

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6  
REKURSIF**



**Disusun Oleh :**

**Annasya Maulafidatu Zahra / 2311102246**

**S1-IF-11-05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom., M.Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Rekursif adalah konsep penting dalam pemrograman yang melibatkan sebuah fungsi yang memanggil dirinya sendiri untuk menyelesaikan suatu masalah. Dalam bahasa pemrograman Go (Golang), rekursif digunakan untuk menyederhanakan kode dan membuatnya lebih mudah dipahami, terutama dalam menyelesaikan masalah-masalah yang memiliki sifat berulang. Ciri utama dari fungsi rekursif adalah adanya pengulangan, di mana fungsi tersebut memanggil dirinya sendiri, serta adanya kondisi keluar yang jelas untuk mencegah terjadinya infinite loop.

Struktur dasar dari fungsi rekursif dalam Golang biasanya terdiri dari dua bagian utama: kondisi akhir dan ekspresi rekursif. Kondisi akhir adalah titik di mana fungsi akan berhenti memanggil dirinya sendiri, sedangkan ekspresi rekursif adalah bagian di mana fungsi memanggil dirinya dengan parameter yang telah dimodifikasi. Contohnya, dalam menghitung faktorial, kita dapat mendefinisikan fungsi `factorial(n int)` yang akan mengembalikan 1 jika `n` adalah 0 atau 1, dan jika tidak, akan mengalikan `n` dengan hasil dari `factorial(n - 1)`.

Meskipun rekursif menawarkan banyak keuntungan, seperti membuat kode lebih ringkas dan mudah dibaca, ada juga beberapa kerugian yang perlu diperhatikan. Salah satu risiko utama adalah potensi terjadinya stack overflow jika kondisi keluar tidak ditangani dengan baik. Selain itu, penggunaan rekursif dapat menyebabkan masalah efisiensi, terutama dalam kasus di mana banyak panggilan fungsi dilakukan secara berulang tanpa penyimpanan hasil sementara (memoization). Oleh karena itu, penting bagi pengembang untuk mempertimbangkan kapan harus menggunakan rekursif dan bagaimana cara mengoptimalkannya agar tidak mengorbankan performa aplikasi.

Secara keseluruhan, pemahaman tentang rekursif dalam Golang sangat penting bagi pengembang untuk dapat memanfaatkan teknik ini secara efektif. Dengan memahami cara kerja rekursif dan menerapkannya pada masalah yang tepat, programmer dapat menciptakan solusi yang elegan dan efisien. Namun, selalu penting untuk mengevaluasi kelebihan dan kekurangan dari pendekatan ini agar dapat mencapai hasil terbaik dalam pengembangan perangkat lunak.

## II. GUIDED

### 1. Guided 1

#### Soal Studi Case

Memuat baris bilangan dari n hingga 1

#### Sourcecode

```
package main

import "fmt"

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

func baris(bilangan int) {
    if bilangan == 1 {
        fmt.Println(1)
    } else {
        fmt.Println(bilangan)
        baris(bilangan - 1)
    }
}
```

#### Screenshoot Output

```
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\guided1.go"
12
12
11
10
9
8
7
6
5
4
3
2
1
PS D:\semester 3\Praktikum Alpro 2\modull 6> █
```

#### Deskripsi Program

Kode di atas adalah program Go yang dapat mencetak angka dari n hingga 1 secara menurun dengan menggunakan rekursi. Di dalam main, program meminta input pengguna untuk menentukan nilai n, yang kemudian dikirim sebagai argumen ke fungsi baris. Fungsi baris adalah fungsi rekursif dengan parameter bilangan dan mencetak nilai bilangan pada setiap pemanggilan, lalu memanggil dirinya sendiri dengan argumen bilangan - 1, sehingga nilai bilangan secara bertahap akan menurun sampai mencapai 1.

Terdapat blok if-else dalam fungsi baris yang digunakan untuk menentukan kapan rekursi berhenti. Jika bilangan sama dengan 1, fungsi mencetak angka 1 dan tidak memanggil dirinya lagi, sehingga rekursi berhenti.

