

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 3  
FUNCTION**



Oleh:

Rifki Nur Wibowo

21102308

IF-11-07

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## **I. DASAR TEORI**

Fungsi adalah blok kode atau sekumpulan instruksi yang dirancang untuk menyelesaikan tugas tertentu dan dapat dipanggil kapan saja dalam program. Fungsi membantu memisahkan logika program menjadi bagian-bagian yang lebih kecil dan terorganisir, yang dapat diakses berulang kali tanpa menuliskan ulang kode yang sama.

## II. GUIDED

Program

```
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

A screenshot of a code editor's output window. The window has a dark background. At the top, there is a toolbar with a dropdown arrow, a button labeled 'Run', a button with a speech bubble icon labeled 'Ask AI', a text indicator '9s • 4 minutes ago', a checkmark icon, and a small upward arrow icon. Below the toolbar, the output text is displayed in a light color. It shows the input '2 5' on the first line and the output '20' on the second line.

```
2 5
20
```

### III. UNGUIDED

#### 1. Program

```
package main

import "fmt"

func factorial(n int) int {
    if n == 0 {
        return 1
    }
    return n * factorial(n-1)
}

func permutation(n, r int) int {
    if n < r {
        return 0
    }
    return factorial(n) / factorial(n-r)
}

func combination(n, r int) int {
    if n < r {
        return 0
    }
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func main() {
    var a, b, c, d int
    fmt.Println("Masukkan:")
    fmt.Scanln(&a, &b, &c, &d)

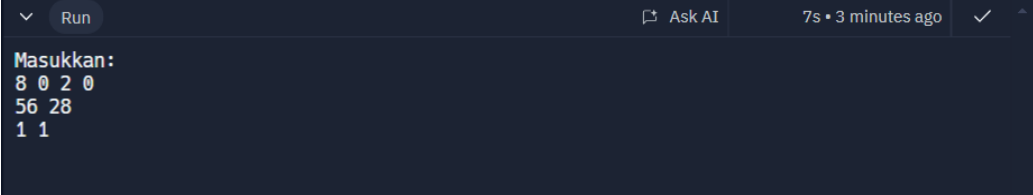
    if a < c || b < d {
```

```

        fmt.Println("Syarat tidak terpenuhi. a harus lebih besar
atau sama dengan c, dan b harus lebih besar atau sama dengan d.")
        return
    }
    fmt.Println(permutation(a, c), combination(a, c))
    fmt.Println(permutation(b, d), combination(b, d))
}

```

## Output



```

Masukkan:
8 0 2 0
56 28
1 1

```

## 2. Program

```

package main

import "fmt"

func f(x int) int {
    return x * x
}

func g(x int) int {
    return x - 2
}

func h(x int) int {
    return x + 1
}

func fogoh(a int) int {
    return f(g(h(a)))
}

func gohof(b int) int {
    return g(h(f(b)))
}

```

```

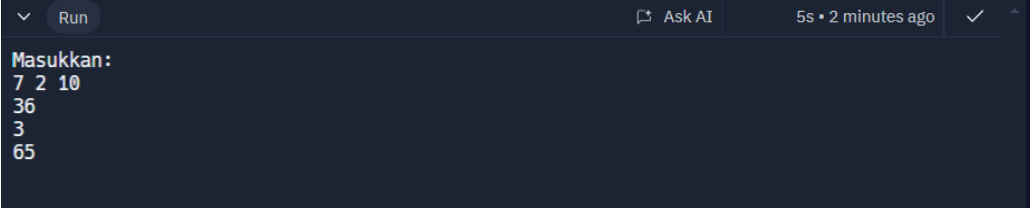
func hofog(c int) int {
    return h(f(g(c)))
}

func main() {
    var a, b, c int
    fmt.Println("Masukkan:")
    fmt.Scanln(&a, &b, &c)

    fmt.Println(fogoh(a))
    fmt.Println( gohof(b))
    fmt.Println( hofog(c))
}

```

## Output



```

Masukkan:
7 2 10
36
3
65
3
65

```

## 3. Program

```

package main

import (
    "fmt"
    "math"
)

func jarak(ax, ay, bx, by float64) float64 {
    return math.Sqrt(math.Pow(ax-bx, 2) + math.Pow(ay-by, 2))
}

func diDalamLingkaran(cx, cy, r, x, y float64) bool {
    return jarak(cx, cy, x, y) <= r
}

func main() {
    var cx1, cy1, r1 float64
    fmt.Print()
    fmt.Scan(&cx1, &cy1, &r1)
}

```

```
var cx2, cy2, r2 float64
fmt.Print()
fmt.Scan(&cx2, &cy2, &r2)

var x, y float64
fmt.Print()
fmt.Scan(&x, &y)

dalamLingkaran1 := diDalamLingkaran(cx1, cy1, r1, x, y)
dalamLingkaran2 := diDalamLingkaran(cx2, cy2, r2, x, y)

if dalamLingkaran1 && dalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if dalamLingkaran1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if dalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}
```

## Output

```
Run Ask AI 16s • 2 minutes ago ✓
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
```