

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 2**  
**MODUL 3**  
**FUNGSI**



**DISUSUN OLEH:**  
**ANASTASIA ADINDA NARENDRA INDRIANTO**  
**103112400085**  
**S1 IF-12-01**  
**DOSEN:**  
**Dimas Fanny Hebrasianto Permadi, S.ST., M.KOM**

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## I. DASAR TEORI

### 1. Pengertian Bahasa Pemrograman Golang

Golang adalah bahasa pemrograman yang memiliki sejumlah kelebihan yang tidak dimiliki bahasa pemrograman yang lainnya. Hadirnya bahasa pemrograman Go Language (Golang) semakin dirasakan oleh para pengembang. Tidak heran jika banyak orang yang mulai belajar bahasa pemrograman yang satu ini.

Golang merupakan bahasa pemrograman yang dibuat Google dan tujuannya untuk menyempurnakan bahasa pemrograman yang ada, seperti C, Python dan yang lainnya. Golang bisa jadi pilihan yang tepat saat membuat aplikasi baru.

### 2. Pengertian Input dan Output

- i. *Input* atau masukan adalah data yang diberikan ke dalam program. Masukan ini bisa berasal dari berbagai sumber, seperti pengguna melalui keyboard, mouse, atau suara, dan juga bisa berasal dari sensor atau perangkat lain yang terhubung ke komputer. Dalam pemrograman, input diperlakukan sebagai bahan baku yang akan diproses oleh program.
- ii. *Output* atau keluaran adalah hasil yang diproduksi oleh program setelah mengolah input. Output bisa berbentuk tampilan pada layar, cetakan pada printer, suara, ataupun penyimpanan data ke file. Inti dari program yang kita tulis sebenarnya adalah menghasilkan output yang bermakna dari input yang diberikan.

### 3. Pengertian Tipe Data

Tipe data adalah kategori data yang akan diproses oleh sebuah program komputer. Dengan tipe data, program dapat mengetahui cara menangani data dengan benar, seperti melakukan operasi matematika pada angka atau memanipulasi string untuk tujuan tertentu. Secara sederhana, pengertian tipe data adalah kategori data yang akan diproses oleh sebuah program komputer. Dengan tipe data, program dapat mengetahui cara menangani data dengan benar, seperti melakukan operasi matematika pada angka atau memanipulasi string untuk tujuan tertentu. Contoh paling sederhana dari tipe data adalah tipe data **integer** yang digunakan untuk menyimpan angka bulat atau tipe data **string** yang digunakan untuk menyimpan rangkaian karakter. Penggunaan tipe data yang benar dalam suatu program memastikan bahwa data diolah secara tepat dan mengurangi risiko kesalahan atau bug dalam program tersebut.

### 4. Fungsi Tipe Data

- i. **Menentukan Jenis Nilai:** Tipe data memberi tahu program jenis nilai yang akan disimpan dalam variabel. Misalnya, jika kita ingin menyimpan angka, kita menggunakan tipe data integer atau float, sedangkan untuk menyimpan teks, kita menggunakan tipe data string. Ini membantu program memahami bagaimana cara menangani data tersebut.
- ii. **Efisiensi Penggunaan Memori:** Setiap tipe data memerlukan jumlah memori yang berbeda. Jika kita memilih tipe data yang tepat, program bisa menggunakan memori lebih efisien. Misalnya, menggunakan tipe data yang lebih kecil untuk angka yang tidak terlalu besar akan menghemat ruang di memori.
- iii. **Menjamin Konsistensi Data:** Dengan menentukan tipe data, kita memastikan bahwa variabel hanya bisa menyimpan jenis nilai yang sesuai. Misalnya, jika kita mendeklarasikan variabel sebagai tipe integer, program tidak akan bisa memasukkan teks atau jenis data lainnya ke dalamnya. Ini membantu menjaga agar data tetap konsisten dan sesuai dengan yang diharapkan.
- iv. **Memudahkan Operasi pada Data:** Tipe data juga menentukan operasi apa saja yang bisa dilakukan pada data. Misalnya, kita bisa melakukan perhitungan

matematika pada angka, tetapi kita tidak bisa melakukan hal yang sama pada teks. Jadi, dengan menentukan tipe data yang tepat, kita bisa melakukan operasi yang sesuai dengan jenis data yang kita miliki.