## 2. Guided 2

### Soal Studi Case

Menghitung hasil penjumlahan 1 hingga n

### Sourcecode

```
package main

import "fmt"

func penjumlahan(n int) int {
    if n == 1 {
        return 1
    } else {
        return n + penjumlahan(n-1)
    }
}

func main() {
    var n int
    fmt.Scan(&n)
    fmt.Println(penjumlahan(n))
}
```

## Screenshoot Output

```
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\guided2.go"
5
15
PS D:\semester 3\Praktikum Alpro 2\modull 6>
```

## Deskripsi Program

Kode di atas dibuat untuk menghitung jumlah total dari angka 1 sampai n menggunakan fungsi rekursif bernama penjumlahan. Di bagian utama (main), program meminta pengguna memasukkan angka n, lalu memanggil fungsi penjumlahan(n) untuk menghitung hasilnya. Fungsi penjumlahan bekerja dengan cara menambahkan angka n dengan hasil penjumlahan(n-1) sampai akhirnya mencapai angka 1, di mana fungsi berhenti memanggil dirinya sendiri .

Secara sederhana, jika pengguna memasukkan  $n = 4$ , maka fungsi akan menghitung  $4 + 3 + 2 + 1$  secara bertahap. Prosesnya adalah: pertama penjumlahan(4) memanggil penjumlahan(3), lalu penjumlahan(3) memanggil penjumlahan(2), dan seterusnya hingga penjumlahan(1), yang langsung mengembalikan nilai 1. Hasil akhirnya akan dihitung sebagai penjumlahan dari semua angka ini, yang kemudian ditampilkan oleh program.

### 3. Guided 3

#### Soal Studi Case

Mencari 2 pangkat n atau  $2^n$

#### Sourcecode

```
package main

import "fmt"

func pangkat(n int) int {
    if n == 0 {
        return 1
    } else {
        return 2 * pangkat(n-1)
    }
}

func main() {
    var n int
    fmt.Print("masukkan nilai n:")
    fmt.Scanln(&n)
```

```
    fmt.Println("Hasil dari 2 pangkat", n, "adalah :",  
pangkat(n))  
}
```

### Screenshoot Output

```
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\guided3.go"  
masukkan nilai n: 4  
Hasil dari 2 pangkat 4 adalah : 16  
PS D:\semester 3\Praktikum Alpro 2\modull 6> |  
Add Logs Improve Code Version Control Share Code Link Search Error Ln 2
```

### Deskripsi Program

Kode di atas adalah program Go yang menghitung hasil dari 2 pangkat n menggunakan fungsi rekursif pangkat. Di dalam main, program meminta pengguna memasukkan nilai n, lalu memanggil fungsi pangkat(n) untuk menghitung hasilnya dan menampilkannya.

Fungsi pangkat bekerja dengan mengecek apakah n sama dengan 0. Jika iya, fungsi mengembalikan 1 (karena  $2^0 = 1$ )). Jika n lebih besar dari 0, fungsi akan mengalikan 2 dengan hasil dari pangkat(n-1). Fungsi ini akan terus memanggil dirinya sendiri dengan nilai n yang berkurang hingga mencapai 0.

#### 4. Guided 4

##### Soal Studi Case

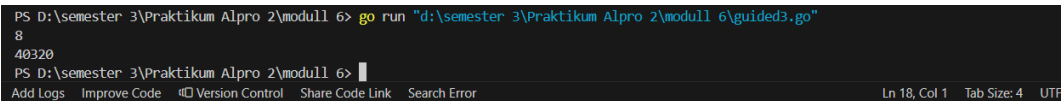
Mencari nilai faktorial atau n!

##### Sourcecode

```
package main  
  
import "fmt"  
  
var n int  
  
func faktorial(n int) int {  
    if n == 0 || n == 1 {  
        return 1  
    } else {  
        return n * faktorial(n-1)  
    }  
}
```

```
}  
func main() {  
    fmt.Scan(&n)  
    fmt.Println(faktorial(n))  
}
```

## Screenshoot Output



```
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\guided3.go"  
8  
40320  
PS D:\semester 3\Praktikum Alpro 2\modull 6> |  
Add Logs Improve Code Version Control Share Code Link Search Error Ln 18, Col 1 Tab Size: 4 UTF
```

## Deskripsi Program

Kode di atas adalah program Go yang menghitung hasil dari  $2^n$  pangkat  $n$  menggunakan fungsi rekursif pangkat. Di dalam main, program meminta pengguna memasukkan nilai  $n$ , lalu memanggil fungsi  $\text{pangkat}(n)$  untuk menghitung hasilnya dan menampilkannya.

Fungsi pangkat bekerja dengan mengecek apakah  $n$  sama dengan 0. Jika iya, fungsi mengembalikan 1 (karena  $2^0 = 1$ )). Jika  $n$  lebih besar dari 0, fungsi akan mengalikan 2 dengan hasil dari  $\text{pangkat}(n-1)$ . Fungsi ini akan terus memanggil dirinya sendiri dengan nilai  $n$  yang berkurang hingga mencapai 0.

