

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL III  
FUNCTION**



**Disusun Oleh:**  
Bayu Kuncoro Adi / 2311102031  
S1 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

Fungsi adalah kumpulan blok kode yang diberi nama untuk mempermudah penggunaannya. Dengan menerapkan fungsi dengan baik, kode menjadi lebih modular dan mengikuti prinsip DRY (Don't Repeat Yourself). Artinya, kita tidak perlu menulis ulang kode yang sama berkali-kali, cukup mendefinisikan fungsi tersebut sekali, lalu memanggilnya saat diperlukan.

Suatu subprogram dikatakan fungsi apabila:

- Ada deklarasi tipe nilai yang dikembalikan.
- Terdapat kata kunci return dalam badan subprogram.

Maka fungsi digunakan jika suatu nilai biasanya diperlukan, seperti:

- Assignment nilai ke suatu variabel.
- Bagian dari ekspresi
- Bagian dari argumen suatu subprogram, dsb.

### 1. Deklarasi Function

	Notasi Algoritma
1	function <nama function> (<params>) -> <type>
2	kamus
3	{deklarasi variabel lokal dari fungsi}
4	...
5	algoritma
6	{badan algoritma fungsi}
7	...
8	return <value/variabel>
9	endfunction
	Notasi dalam bahasa Go
10	func <nama function> (<params>) <type> {
11	/* deklarasi variabel lokal dari fungsi */
12	...
13	/* badan algoritma fungsi*/
14	...
15	return <value/variabel>
16	
17	}

	Notasi Algoritma
1	function volumeTabung(jari_jari,tinggi : integer) -> real
2	kamus
3	luasAlas, volume: real
4	algoritma
5	luasAlas <- 3.14 * (jari_jari * jari_jari)
6	volume <- luasAlas * tinggi
7	return volume
8	endfunction
	Notasi dalam bahasa Go
10	func volumeTabung(jari_jari,tinggi int) float64 {
11	var luasAlas,volume float64
12	luasAlas = 3.14 * float64(jari_jari * jari_jari)
13	volume = luasAlas * tinggi
14	return volume
15	}

## 2. Cara Pemanggilan Function

Sama halnya dengan prosedur, pemanggilan fungsi cukup dilakukan dengan penulisan nama fungsi beserta argument yang diminta oleh parameter dari fungsi. Perbedaannya dengan prosedur adalah fungsi bisa di-assign ke suatu variabel, menjadi bagian dari ekspresi, dan argument dari suatu subprogram.

	Notasi Algoritma
1	program ContohProsedur
2	kamus
3	r,t : integer
4	v1,v2 : real
5	algoritma
6	r <- 5;
7	t <- 10
8	v1 <- volumeTabung(r,t) <span style="float: right;">{cara pemanggilan #1}</span>
9	v2 <- volumeTabung(r,t) + volumeTabung(15,t) <span style="float: right;">{cara pemanggilan #2}</span>
10	output(volumeTabung(14,100)) <span style="float: right;">{cara pemanggilan #3}</span>
11	endprogram
	Notasi dalam bahasa Go
12	func main() {
13	var r,t int
14	var v1,v2 float64
15	r = 5
16	t = 10
17	v1 = volumeTabung(r,t) <span style="float: right;">// cara pemanggilan #1</span>
18	v2 = volumeTabung(r,t) + volumeTabung(15,t) <span style="float: right;">// cara pemanggilan #2</span>
19	fmt.Println(volumeTabung(14,100)) <span style="float: right;">// cara pemanggilan #3</span>
20	}

### 3. Contoh Program dengan Function

Buatlah sebuah program beserta fungsi bilangan bulat yang menyatakan nilai a permutasi b apabila  $a \geq b$  atau b permutasi a untuk kemungkinan yang lain.

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)
}
```

Program di atas menghitung permutasi dua angka yang dimasukkan oleh pengguna. Program meminta pengguna untuk memasukkan dua bilangan bulat, 'a' dan 'b'. Jika nilai 'a' lebih besar atau sama dengan 'b', maka fungsi 'permutasi(a, b)' akan dipanggil, jika sebaliknya maka fungsi 'permutasi(b, a)' yang dipanggil. Fungsi 'permutasi(n, r)' menghitung permutasi menggunakan rumus  $P(n, r) = \frac{n!}{(n-r)!}$ , di mana 'faktorial(n)' adalah fungsi yang menghitung nilai faktorial dari angka 'n'. Program ini kemudian mencetak hasil perhitungan permutasi tersebut ke layar.

## II. GUIDED

### 1. Contoh program dengan Function

