

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 5
REKURSIF**



**Telkom
University**

Oleh:

M.HANIF AL FAIZ

103112400042

12IF-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Dasar Teori Modul 6: Rekursif

Rekursif adalah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah. Teknik ini digunakan untuk menyelesaikan masalah yang bisa dipecah menjadi sub-masalah yang serupa dengan masalah utama. Dalam rekursi, terdapat dua komponen utama yang harus ada, yaitu:

1. **Base-case:** Kondisi yang menentukan kapan proses rekursif harus berhenti. Tanpa base-case, rekursif akan berjalan tanpa henti dan menyebabkan stack overflow. Base-case merupakan komponen penting yang harus diketahui terlebih dahulu saat membuat program rekursif.
2. **Recursive-case:** Bagian yang memanggil kembali fungsi itu sendiri. Recursive-case ini adalah kebalikan dari base-case dan akan terus berjalan hingga base-case tercapai.

Contoh program rekursif dalam modul ini termasuk menghitung nilai faktorial, deret Fibonacci, dan pangkat. Rekursi juga dapat digunakan untuk menyelesaikan masalah lain seperti menampilkan angka dalam urutan tertentu atau mencari faktor dari sebuah angka.

Penting untuk memahami dasar teori rekursif, karena teknik ini menawarkan cara yang elegan dan efisien dalam memecahkan masalah yang memiliki pola berulang atau dapat dipecah menjadi sub-masalah yang lebih kecil.

II. GUIDED

1.

```
package main

import "fmt"

// Fungsi iteratif untuk menghitung pangkat (base^exp)
func pangkatIteratif(base, exp int) int {
    hasil := 1
    for i := 0; i < exp; i++ {
        hasil *= base
    }
    return hasil
}

// Fungsi iteratif untuk menghitung faktorial (n!)
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 D:\10311240042_MODUL5> go run d:\10311240042_MODUL5\cosol1.go
Masukkan bilangan: 5
Masukkan pangkat: 2
5^2 = 25
Masukkan angka untuk faktorial: 5
5! = 120
```

DESKRIPSI:

PROGRAM YANG MENJALANKAN PROSES MATEMATIKA TERDIRI DARI EKSPONENSIAL DAN FAKTORIAL YANG DIMANA FUNGSI INI MENGGUNAKAN FOR LOOP DI DALAM PROGRAMNYA

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
    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))
    fmt.Print("masukkan angka untuk faktorial: ")
    fmt.Scanln(&n)

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

```
}
```

OUTPUT:

```
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\cosol2.go
masukkan bilangan:
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\cosol2.go
masukkan bilangan: 6
masukkan pangkat: 6
6^6 = 46656
masukkan angka untuk faktorial: 6
6! = 720
```

DESKRIPSI:

SAMA SEPERTI DESKRIPSI PROGRAM YANG PERTAMA YANG DIMANA MENJALANKAN PROGRAM EXPONEN DAN FAKTORIAL NAMU YANG MEMBEDAKAN ADALAH FUNGSI INI MENGGUNAKAN FUNGSI REKURSIF YANG DIMANA FUNGSI BERIKUT MEMANGGIL DIRINYA SENDIRI

III. UNGUIDED

1.

```
// M.HANIF AL FAIZ
// 103112400042
package main

import "fmt"

func fibonacci(n int) int {
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    } else {
        return fibonacci(n-1) + fibonacci(n-2)
    }
}

func main() {
    for i := 0; i <= 10; i++ {
        fmt.Printf("Fibonacci(%d) = %d\n", i, fibonacci(i))
    }
}
```

OUTPUT:

```
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol1.go
Fibonacci(0) = 0
Fibonacci(1) = 1
Fibonacci(2) = 1
Fibonacci(3) = 2
Fibonacci(4) = 3
Fibonacci(5) = 5
Fibonacci(6) = 8
Fibonacci(7) = 13
Fibonacci(8) = 21
Fibonacci(9) = 34
Fibonacci(10) = 55
```

DESKRIPSI:

FUNGSI REKURSIF YANG MENJALANKAN PROGRAM YANG MENCETAK
DERET FIBONACCI

2.

```
// M.HANIF AL FAIZ
// 103112400042
package main

import (
    "fmt"
)

func printStars(n int, current int) {
    if current > n {
        return
    }

    for i := 0; i < current; i++ {
        fmt.Print("*")
    }
    fmt.Println()
    printStars(n, current+1)
}

func main() {
    var N int
    fmt.Print("Masukkan nilai N: ")
    fmt.Scan(&N)
    printStars(N, 1)
}
```

