

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL III**

**FUNGSI**



Oleh:

NAMA : ARNANDA SETYA NOSA PUTRA

NIM : 2311102042

KELAS : IF 11 02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## I. DASAR TEORI

Fungsi adalah sekumpulan instruksi yang menghasilkan suatu nilai, biasanya dengan memetakan input ke nilai lain. Oleh karena itu, fungsi selalu mengembalikan hasil. Suatu subprogram dianggap sebagai fungsi jika:

1. Mendeklarasikan tipe data yang akan dikembalikan, dan
2. Memiliki kata kunci `return` di dalam tubuh subprogram.

Fungsi digunakan ketika suatu nilai dibutuhkan, seperti:

- Menetapkan nilai ke variabel
- Sebagai bagian dari ekspresi
- Sebagai argumen dalam subprogram, dan lain-lain.

Karena itu, nama fungsi sebaiknya mencerminkan nilai yang dihasilkan, misalnya menggunakan kata benda atau kata sifat. Contoh nama fungsi: median, rerata, nilaiTerbesar, ditemukan, selesai, dan sebagainya.

- Deklarasi fungsi

Deklarasi fungsi sama dengan prosedur, yaitu berada pada blok yang terpisah

Notasi Algoritma
<pre>function &lt;nama kamus     {deklarasi function&gt;(&lt;params&gt;) -&gt; &lt;type&gt;     ... algoritma     {badan algoritma fungsi} ...     return &lt;value/variabel&gt; endfunction</pre>
Notasi dalam bahasa Go
<pre>func &lt;nama function&gt;(&lt;params&gt;) &lt;type&gt; {  /* deklarasi variabel lokal dari fungsi */  ...  /* badan algoritma fungsi */  ...  return &lt;value/variabel&gt;  }</pre>

## Cara Pemanggilan Function

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

Notasi Algoritma	
1	program KAMUS    ContohProsedur
2	r,t : integer v1,v2 : real
3	algoritma r <- 5; t <- 10 v1 <- volumeTabung(r,t)
4	v2 <- volumeTabung(r,t)    volumeTabung(15,t) {cara    #1} output(volumeTabung(14,100))    {cara    #2} endprogram    {cara    #3}
Notasi dalam bahasa Go	
	func main() { var r,t int var v1,v2 float64 5 10 v1    volumeTabung(r,t)    // cara    #1  v2 = volumeTabung(r,t) + volumeTabung(15,t) fmt.Println(volumeTabung(14,100))    // cara    #2 }

Pada contoh pemanggilan fungsi di atas terlihat tidak ada perbedaan pada saat pemanggilan fungsi pada pseudocode ataupun Golang. Di sini terlihat fungsi bisa di-assign ke suatu variabel pada saat pemanggilan, bisa dioperasikan sesuai dengan tipe data yang dikembalikan, dan juga bisa langsung ditampilkan dengan perintah output ataupun print.

## II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

### SOAL

NO 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 {
    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))
}
```

Output :

```
PS C:\Users\HP> go run "C:\Users\HP\A
5 10
30240
PS C:\Users\HP> □
```

Penjelasan :

Program di atas menghitung permutasi dari dua bilangan yang dimasukkan oleh pengguna. Pertama, program membaca dua nilai integer, yaitu `a` dan `b`. Kemudian, jika `a` lebih besar atau sama dengan `b`, program memanggil fungsi `permutasi(a, b)`, dan sebaliknya memanggil `permutasi(b, a)` jika tidak. Fungsi `permutasi` menggunakan fungsi `faktorial` untuk menghitung nilai permutasi dengan rumus:

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

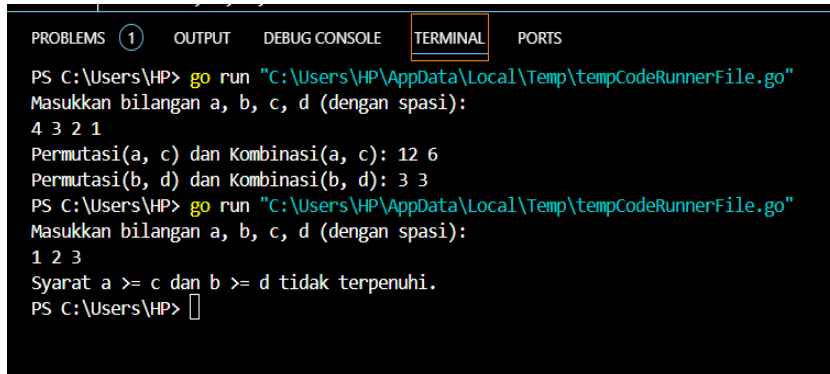
di mana faktorial dihitung dalam fungsi `faktorial`.

## GUIDED

NO 2.

