

# **ALGORITMA PEMOGRAMAN 2**

## **MODUL 5**

### **"REKURSIF"**



**Oleh:**

**NAMA: Davi Ilyas Renaldo**

**NIM: 103112400062**

**KELAS: 12-IF-01**

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## I.DASAR TEORI

Pada modul-modul sebelumnya sudah dijelaskan bahwa suatu subprogram baik fungsi atau prosedur bisa memanggil subprogram lainnya. Hal ini tidak menutup kemungkinan bahwa subprogram yang dipanggil adalah dirinya sendiri. Dalam pemrograman teknik ini dikenal dengan istilah rekursif. Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identik dari masalah utama.

	Notasi Algoritma	Notasi dalam bahasa GO
1	procedure cetak(in x:integer)	func cetak(x int){
2	algoritma	fmt.Println(x)
3	output(x)	cetak(x+1)
4	cetak(x+1)	}
5	endprocedure	

Apabila diperhatikan subprogram **cetak()** di atas, terlihat pada baris ke-4 terdapat pemanggilan subprogram **cetak()** kembali. Misalnya apabila kita eksekusi perintah **cetak(5)** maka akan menampilkan angka 5 6 7 8 9...dst tanpa henti. Artinya setiap pemanggilan subprogram **cetak()** nilai **x** akan selalu bertambah 1 (**increment by one**) secara **terus menerus tanpa henti**.

Rekursif adalah proses pengulangan barang dengan cara yang mirip. Konsep yang sama berlaku dalam bahasa pemrograman juga. Jika suatu program memungkinkan untuk memanggil suatu fungsi di dalam fungsi yang sama, maka itu disebut panggilan fungsi rekursif.

```
1 func recursion() {
2     recursion() /* function calls itself */
3 }
4 func main() {
5     recursion()
6 }
```

Bahasa pemrograman Go mendukung rekursif. Artinya, ini memungkinkan suatu fungsi untuk memanggil dirinya sendiri. Tetapi saat menggunakan rekursif, programmer perlu berhati-hati untuk mendefinisikan kondisi keluar dari fungsi, jika tidak maka akan menjadi loop tak terbatas. Contoh rekursif dalam Go.

Fungsi rekursif sangat berguna untuk menyelesaikan banyak masalah matematika seperti menghitung faktorial suatu angka, menghasilkan seri Fibonacci, dll.

Contoh 1: Menghitung Faktorial Menggunakan Recursion dalam Go

Contoh berikut menghitung faktorial dari angka yang diberikan menggunakan fungsi rekursif –

```
1 package main
2
3 import "fmt"
4
5 func factorial(i int)int {
6     if(i <= 1) {
7         return 1
8     }
9     return i * factorial(i - 1)
10 }
11 func main() {
12     var i int = 15
13     fmt.Printf("Factorial of %d is %d", i, factorial(i))
14 }
```

Ketika kode di atas dikompilasi dan dieksekusi, ia menghasilkan hasil sebagai berikut –

```
1 Factorial of 15 is 2004310016
```

## II. GUIDED

1.

```
package main
import "fmt"
func pangkatIteratif(base, exp int) int {
    hasil := 1

    for i := 0; i < exp; i++ {
        hasil *= base
    }

    return hasil
}
func faktorialIteratif(n int) int {
    hasil := 1
    for i := 2; i <= n; i++ {
        hasil *= i
    }
    return hasil
}
func main() {
    var base, exp, n int

    //Input Pangkat
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exp)

    fmt.Printf("%d^%d = %d\n", base, exp, pangkatIteratif(base, exp))

    //Input Faktorial
    fmt.Print("Masukkan angka untuk faktorial: ")
    fmt.Scanln(&n)

    fmt.Printf("%d! = %d\n", n, faktorialIteratif(n))
}
```

### OUTPUT:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\GUIDED 5\1.go"
Masukkan bilangan: 4
Masukkan pangkat: 3
4^3 = 64
Masukkan angka untuk faktorial: 6
6! = 720
PS C:\Users\ACER\OneDrive\alpro2\modul1> █
```

### DESKRIPSI:

Program ini adalah program untuk menghitung hasil perpangkatan dan faktorial dengan metode iteratif. Pengguna diminta untuk memasukkan angka bilangan, pangkat dan faktorial. Maka program ini akan berjalan menghitung hasil dari angka yang dimasukkan oleh pengguna.

2.

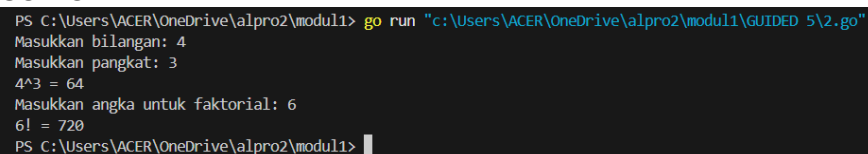
```
package main
import "fmt"
func pangkatRekursif(base, exp int) int {
    if exp == 0 {
        return 1
    }
    return base * pangkatRekursif(base, exp-1)
}
func faktorialRekursif(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * faktorialRekursif(n-1)
}
func main() {
    var base, exp, n int
    // Input pangkat
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exp)

    fmt.Printf("%d^%d = %d\n", base, exp, pangkatRekursif(base, exp))