OUTPUT:

```
Masukkan nilai N: 5
*
**
***
****
*****

PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol2.go
Masukkan nilai N: 3
*
**
***

PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol2.go
Masukkan nilai N: 1
*
```

DESKRIPSI:

PROGRAM YANG MENCETAK BINTANG YANG SESUAI DENGAN NILAI-N YANG KITA MASUKKAN

3.

```
package main

import (
    "fmt"
)

func printFactors(n int, current int) {
    if current > n {
        return
    }
    if n%current == 0 {
        fmt.Print(current, " ")
    }
    printFactors(n, current+1)
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    fmt.Printf("Faktor dari %d: ", N)
    printFactors(N, 1)
    fmt.Println()
}
```

OUTPUT:

```
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol3.go
Masukkan bilangan bulat positif N: 5
Faktor dari 5: 1 5
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol3.go
Masukkan bilangan bulat positif N: 12
Faktor dari 12: 1 2 3 4 6 12
```

DESKRIPSI:

PROGRAM YANG MENGIMPLEMENTASIKAN REKURSIF UNTUK
MENAMPILKAN FAKTOR BILANGAN SUATU N, ATAU BILANGAN APA SAJA
YANG HABIS MEMBAGI N

4.

```
// M.HANIF AL FAIZ
// 103112400042
package main

import "fmt"

func printSequence(n int) {
    fmt.Print(n, " ")
    if n > 1 {
        printSequence(n - 1)
    }
    if n != 1 {
        fmt.Print(n, " ")
    }
}

func main() {
    var N int
    fmt.Print("Masukkan sebuah bilangan bulat positif: ")
    fmt.Scan(&N)
    printSequence(N)
}
```

OUTPUT:

```
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol4.go
Masukkan sebuah bilangan bulat positif: 5
5 4 3 2 1 2 3 4 5
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol4.go
Masukkan sebuah bilangan bulat positif: 9
9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
```

DESKRIPSI:

program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu yang dimana masukannya terdiri dari sebuah bilangan positif N dan keluarannya barisan bilangan dari N hingga 1 dan kembali ke N.

5.

```
// M.HANIF AL FAIZ
// 103112400042
package main

import (
    "fmt"
)

func cetakBilanganGanjil(n int, current int) {
    if current > n {
        return
    }
    if current%2 != 0 {
        fmt.Print(current, " ")
    }
    cetakBilanganGanjil(n, current+1)
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    fmt.Printf("Bilangan ganjil dari 1 hingga %d: ", N)
    cetakBilanganGanjil(N, 1)
    fmt.Println()
}
```

OUTPUT:

```
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol5.go
Masukkan bilangan bulat positif N: 5
Bilangan ganjil dari 1 hingga 5: 1 3 5
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol5.go
Masukkan bilangan bulat positif N: 20
Bilangan ganjil dari 1 hingga 20: 1 3 5 7 9 11 13 15 17 19
```

DESKRIPSI:

program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

6.

```
// M.HANIF AL FAIZ
```

```
// 103112400042
package main

import (
    "fmt"
)

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

func main() {
    var x, y int
    fmt.Print("Masukkan bilangan bulat x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan bilangan bulat y: ")
    fmt.Scan(&y)

    hasil := pangkat(x, y)
    fmt.Printf("%d dipangkatkan %d = %d\n", x, y, hasil)
}
```

OUTPUT:

```
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol6.go
Masukkan bilangan bulat x: 2
Masukkan bilangan bulat y: 2
2 dipangkatkan 2 = 4.00
PS D:\103112400042_MODUL5> go run d:\103112400042_MODUL5\latsol6.go
Masukkan bilangan bulat x: 5
Masukkan bilangan bulat y: 3
5 dipangkatkan 3 = 125.00
```

DESKRIPSI:

program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan.

IV. KESIMPULAN

Laporan ini menekankan pentingnya rekursi dalam menyelesaikan masalah yang melibatkan tugas berulang, seperti perhitungan matematika dan pembangkitan urutan angka. Rekursi dibandingkan dengan pendekatan iteratif,

menunjukkan bagaimana rekursi dapat menyederhanakan implementasi dan meningkatkan kemampuan pemecahan masalah dalam situasi tertentu.

Praktikum ini membantu memahami rekursi dan penerapannya dalam menyelesaikan masalah umum dalam ilmu komputer

V. REFERENSI

Telkom University. (n.d.). *Modul Praktikum Algoritma dan Pemrograman*
Telkom University