## I. UNGUIDED

### 1. Unguided 1

#### Soal Studi Case

Deret fibonacci adalah sebuah deret dengan nilai suku ke-0 dan ke-1 adalah 0 dan 1, dan nilai suku ke-n selanjutnya adalah hasil penjumlahan dua suku sebelumnya. Secara umum dapat diformulasikan  $S_n = S_{n-1} + S_{n-2}$ . Berikut ini adalah contoh nilai deret fibonacci hingga suku ke-10. Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci tersebut

$n$	0	1	2	3	4	5	6	7	8	9	10
$S_n$	0	1	1	2	3	5	8	13	21	34	55

#### Sourcecode

```
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() {
    var n int
    fmt.Print("Masukkan nilai n: ")
    fmt.Scanln(&n)

    fmt.Println("Deret Fibonacci hingga suku ke-", n, ":")
    for i := 0; i <= n; i++ {
        fmt.Print(fibonacci(i), " ")
    }
}
```



## Screenshoot Output

```
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\unguided1.go"
Masukkan nilai n: 10
Deret Fibonacci hingga suku ke- 10 :
0 1 1 2 3 5 8 13 21 34 55
PS D:\semester 3\Praktikum Alpro 2\modull 6>
```

## Deskripsi Program

Kode di atas adalah program Go yang menghitung dan menampilkan deret Fibonacci hingga suku ke-n menggunakan rekursi. Fungsi fibonacci menghitung nilai setiap suku dengan memeriksa dua kondisi dasar: jika n adalah 0, hasilnya 0, dan jika n adalah 1, hasilnya 1. Untuk nilai n yang lebih besar, fungsi menambahkan hasil dari dua suku sebelumnya fibonacci(n-1) + fibonacci(n-2)).

Di dalam main, program meminta input dari pengguna untuk nilai n, lalu menggunakan perulangan for untuk mencetak deret Fibonacci dari suku ke-0 hingga suku ke-n. Setiap suku dihitung dengan memanggil fungsi fibonacci, dan hasilnya ditampilkan berurutan.

## 2. Unguided 2

### Soal Studi Case

Buatlah sebuah program yang digunakan untuk menampilkan pola bintang berikut ini dengan menggunakan fungsi rekursif. N adalah masukan dari user.

No	Masukan	Keluaran
1	5	* ** *** **** *****
2	1	*
3	3	* ** ***

## Sourcecode

```
package main

import "fmt"
```

```

func cetakBintang(jumlah int) {
    if jumlah > 0 {
        fmt.Print("*")
        cetakBintang(jumlah - 1)
    } else {
        fmt.Println()
    }
}

func polaBintang(n, baris int) {
    if baris <= n {
        cetakBintang(baris)
        polaBintang(n, baris+1)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan nilai N: ")
    fmt.Scanln(&n)
    polaBintang(n, 1)
}

```

### Screenshoot Output

```

PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\unguided2.go"
Masukkan nilai N: 6
*
**
***
****
*****
*****
PS D:\semester 3\Praktikum Alpro 2\modull 6>

```

### Deskripsi Program

Kode di atas adalah program Go yang mencetak pola bintang bertingkat menggunakan rekursi. Fungsi `cetakBintang` mencetak sejumlah bintang dalam satu baris, berkurang satu setiap kali dipanggil hingga selesai. Fungsi `polaBintang` mengatur jumlah baris, mulai dari baris pertama hingga baris ke-*n* (input pengguna), dengan memanggil `cetakBintang` untuk mencetak bintang sesuai jumlah baris saat ini. Program ini menghasilkan pola bertingkat di mana jumlah bintang bertambah setiap baris.

### 3. Unguided 3

#### Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu N, atau bilangan yang apa saja yang habis membagi N. Masukan terdiri dari sebuah bilangan bulat positif N. Keluaran terdiri dari barisan bilangan yang menjadi faktor dari N (terurut dari 1 hingga N ya).