```
package main

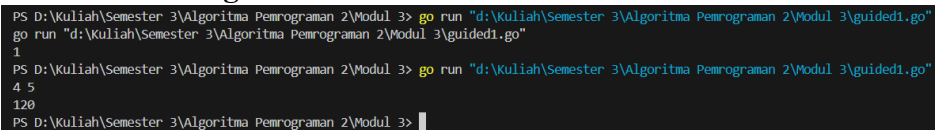
import "fmt"

func main() {
    var a, b int
    fmt.Scan(&a, &b) // Membaca input dari pengguna untuk dua nilai
    integer a dan b
    if a >= b {
        // Jika a lebih besar atau sama dengan b, panggil fungsi
        permutasi dengan parameter (a, b)
        fmt.Println(permutasi(a, b))
    } else {
        // Jika b lebih besar dari a, panggil fungsi permutasi dengan
        parameter (b, a)
        fmt.Println(permutasi(b, a))
    }
}

func faktorial(n int) int {
    var hasil int = 1
    var i int
    // Loop untuk menghitung faktorial dari n
    for i = 1; i <= n; i++ {
        hasil = hasil * i // Mengalikan hasil dengan nilai i pada
        setiap iterasi
    }
    return hasil // Mengembalikan hasil faktorial
}

func permutasi(n, r int) int {
    // Menghitung permutasi nPr dengan membagi faktorial n dengan
    faktorial (n-r)
    return faktorial(n) / faktorial(n-r)
}
```

### Screenshoot Program



```
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\guided1.go"
go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\guided1.go"
1
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\guided1.go"
4 5
120
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3>
```

## Deskripsi dan Cara Kerja Program

Program di atas menghitung permutasi dua angka yang dimasukkan oleh pengguna. Program meminta pengguna untuk memasukkan dua bilangan bulat, `a` dan `b`. Jika nilai `a` lebih besar atau sama dengan `b`, maka fungsi `permutasi(a, b)` akan dipanggil, jika sebaliknya maka fungsi `permutasi(b, a)` yang dipanggil. Fungsi `permutasi(n, r)` menghitung permutasi menggunakan rumus  $P(n, r) = \frac{n!}{(n-r)!}$ , di mana `faktorial(n)` adalah fungsi yang menghitung nilai faktorial dari angka `n`. Program ini kemudian mencetak hasil perhitungan permutasi tersebut ke layar.

## 2. Menhitung Luas dan Keliling Persegi

```
package main

import (
    "fmt"
)

func main() {
    var sisi float64

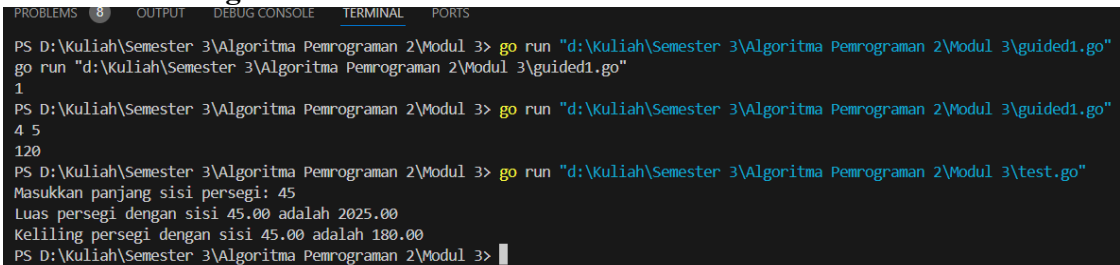
    fmt.Print("Masukkan panjang sisi persegi: ")
    fmt.Scanln(&sisi)

    luas := sisi * sisi

    keliling := 4 * sisi

    fmt.Printf("Luas persegi dengan sisi %.2f adalah %.2f\n", sisi,
luas)
    fmt.Printf("Keliling persegi dengan sisi %.2f adalah %.2f\n",
sisi, keliling)
}
```

## Screenshoot Program



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\guided1.go"
go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\guided1.go"
1
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\guided1.go"
4 5
120
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\test.go"
Masukkan panjang sisi persegi: 45
Luas persegi dengan sisi 45.00 adalah 2025.00
Keliling persegi dengan sisi 45.00 adalah 180.00
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> |
```

**Deskripsi dan Cara Kerja Program**

Program di atas menghitung luas dan keliling sebuah persegi berdasarkan panjang sisi yang dimasukkan oleh pengguna. Program pertama-tama meminta pengguna untuk memasukkan nilai panjang sisi persegi ('sisi'), lalu menghitung luas persegi dengan rumus  $\text{luas} = \text{sisi} \times \text{sisi}$  dan keliling persegi dengan rumus  $\text{keliling} = 4 \times \text{sisi}$ . Hasil perhitungan luas dan keliling tersebut kemudian ditampilkan di layar dengan format dua angka desimal.

