

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 5
MATERI REKURSIF**



Disusun Oleh:

NAMA: MULIA AKBAR NANDA PRATAMA

NIM: 103112400034

KELAS: 12 IF 01

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025/2026**

I. DASAR TEORI

Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub masalah yang identic dari masalah utama. Teknik rekursif ini merupakan salah satu alternatif untuk menggantikan struktur control perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur). Base case adalah kondisi proses rekursif berhenti. Base case merupakan hal terpenting dan pertama yang harus diketahui Ketika akan membuat program rekursif. Recursive case adalah kondisi di mana proses pemanggilan dirinya sendiri dilakukan. Kondisi recursive case adalah komplemen atau negasi dari base case. Algoritma rekursif terdiri dari dua komponen utama yaitu base case (basis), yaitu bagian untuk mengentikan proses rekursif dan menjadi komponen terpenting dalam sebuah rekursif, dan yang kedua ada recursive case, yaitu bagian pemanggilan subprogramnya.

II. GUIDED' SOURCE CODE

```
C: > VSCODE > 103112400034_MODUL5 > 103112400034_GUIDED 1 > -go g1.go > ...  
1  // MULIA AKBAR NANDA PRATAMA  
2  // 103112400034  
3  package main  
4  
5  import "fmt"  
6  
7  func pangkatIteratif(base, exp int) int {  
8      hasil := 1  
9      for i := 0; i < exp; i++ {  
10         |   hasil *= base  
11     }  
12     return hasil  
13 }  
14  
15 func faktorialIteratif(n int) int {  
16     hasil := 1  
17     for i := 2; i <= n; i++ {  
18         |   hasil *= i  
19     }  
20     return hasil  
21 }  
22  
23 func main() {  
24     var base, exp, n int  
25  
26     fmt.Print("masukkan bilangan: ")  
27     fmt.Scanln(&base)  
28     fmt.Print("masukkan pangkat: ")  
29     fmt.Scanln(&exp)  
30  
31     fmt.Printf("%d^%d = %d\n", base, exp, pangkatIteratif(base, exp))  
32  
33     fmt.Print("masukkan angka untuk faktorial: ")  
34     fmt.Scanln(&n)  
35  
36     fmt.Printf("%d! = %d\n", n, faktorialIteratif(n))  
37 }  
38
```

```
PS C:\Users\pratama> go run "c:\VSCODE\103112400034_MODUL5\103112400034_GUIDED 1\g1.go"  
masukkan bilangan: 5  
masukkan pangkat: 2  
5^2 = 25  
masukkan angka untuk faktorial: 5  
5! = 120  
PS C:\Users\pratama> |
```

Deskripsi: program di atas merupakan program yang dibuat dengan Bahasa Go. Program tersebut akan berjalan Ketika user memasukkan input berupa bilangan bulat. Program tersebut akan

menghitung pangkat dari sebuah bilangan dan factorial dari sebuah angka dan menggunakan metode iterative yang di mana menggunakan loop untuk melakukan suatu proses berulang kali sampai kondisi tertentu terpenuhi.

```

C: > VSCODE > 103112400034_MODUL5 > 103112400034_GUIDED 2 > go g2.go > faktorialRekursif
1  // MULIA AKBAR NANDA PRATAMA
2  // 103112400034
3  package main
4
5  import "fmt"
6
7  func pangkatRekursif(base, exp int) int {
8
9      if exp == 0 {
10         return 1
11     }
12     return base * pangkatRekursif(base, exp-1)
13 }
14
15 func faktorialRekursif(n int) int {
16
17     if n == 0 || n == 1 {
18         return 1
19     }
20     return n * faktorialRekursif(n-1)
21 }
22
23 func main() {
24     var base, exp, n int
25     fmt.Print("masukkan bilangan: ")
26     fmt.Scanln(&base)
27     fmt.Print("masukkan pangkat: ")
28     fmt.Scanln(&exp)
29
30     fmt.Printf("%d^%d = %d\n", base, exp, pangkatRekursif(base, exp))
31     fmt.Print("masukkan angka untuk faktorial: ")
32     fmt.Scanln(&n)
33
34     fmt.Printf("%d! = %d\n", n, faktorialRekursif(n))
35 }

```

```

PS C:\Users\pratama> go run "c:\VSCODE\103112400034_MODUL5\103112400034_GUIDED 2\g2.go"
masukkan bilangan: 5
masukkan pangkat: 2
5^2 = 25
masukkan angka untuk faktorial: 10
10! = 3628800
PS C:\Users\pratama>

```

Deskripsi: program di atas merupakan program yang dibuat dalam bahasa Go. Program tersebut digunakan untuk menghitung pangkat dari sebuah bilangan dan factorial dari sebuah angka dan menggunakan metode rekursif yang di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan suatu masalah.