    // Input faktorial
    fmt.Print("Masukkan angka untuk faktorial: ")
    fmt.Scanln(&n)

    fmt.Printf("%d! = %d\n", n, faktorialRekursif(n))
}
```

#### OUTPUT:



```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\GUIDED 5\2.go"
Masukkan bilangan: 4
Masukkan pangkat: 3
4^3 = 64
Masukkan angka untuk faktorial: 6
6! = 720
PS C:\Users\ACER\OneDrive\alpro2\modul1>
```

#### DESKRIPSI:

Program ini adalah program untuk menghitung hasil perpangkatan dan faktorial dengan metode rekursif. Pengguna diminta untuk memasukkan angka bilangan, pangkat dan faktorial. Maka program ini akan berjalan menghitung hasil dari angka yang dimasukkan oleh pengguna.

### III. UNGUIDED

1.

```
// M.DAVI ILYAS RENALDO
// 103112400062
package main
import "fmt"
func bintang(n, x int) {
    if x > n {
        return
    }
    for i := 0; i < x; i++ {
        fmt.Print("*")
    }
    fmt.Println()

    bintang(n, x+1)
}

func main() {
    var n int
    fmt.Scan(&n)

    bintang(n, 1)
}
```

#### OUTPUT:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 5\1.go"
5
*
**
***
****
*****
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 5\1.go"
1
*
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 5\1.go"
3
*
**
***
```

#### DESKRIPSI:

Program ini adalah program yang menghasilkan atau membentuk pola segitiga siku-siku dengan metode rekursif, yang tingginya ditentukan oleh pengguna berdasarkan angka yang dimasukkan oleh pengguna.

2.

```
// M.DAVI ILYAS RENALDO
// 103112400062
package main
import "fmt"
func fakRekursif(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    fakRekursif(n, i+1)
}

func main() {
    var n int
    fmt.Scan(&n)
    fakRekursif(n, 1)
    fmt.Println()
}
```

### OUTPUT:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 5\2.go"
5
1 5
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 5\2.go"
12
1 2 3 4 6 12
PS C:\Users\ACER\OneDrive\alpro2\modul1> |
```

### DESKRIPSI:

Program ini adalah suatu program yang mengimplementasikan fungsi rekursif untuk mencari dan mencetak faktor-faktor dari suatu bilangan bulat positif yang dimasukkan oleh pengguna.

3.

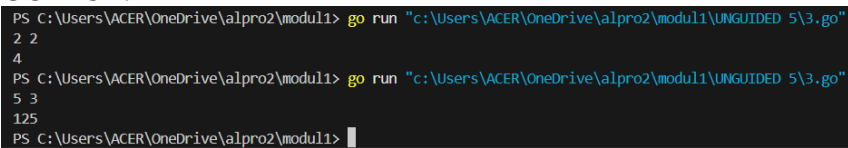
```
// M.DAVI ILYAS RENALDO
// 103112400062
package main

import "fmt"

func pangkatRekursif(x, y int) int {
    if y == 0 {
        return 1
    }
    return x * pangkatRekursif(x, y-1)
}

func main() {
    var x, y int
    fmt.Scan(&x, &y)
    hasil := pangkatRekursif(x, y)
    fmt.Println(hasil)
}
```

#### OUTPUT:



```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 5\3.go"
2 2
4
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 5\3.go"
5 3
125
PS C:\Users\ACER\OneDrive\alpro2\modul1>
```

#### DESKRIPSI:

Program ini adalah program yang menghitung hasil pemangkatan suatu bilangan rekursif. Pengguna akan diminta untuk memasukkan sepasang bilangan bulat (x,y) dimana (y) adalah pangkat dari (x). Maka program ini akan menghitung hasil perpangkatan sesuai angka yang sudah dimasukkan oleh pengguna.

#### **IV.KESIMPULAN**

#### **V.REFERENSI**

- modul 5 REKURSIF praktikum algoritma & pemograman 2
- <https://kursuswebprogramming.com/recursion-dalam-bahasa-golang/>