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

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

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	

```
function factorial(n: integer) → integer
{mengembalikan nilai faktorial dari n}

function permutation(n, r : integer) → integer
{Mengembalikan hasil n permutasi r, dan n >= r}

function combination(n, r : integer) → integer
{Mengembalikan hasil n kombinasi r, dan n >= r}
```

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung faktorial
func factorial(n int) int {
    if n == 0 {
        return 1
    }
    result := 1
```



```

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

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

// Fungsi untuk menghitung kombinasi
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 4 bilangan: ")
    fmt.Scan(&a, &b, &c, &d)

    // Baris pertama: permutasi dan kombinasi a terhadap c
    p1 := permutation(a, c)
    c1 := combination(a, c)
    fmt.Printf("%d %d\n", p1, c1)

    // Baris kedua: permutasi dan kombinasi b terhadap d
    p2 := permutation(b, d)
    c2 := combination(b, d)
    fmt.Printf("%d %d\n", p2, c2)
}

```

### Screenshoot Program

```

PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\unguided1.go"
Masukkan 4 bilangan: 3 4 5 6
6 0
24 0
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3>

```

### Deskripsi dan Cara Kerja Program

Program di atas menghitung permutasi dan kombinasi dari dua pasang bilangan yang dimasukkan oleh pengguna. Pengguna diminta untuk memasukkan empat bilangan bulat: `a`, `b`, `c`, dan `d`. Program kemudian menghitung dan menampilkan permutasi dan kombinasi dari `a` terhadap `c` (menggunakan fungsi `permutation` dan `combination`), serta dari `b` terhadap `d`. Fungsi `permutation(n, r)` menghitung

permutasi menggunakan rumus  $P(n, r) = \frac{n!}{(n-r)!}$ , dan fungsi `combination(n,

r)` menghitung kombinasi menggunakan rumus  $C(n, r) = \frac{n!}{r!(n-r)!}$ . Hasil permutasi dan kombinasi ditampilkan dalam dua baris: baris pertama untuk `a` dan `c`, baris kedua untuk `b` dan `d`.

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	$(f \circ g \circ h)(5) = 4$ $(g \circ h \circ f)(5) = 63$ $(h \circ f \circ g)(5) = 5$

```
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
}

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

    // Menerima input nilai a dan menghitung f(g(h(a)))
    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)
    fmt.Println("Hasil (f \circ g \circ h)(a) =", f(g(h(a))))

    // Menerima input nilai b dan menghitung g(h(f(b)))
    fmt.Print("Masukkan nilai b: ")
```

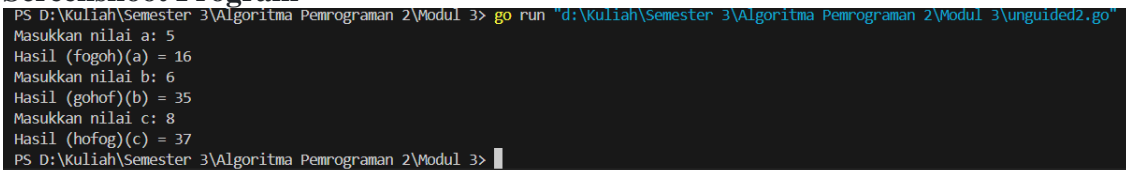
```

    fmt.Scan(&b)
    fmt.Println("Hasil (gohof) (b) =", g(h(f(b))))

    // Menerima input nilai c dan menghitung h(f(g(c)))
    fmt.Print("Masukkan nilai c: ")
    fmt.Scan(&c)
    fmt.Println("Hasil (hofog) (c) =", h(f(g(c))))
}

```

### Screenshoot Program