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

```
un1.go > ...
1 // MULIA AKBAR NANDA PRATAMA
2 // 103112400034
3 package main
4
5 import "fmt"
6
7 func fibonacci(n int) int {
8     if n <= 1 {
9         return n
10    }
11    return fibonacci(n-1) + fibonacci(n-2)
12 }
13
14 func main() {
15     var n int
16     fmt.Print("n: ")
17     fmt.Scanln(&n)
18
19     fmt.Print("Sn: ")
20     for i := 0; i <= 10; i++ {
21         fmt.Printf("%d ", fibonacci(i))
22     }
23     fmt.Println()
24 }
```

```
PS C:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED 1> go run "c:\VSCODE\103112400034_UNGUIDED\un1.go"
n: 10
Sn: 0 1 1 2 3 5 8 13 21 34 55
PS C:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED 1>
```

Deskripsi: program di atas merupakan program yang dibuat dalam Bahasa Go. Program tersebut mengimplementasikan fungsi rekursif untuk menghasilkan deret Fibonacci hingga sukuk e 10. Deret Fibonacci adalah urutan angka yang di mana setiap angka setelah dua angka pertama adalah jumlah dari dua angka sebelumnya. Dua angka pertama dalam deret Fibonacci adalah 0 dan 1.

```

GO un2.go > cetakBintang

1  // MULIA AKBAR NANDA PRATAMA
2  // 103112400034
3  package main
4
5  import "fmt"
6  func main() {
7      |   var n int
8      |   fmt.Print("masukkan: ")
9      |   fmt.Scan(&n)
10     |
11     |   cetakBintang(n)
12 }
13 func cetakBintang(n int) {
14     |   if n == 0 {
15     |   |   return
16     |   }
17     |   cetakBintang(n - 1)
18     |   for i := 0; i < n; i++ {
19     |   |   fmt.Print("*")
20     |   }
21     |   fmt.Println()
22 }
23

```

```

PS C:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED 2> go run "c:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED\un2.go"
masukkan: 3
*
**
***

PS C:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED 2> go run "c:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED\un2.go"
masukkan: 4
*
**
***
****

PS C:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED 2>

```

Deskripsi: program di atas merupakan program yang dibuat dalam Bahasa Go. Program di atas dibuat untuk menghasilkan pola Bintang segitiga siku-siku dengan menggunakan fungsi rekursif. Pola Bintang yang dihasilkan memiliki jumlah baris

yang ditentukan oleh input pengguna. Setiap baris memiliki jumlah Bintang yang sesuai dengan nomor barisnya.


```

1 // MULIA AKBAR NANDA PRATAMA
2 // 103112400034
3 package main
4
5 import "fmt"
6 func cariFaktor(n int, faktor int) {
7     if faktor > n {
8         return
9     }
10    if n%faktor == 0 {
11        fmt.Printf("%d ", faktor)
12    }
13    cariFaktor(n, faktor+1)
14 }
15
16 func main() {
17     var n int
18     fmt.Print("N: ")
19     fmt.Scan(&n)
20
21     fmt.Printf("Faktor dari %d adalah: ", n)
22     cariFaktor(n, 1)
23     fmt.Println()
24 }

```

```

PS C:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED 3> go run "c:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED\un3.go"
N: 5
Faktor dari 5 adalah: 1 5
PS C:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED 3> go run "c:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED\un3.go"
N: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS C:\VSCODE\103112400034_MODUL5\103112400034_UNGUIDED 3>

```

Deskripsi: program di atas merupakan program yang dibuat menggunakan Bahasa Go. Program dibuat untuk menemukan dan menampilkan semua faktor dari bilangan bulat positif yang dimasukkan oleh pengguna. Faktor-faktor tersebut adalah bilangan-bilangan yang habis membagi bilangan yang dimasukkan. Program ini menggunakan fungsi rekursif.

IV. KESIMPULAN

Konsep rekursif untuk metode penyelesaian masalah dalam pemrograman, khususnya untuk Bahasa Go. Rekursif digunakan sebagai alternatif dari perulangan untuk menyelesaikan berbagai persoalan komputasi, seperti hitung factorial, pangkat bilangan, deret Fibonacci, pola Bintang, dan mencari faktor bilangan. Dari berbagai implementasi yang dilakukan, dapat disimpulkan bahwa rekursif memiliki keunggulan dalam perjanjian kode dan meningkatkan keterbacaan program. Namun, penggunaannya harus mempertimbangkan base case untuk menghindari rekursi tak terbatas yang dapat menyebabkan stack overflow.

Secara keseluruhan, praktikum ini memberikan pemahaman mendalam tentang rekursif dan penerapannya dalam pemrograman, sekaligus memperkuat keterampilan dalam menggunakan bahasa Go untuk menyelesaikan berbagai permasalahan secara efisien.

V. REFERENSI

Modul 5 Praktikum Algoritma dan Pemrograman 2