```
1 package main
2 import "fmt"
3
4 func factorial(n int) int {
5     if n == 0 || n == 1 {
6         return 1
7     }
8     result := 1
9     for i := 2; i <= n; i++ {
10        result *= i
11    }
12    return result
13 }
14
15 func permutation(n, r int) int {
16     return factorial(n) / factorial(n-r)
17 }
18
19 func combination(n, r int) int {
20     return factorial(n) / (factorial(r) * factorial(n-r))
21 }
22
23 func main() {
24     var a, b, c, d int
25     fmt.Println("Masukkan bilangan a, b, c, d (dengan spasi): ")
26     fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)
27
28     if a >= c && b >= d {
29         permutasiAC := permutation(a, c)
30         kombinasiAC := combination(a, c)
31
32         permutasiBD := permutation(b, d)
33         kombinasiBD := combination(b, d)
34
35         fmt.Println("Permutasi(a, c) dan Kombinasi(a, c):", permutasiAC, kombinasiAC)
36         fmt.Println("Permutasi(b, d) dan Kombinasi(b, d):", permutasiBD, kombinasiBD)
37     } else {
38         fmt.Println("Syarat a >= c dan b >= d tidak terpenuhi.")
39     }
40 }
```

Output :

A screenshot of a terminal window with a dark background. The terminal shows the execution of a Go program. The first run takes input '4 3 2 1' and outputs 'Permutasi(a, c) dan Kombinasi(a, c): 12 6' and 'Permutasi(b, d) dan Kombinasi(b, d): 3 3'. The second run takes input '1 2 3' and outputs an error message: 'Syarat a >= c dan b >= d tidak terpenuhi.' The terminal tabs at the top are 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and highlighted with a yellow box), and 'PORTS'.

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan bilangan a, b, c, d (dengan spasi):
4 3 2 1
Permutasi(a, c) dan Kombinasi(a, c): 12 6
Permutasi(b, d) dan Kombinasi(b, d): 3 3
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan bilangan a, b, c, d (dengan spasi):
1 2 3
Syarat a >= c dan b >= d tidak terpenuhi.
PS C:\Users\HP>
```

Penjelasan :

Program di atas adalah sebuah aplikasi sederhana dalam bahasa Go yang menghitung permutasi dan kombinasi dari dua pasang angka yang diinput oleh pengguna. Fungsi `factorial` digunakan untuk menghitung faktorial suatu bilangan, sedangkan fungsi `permutation` dan `combination` menggunakan faktorial untuk menghitung permutasi dan kombinasi. Dalam fungsi `main`, program meminta pengguna memasukkan empat bilangan (a, b, c, d). Jika syarat `a >= c` dan `b >= d` terpenuhi, program akan menghitung dan menampilkan permutasi dan kombinasi dari (a, c) dan (b, d). Jika syarat tidak terpenuhi, program menampilkan pesan kesalahan.

### III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

#### SOAL 3

[Lingkaran] 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".

*Source code :*

```
1 package main
2
3 import "fmt"
4 import "math"
5
6 func main() {
7     var cx1, cy1, r1 int
8     var cx2, cy2, r2 int
9     var x, y int
10
11     fmt.Scan(&cx1, &cy1, &r1)
12     fmt.Scan(&cx2, &cy2, &r2)
13     fmt.Scan(&x, &y)
14
15     if cekDalamLingkaran(x, y, cx1, cy1, r1) && cekDalamLingkaran(x, y, cx2, cy2, r2) {
16         |   fmt.Println("Titik di dalam lingkaran 1 dan 2")
17     } else if cekDalamLingkaran(x, y, cx1, cy1, r1) {
18         |   fmt.Println("Titik di dalam lingkaran 1")
19     } else if cekDalamLingkaran(x, y, cx2, cy2, r2) {
20         |   fmt.Println("Titik di dalam lingkaran 2")
21     } else {
22         |   fmt.Println("Titik di luar lingkaran 1 dan 2")
23     }
24 }
25
26 func cekDalamLingkaran(x, y, cx, cy, r int) bool {
27     |   var jarak float64
28     |   jarak = math.Sqrt(float64((x-cx)*(x-cx) + (y-cy)*(y-cy)))
29     |   return jarak <= float64(r)
30 }
31
```

Output :

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempC
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempC
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempC
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempC
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
PS C:\Users\HP> 
```

Penjelasan :

Program ini bertujuan untuk menentukan posisi suatu titik dalam kaitannya dengan dua lingkaran yang berbeda. Pertama, program membaca input untuk pusat dan jari-jari kedua lingkaran, serta koordinat titik yang akan diperiksa. Kemudian, dengan menggunakan fungsi `cekDalamLingkaran`, program menghitung jarak antara titik dan pusat setiap lingkaran. Berdasarkan hasil perhitungan, program mencetak apakah titik tersebut berada di dalam kedua lingkaran, hanya di dalam salah satu lingkaran, atau di luar keduanya. Fungsi `cekDalamLingkaran` menghitung jarak menggunakan rumus jarak Euclidean dan membandingkannya dengan jari-jari lingkaran untuk menentukan posisi titik.