```

PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\unguided2.go"
Masukkan nilai a: 5
Hasil (fogoh)(a) = 16
Masukkan nilai b: 6
Hasil (gohof)(b) = 35
Masukkan nilai c: 8
Hasil (hofog)(c) = 37
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3>

```

### Deskripsi dan Cara Kerja Program

Program di atas menghitung komposisi tiga fungsi matematika:  $f(x) = x^2$ ,  $g(x) = x - 2$ , dan  $h(x) = x + 1$ , berdasarkan nilai input yang diberikan oleh pengguna. Program meminta pengguna memasukkan tiga bilangan bulat: `a`, `b`, dan `c`. Untuk setiap nilai, program menghitung hasil komposisi fungsi secara berurutan:  $f(g(h(a)))$  (dihitung dengan cara memanggil fungsi  $h(a)$ ), lalu  $g$ , kemudian  $f$ ),  $g(h(f(b)))$ , dan  $h(f(g(c)))$ . Hasil dari tiap komposisi fungsi tersebut kemudian ditampilkan kepada pengguna.

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".

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3	Titik di dalam lingkaran 2
	4 5 6 7 8	
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

Fungsi jarak menghitung jarak titik  $(a, b)$  dan  $(c, d)$  Dimana rumus jarak adalah:

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

```
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
}
```

```
// Fungsi untuk mengecek posisi titik terhadap dua lingkaran
func checkPosition(cx1, cy1, r1, cx2, cy2, r2, x, y float64) string {
    inCircle1 := didalam(cx1, cy1, r1, x, y)
    inCircle2 := didalam(cx2, cy2, r2, x, y)

    if inCircle1 && inCircle2 {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if inCircle1 {
        return "Titik di dalam lingkaran 1"
    } else if inCircle2 {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}

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

    // Masukkan data sesuai dengan format yang diberikan
    fmt.Println("Masukkan koordinat pusat dan radius lingkaran
1:")
    fmt.Scan(&cx1, &cy1, &r1)

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

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

    // Cek posisi titik sembarang
    result := checkPosition(cx1, cy1, r1, cx2, cy2, r2, x, y)
    fmt.Println(result)
}
```

## Screenshoot Program

```
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> go run "d:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3\unguided3.go"
Masukkan koordinat pusat dan radius lingkaran 1:
5
6
4
Masukkan koordinat pusat dan radius lingkaran 2:
4
1
2
Masukkan koordinat titik sembarang:
2
3
Titik di luar lingkaran 1 dan 2
PS D:\Kuliah\Semester 3\Algoritma Pemrograman 2\Modul 3> 4
```

### **Deskripsi dan Cara Kerja Program**

Program di atas digunakan untuk mengecek posisi suatu titik sembarang terhadap dua lingkaran yang diberikan. Program meminta pengguna untuk memasukkan koordinat pusat dan radius dari dua lingkaran, serta koordinat titik sembarang. Fungsi `jarak` digunakan untuk menghitung jarak antara dua titik, sementara fungsi `didalam` mengecek apakah titik berada di dalam lingkaran dengan menggunakan rumus jarak antara titik dan pusat lingkaran. Fungsi `checkPosition` kemudian menentukan apakah titik tersebut berada di dalam lingkaran pertama, lingkaran kedua, keduanya, atau di luar kedua lingkaran. Hasilnya kemudian dicetak ke layar untuk memberi tahu posisi titik tersebut relatif terhadap lingkaran-lingkaran yang diberikan.

## **KESIMPULAN**

Fungsi dalam pemrograman adalah alat penting untuk membuat kode lebih modular dan menghindari pengulangan. Penggunaan fungsi seperti faktorial, permutasi, dan kombinasi membantu menyederhanakan perhitungan matematika yang kompleks seperti menghitung permutasi atau kombinasi angka, sehingga memudahkan implementasi perhitungan tersebut dalam program. Selain itu, konsep fungsi juga diterapkan dalam menghitung komposisi fungsi matematika sederhana, seperti  $f(x)$ ,  $g(x)$ , dan  $h(x)$ , yang memungkinkan perhitungan komposisi dilakukan dengan cara yang lebih terstruktur dan efisien.

Pada bagian lain, penggunaan fungsi dalam penghitungan geometris seperti menentukan posisi titik sembarang terhadap dua lingkaran menunjukkan bagaimana fungsi dapat diterapkan untuk menyelesaikan masalah spasial. Dengan memanfaatkan fungsi untuk menghitung jarak dan mengevaluasi posisi titik, program dapat secara akurat menentukan apakah sebuah titik berada di dalam atau di luar lingkaran. Ini memperlihatkan fleksibilitas dan keandalan penggunaan fungsi dalam berbagai jenis perhitungan, baik matematis maupun geometris, yang membuat kode lebih mudah dipahami dan dipelihara.

## **DAFTAR PUSTAKA**

Modul 3 Praktikum Algoritma dan Pemrograman 2