## 5. Pengertian If-Else dan Fungsinya

If-else adalah struktur percabangan dalam pemrograman yang digunakan untuk mengeksekusi kode berdasarkan suatu kondisi. Jika kondisi dalam if bernilai true, maka blok kode di dalamnya akan dijalankan. Jika tidak, program akan memeriksa kondisi dalam else if sebagai alternatif. Jika semua kondisi false, maka else akan dijalankan sebagai pilihan terakhir. Dengan menggunakan if-else, program dapat mengambil keputusan dan menjalankan instruksi yang sesuai berdasarkan kondisi yang diberikan.

If-else berfungsi untuk mengontrol alur program dengan mengeksekusi kode berdasarkan suatu kondisi. If digunakan untuk memeriksa apakah suatu kondisi bernilai true, jika ya, maka blok kode di dalamnya akan dijalankan. Jika tidak, program dapat menggunakan else-if untuk mengevaluasi beberapa kondisi tambahan secara berurutan. Jika semua kondisi false, maka else akan dieksekusi sebagai pilihan terakhir. Selain itu, terdapat if-else bersarang, yang memungkinkan pengecekan kondisi di dalam kondisi lain untuk menangani keputusan yang lebih kompleks.

## 6. Pengertian While Loop dan Fungsinya

While loop adalah metode perulangan yang mengeksekusi blok kode selama kondisi yang diberikan bernilai true dan akan berhenti ketika kondisi berubah menjadi false. Perulangan ini sangat berguna dalam kasus di mana jumlah iterasi belum diketahui secara pasti, seperti membaca input hingga valid atau menjalankan suatu proses hingga syarat tertentu terpenuhi. Fungsinya adalah;

- i. **Menjalankan Perulangan Berdasarkan Kondisi** – While loop memastikan suatu proses terus berjalan selama kondisi bernilai **true**.
- ii. **Mengatasi Iterasi yang Tidak Diketahui Jumlahnya** – Digunakan ketika jumlah perulangan tidak bisa ditentukan sejak awal, seperti menunggu input yang valid.
- iii. **Mengoptimalkan Kontrol Program** – Membantu mengelola eksekusi kode agar hanya berjalan saat kondisi tertentu terpenuhi, meningkatkan efisiensi program.

## 7. Pengertian Fungsi dan Manfaatnya

Fungsi merupakan rangkaian instruksi yang menghasilkan suatu nilai dengan memetakan input ke output tertentu. Dalam pemrograman, suatu subprogram dikategorikan sebagai fungsi apabila memiliki deklarasi tipe nilai yang dikembalikan dan menggunakan kata kunci return dalam tubuhnya. Fungsi digunakan dalam berbagai situasi, seperti menetapkan nilai ke suatu variabel melalui assignment, menjadi bagian dari suatu ekspresi, atau digunakan sebagai argumen dalam subprogram lain. Karena fungsi selalu menghasilkan nilai, penamaannya sebaiknya bersifat deskriptif dan mencerminkan hasil yang diberikan, seperti *median*, *rerata*, *nilaiTerbesar*, atau *selesai*.

Function memungkinkan programmer membagi kode menjadi segmen-segmen kecil yang lebih terkelola, masing-masing melakukan bagian tertentu dari tugas yang lebih besar. Tidak hanya membantu dalam mengorganisasi kode secara lebih efisien, hal ini juga memudahkan pemeliharaan dan pengujian kode.

