

    // membaca input untuk lingkaran 2 (pusat dan jari-jari) dan menyimpannya
    fmt.Scan(&cy2, &cx2, &r2)

    // membaca input untuk titik sembarang (koordinat titik) dan menyimpannya
    fmt.Scan(&y, &x)

    // konversi nilai inputan ke float64 agar sinkron dengan fungsi perhitungan
    cy1f, cx1f, r1f := float64(cy1), float64(cx1), float64(r1)
    cy2f, cx2f, r2f := float64(cy2), float64(cx2), float64(r2)
    yf, xf := float64(y), float64(x)

    // mengecek apakah titik berada di dalam lingkaran 1 dan/atau lingkaran 2
    dalam1 := didalam(cx1f, cy1f, r1f, xf, yf)
    dalam2 := didalam(cx2f, cy2f, r2f, xf, yf)

    // output hasil berdasarkan kondisi apakah titik berada di dalam lingkaran 1 dan/atau 2
    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")
    }
}

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

// membuat fungsi untuk mengecek apakah titik (x,y) berada di dalam lingkaran dengan pusat (cx,cy) dan jari-jari r
func didalam(cx, cy, r, x, y float64) bool {
    // menghitung jarak dari titik (x, y) ke pusat lingkaran (cx, cy)
    // jika jarak kurang dari atau sama dengan jari-jari lingkaran, maka titik berada di dalam lingkaran
    return jarak(cx, cy, x, y) <= r
}
```

Output :


mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided3.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided3.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided3.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/103112400066_MODUL3/Unguided3.go"
1 1 5
8 8 4
12 20
Titik di luar lingkaran 1 dan 2

IV. KESIMPULAN

Fungsi dalam pemrograman Go membantu membuat kode lebih modular dan efisien. Fungsi utama main() adalah titik awal eksekusi program, sementara fungsi lainnya dapat dibuat dengan kata kunci func. Fungsi bisa mengembalikan nilai menggunakan return, dan yang tidak mengembalikan nilai disebut fungsi void.

Go juga menyediakan fungsi seperti **rand.New()** untuk menghasilkan angka acak dengan "seed" waktu sistem. Package dapat diimpor sekaligus dalam satu blok, dan deklarasi parameter bertipe sama dapat disingkat dalam satu baris. Penggunaan fungsi dan manajemen package mempermudah pengembangan perangkat lunak yang lebih efisien.

V. REFERENSI

A.19. Fungsi Multiple Return. (n.d.). Retrieved from https://dasarpemrogramangolang.novalagung.com/A-fungsi-multiple-return.html