## II. GUIDED

### 1. Guide 1

#### Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085 S1IF-12-01
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:

```
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Guided\Guide1.go"
5 10
30240
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Guided\Guide1.go"
10 20
670442572800
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> █
```

### Deskripsi Program:

Program Guide1.go dengan bahasa Go lang dibuat untuk menghitung permutasi dari dua bilangan yang dimasukkan oleh pengguna. Setelah menerima input, program membandingkan kedua bilangan dan memastikan bilangan yang lebih besar digunakan sebagai n, sedangkan yang lebih kecil sebagai r, untuk memastikan perhitungan permutasi valid.

Perhitungan permutasi dilakukan menggunakan rumus  $P(n,r)=n!(n-r)/P(n, r)!$ , di mana faktorial dihitung menggunakan fungsi faktorial(n). Fungsi permutasi(n, r) akan mengembalikan hasil perhitungan sesuai rumus, kecuali jika  $r > n$ , maka hasilnya adalah 0. Hasil perhitungan permutasi kemudian ditampilkan ke layar menggunakan fmt.Println().

## 2. Guide 2

### Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085 S1IF-12-01
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))
    }
}

```

### Output:

```

PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Guided\Guide2.go"
Masukkan jumlah data: 4
Masukkan suhu dalam Celsius:
18
20
22
24
Suhu dalam Fahrenheit:
64.40
68.00
71.60
75.20
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3>

```

### Deskripsi Program:

Program Guide2.go dengan Bahasa Go dibuat dengan tujuan untuk mengonversi suhu dari Celsius ke Fahrenheit. Pengguna diminta untuk memasukkan jumlah data suhu yang ingin dikonversi. Jika input yang diberikan tidak valid, seperti bukan angka atau angka negatif, program akan menampilkan pesan kesalahan dan berhenti. Setelah jumlah data ditentukan, pengguna diminta untuk memasukkan suhu dalam Celsius yang kemudian disimpan dalam sebuah array.

Setelah semua data suhu dalam Celsius dimasukkan, program akan mengonversinya ke Fahrenheit menggunakan fungsi `celsiusToFahrenheit`, yang menerapkan rumus konversi  $F = 9.0/5.0C + 32F$ . Hasil konversi kemudian ditampilkan dalam format dua angka di belakang desimal. Dengan demikian, program ini memudahkan pengguna dalam mengubah sejumlah nilai suhu dari Celsius ke Fahrenheit secara efisien dan akurat.

### 3. Guide 3

#### Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085 S1IF-12-01
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:

```
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Guided\Guide3.go"
Masukkan jari-jari tabung: 7
Masukkan tinggi tabung: 18
=====
Luas Permukaan Tabung: 1099.56 satuan²
Volume Tabung: 2770.88 satuan³
=====
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3>
```

### Deskripsi Program:

Program Guide3.go dengan bahasa Go lang digunakan untuk menghitung luas permukaan dan volume tabung berdasarkan jari-jari dan tinggi yang dimasukkan oleh pengguna. Dalam prosesnya, program meminta input untuk jari-jari dan tinggi tabung, kemudian memvalidasi apakah input tersebut berupa angka yang benar. Jika pengguna memasukkan nilai yang tidak valid atau angka negatif, program akan menampilkan pesan kesalahan dan berhenti.

Perhitungan luas permukaan dilakukan dengan rumus  $2\pi r(r+t)$  sedangkan volume dihitung menggunakan rumus  $\pi r^2 t$ . Fungsi luasPermukaanTabung dan volumeTabung digunakan untuk mengolah perhitungan tersebut dengan bantuan library math. Setelah proses perhitungan selesai, hasil luas permukaan dan volume ditampilkan dalam format dua angka di belakang desimal untuk memastikan ketelitian. Program ini memastikan perhitungan dilakukan dengan benar dan memudahkan pengguna dalam mendapatkan hasil dengan cara yang interaktif.



### III. UNGUIDED

#### 1. Unguided 1

##### Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085 S1IF-12-01
package main

import "fmt"

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

// Fungsi untuk menghitung permutasi
func hitungPermutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

// Fungsi untuk menghitung kombinasi
func hitungKombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}

func main() {
    var x, y, p, q int

    // Membaca input
    fmt.Scan(&x, &y, &p, &q)

    // Validasi input
    if x >= p && y >= q {
        fmt.Println(hitungPermutasi(x, p), hitungKombinasi(x, p))
        fmt.Println(hitungPermutasi(y, q), hitungKombinasi(y, q))
    } else {
        fmt.Println("Input kurang tepat")
    }
}
```

## Output:

```
PS D:\Semester 2\ALPRO 2\103112400085_MODULE3> go run "d:\Semester 2\ALPRO 2\103112400085_MODULE3\Unguided\Unguided1.go"
5 10 3 10
60 10
3628800 1
PS D:\Semester 2\ALPRO 2\103112400085_MODULE3> go run "d:\Semester 2\ALPRO 2\103112400085_MODULE3\Unguided\Unguided1.go"
7 14 21 28
Input kurang tepat
PS D:\Semester 2\ALPRO 2\103112400085_MODULE3>

PS D:\Semester 2\ALPRO 2\103112400085_MODULE3> go run "d:\Semester 2\ALPRO 2\103112400085_MODULE3\Unguided\Unguided1.go"
8 0 2 0
56 28
1 1
PS D:\Semester 2\ALPRO 2\103112400085_MODULE3> Go to Line/Column
```

## Deskripsi Program:

Program Unguided1.go dengan bahasa Go lang digunakan untuk menghitung permutasi dan kombinasi berdasarkan empat bilangan yang dimasukkan oleh pengguna. Dua bilangan pertama digunakan sebagai himpunan total ( $n$ ), dan dua bilangan berikutnya sebagai jumlah elemen yang dipilih ( $r$ ). Jika nilai  $n$  lebih besar atau sama dengan  $r$ , program akan menghitung rumus  $P(n,r) = \frac{n!}{(n-r)!}$  dan kombinasi dengan rumus  $C(n,r) = \frac{n!}{r!(n-r)!}$ , lalu menampilkan hasilnya.

Perhitungan dilakukan dengan menggunakan fungsi faktorial, hitungPermutasi, dan hitungKombinasi untuk memastikan efisiensi dan keakuratan hasil. Jika input tidak valid, karena  $x \geq p \ \&\& \ y \geq q$  harus terpenuhi sekaligus. Jika salah satu gagal, perhitungan tidak dilakukan. Solusinya, validasi tiap pasangan ( $x, p$ ) dan ( $y, q$ ) secara terpisah. Dengan validasi ini, program memastikan bahwa perhitungan hanya dilakukan jika kondisi memenuhi syarat, sehingga hasil yang diperoleh tetap benar dan sesuai dengan konsep matematika.

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  $d$ .

## 2. Unguided 2

### Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085 S1IF-12-01

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 D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Unguided\Unguided2.go"
5 5 5
16
24
10
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Unguided\Unguided2.go"
3 8 4
4
63
5
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Unguided\Unguided2.go"
7 2 10
36
3
65
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> █
```

### Deskripsi Program:

Program Unguided2.go dengan bahasa Go lang dibuat untuk melakukan komposisi tiga fungsi, yaitu  $fx(x) = x^2$ ,  $gx(x) = x - 2$ , dan  $hx(x) = x + 1$ . Program membaca tiga bilangan bulat sebagai input, lalu menghitung hasil dari  $fx(gx(hx(a)))$ ,  $gx(hx(fx(b)))$ , dan  $hx(fx(gx(c)))$  dengan menjalankan fungsi terdalam terlebih dahulu. Hasil akhirnya ditampilkan dalam tiga baris sebagai output.

Masukan terdiri dari sebuah bilangan bulat a, b dan c yang dipisahkan oleh spasi. Keluaran terdiri dari tiga baris. Baris pertama adalah (fogoh) (a), baris kedua (gohof)(b), dan baris ketiga adalah (hogof)(c).

### 3. Unguided 3

#### Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085 S1IF-12-01
package main

import (
    "fmt"
    "math"
)

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

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

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

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

    diLingkaran1 := dalamLingkaran(cx1, cy1, r1, x, y)
    diLingkaran2 := dalamLingkaran(cx2, cy2, r2, x, y)

    if diLingkaran1 && diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

```
}  
}
```

### Output:

```
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Unguided\Unguided3.go"  
1 1 5  
8 8 4  
2 2  
Titik di dalam lingkaran 1  
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Unguided\Unguided3.go"  
1 2 3  
4 5 6  
7 8  
Titik di dalam lingkaran 2  
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Unguided\Unguided3.go"  
5 10 15  
-15 4 20  
0 0  
Titik di dalam lingkaran 1 dan 2  
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> go run "d:\Semester 2\ALPRO 2\103112400085_MODUL3\Unguided\Unguided3.go"  
1 1 5  
8 8 4  
15 20  
Titik di luar lingkaran 1 dan 2  
PS D:\Semester 2\ALPRO 2\103112400085_MODUL3> 
```

### Deskripsi Program:

Program Unguided3.go dengan bahasa Go dibuat untuk menentukan apakah suatu titik berada di dalam salah satu atau kedua lingkaran yang diberikan. Jarak antara titik  $(x,y)$  dan pusat lingkaran  $(cx,cy)$  dihitung menggunakan rumus:  $\text{jarak} = \sqrt{(x-cx)^2 + (y-cy)^2}$ . Jika jarak tersebut lebih kecil atau sama dengan jari-jari lingkaran, maka titik berada di dalam lingkaran. Program membaca input untuk dua lingkaran dan satu titik, lalu mencetak hasil berdasarkan posisi titik relatif terhadap lingkaran-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 pusat sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius di representasikan dengan bilangan bulat. Keluaran berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 2", "Titik di lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2".

#### IV. KESIMPULAN

- **Guided**

1. **Perhitungan Permutasi**

- Menghitung permutasi dua bilangan dengan rumus  $P(n, r) = n! / (n-r)!$ .
- Validasi input memastikan perhitungan hanya dilakukan jika  $n \geq r$ .

2. **Konversi Suhu**

- Mengonversi suhu dari Celsius ke Fahrenheit dengan rumus  $F = (9/5 * C) + 32$ .
- Memastikan input valid sebelum menampilkan hasil.

3. **Luas dan Volume Tabung**

- Menghitung luas permukaan ( $2\pi r(r + t)$ ) dan volume ( $\pi r^2 t$ ).
- Menggunakan fungsi terpisah untuk akurasi perhitungan.

- **Unguided**

1. **Perhitungan Permutasi**

- Menghitung permutasi dua bilangan dengan rumus  $P(n, r) = n! / (n-r)!$  dan  $C(n, r) = n! / (r!(n-r)!)$ .
- Perhitungan dilakukan hanya jika  $n \geq r$

2. **Komposisi Fungsi**

- Menghitung hasil dari komposisi tiga fungsi matematika  $f(x)$ ,  $g(x)$ ,  $h(x)$ .
- Menampilkan hasil dalam tiga baris sesuai aturan fungsi.

3. **Posisi Titik terhadap Lingkaran**

- Menentukan apakah titik berada dalam lingkaran dengan rumus jarak akar( $(x-cx)^2 + (y-cy)^2$ ).
- Hasil menunjukkan apakah titik di dalam satu, dua atau diluar lingkaran

## **V. REFRENSI**

<https://sko.dev/wiki/input-dan-output>

<https://codingstudio.id/blog/golang-adalah/>

<https://dif.telkomuniversity.ac.id/tipe-data-pemrograman/>

<https://rpubs.com/maulidarahmah/829044>

[https://repository.unikom.ac.id/62967/1/Materi%20Pertemuan%204\\_Labview%201%2BWhile%20Loop%20%2B%20Shift%20Register.pdf](https://repository.unikom.ac.id/62967/1/Materi%20Pertemuan%204_Labview%201%2BWhile%20Loop%20%2B%20Shift%20Register.pdf)

<https://www.revou.co/kosakata/function>

<https://sko.dev/wiki/fungsi>