No	Masukan	Keluaran
1	5	1 5
2	12	1 2 3 4 6 12

#### Sourcecode

```
package main

import "fmt"

// Fungsi untuk mencetak faktor dari N secara rekursif
func cetakFaktor(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    cetakFaktor(n, i+1)
}

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

    fmt.Printf("Faktor dari %d adalah: ", n)
    cetakFaktor(n, 1)
    fmt.Println()
}
```

## Screenshoot Output

```
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\unguided3.go"
Masukkan bilangan bulat positif N: 10
Faktor dari 10 adalah: 1 2 5 10
PS D:\semester 3\Praktikum Alpro 2\modull 6>
```

## Deskripsi Program

Kode di atas adalah program Go yang mencetak faktor-faktor dari bilangan bulat positif N menggunakan rekursi. Fungsi cetakFaktor memeriksa setiap angka i dari 1 hingga N untuk menentukan apakah angka tersebut adalah faktor dari N. Jika N habis dibagi i maka i adalah faktor, dan program mencetaknya. Fungsi ini kemudian memanggil dirinya sendiri dengan  $i + 1$  hingga mencapai nilai N.

Di dalam main, program meminta pengguna untuk memasukkan nilai N, lalu memanggil cetakFaktor mulai dari 1 untuk mencetak semua faktor dari N.

### 4. Unguided 4

#### Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu. Masukan terdiri dari sebuah bilangan bulat positif N. Keluaran terdiri dari barisan bilangan dari N hingga 1 dan kembali ke N

No	Masukan	Keluaran
1	5	5 4 3 2 1 2 3 4 5
2	9	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9

## Sourcecode

```
package main

import "fmt"

func cetakBarisan(n, current int) {
    if current < 1 {
        return
    }
    fmt.Print(current, " ")
    cetakBarisan(n, current-1)
}
```

```

        if current != n {
            fmt.Print(current, " ")
        }
    }

func main() {
    var n int
    fmt.Scanln(&n)
    cetakBarisan(n, n)
    fmt.Println()
}

```

### Screenshoot Output

```

PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\unguided4.go"
5
5 4 3 2 1 1 2 3 4
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\unguided4.go"
9
9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8
PS D:\semester 3\Praktikum Alpro 2\modull 6>

```

### Deskripsi Program

Program di atas berfungsi untuk menampilkan barisan bilangan dari N hingga 1, kemudian kembali ke N, menggunakan pendekatan rekursif. Program dimulai dengan mendefinisikan fungsi **cetakBarisan**, yang menerima dua parameter: **n**, yang merupakan bilangan input, dan **current**, yang digunakan untuk melacak bilangan yang sedang diproses. Fungsi ini pertama-tama memeriksa apakah **current** kurang dari 1; jika ya, fungsi akan berhenti, menandakan bahwa semua bilangan telah dicetak. Jika tidak, fungsi akan mencetak nilai **current**, kemudian memanggil dirinya sendiri dengan **current** dikurangi 1 untuk melanjutkan proses mencetak bilangan yang lebih kecil. Setelah mencapai 1, fungsi akan mencetak kembali bilangan dari 1 hingga N dengan memanggil **current** lagi. Di dalam fungsi **main**, program membaca input bilangan bulat positif N dari pengguna tanpa memberikan pesan apa pun, lalu memanggil fungsi **cetakBarisan** untuk menghasilkan urutan bilangan sesuai dengan yang diinginkan. Setelah semua bilangan dicetak, program menambahkan baris baru untuk memastikan tampilan output yang rapi. Program ini dirancang untuk berjalan tanpa interaksi tambahan, sehingga fokus pada hasil akhir dari urutan bilangan.

Unguided 5

### Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil. Masukan terdiri dari sebuah bilangan bulat positif N. Keluaran terdiri dari barisan bilangan ganjil dari 1 hingga N.

No	Masukan	Keluaran
1	5	1 3 5
2	20	1 3 5 7 9 11 13 15 17 19

### Sourcecode

```
package main

import "fmt"

func cetakGanjil(n int) {
    if n < 1 {
        return
    }
    cetakGanjil(n - 1)
    if n%2 != 0 {
        fmt.Print(n, " ")
    }
}

func main() {
    var n int
    fmt.Scanln(&n)
    cetakGanjil(n)
    fmt.Println()
}
```

### Screenshoot Output

```
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\modull 6\unguided5.go"
20
1 3 5 7 9 11 13 15 17 19
PS D:\semester 3\Praktikum Alpro 2\modull 6> █
```

### Deskripsi Program

Program ini berfungsi untuk menampilkan semua bilangan ganjil dari 1 hingga N menggunakan metode rekursif. Di dalamnya terdapat fungsi bernama cetakGanjil yang menerima satu parameter, yaitu n, yang merupakan bilangan input dari pengguna. Fungsi ini akan berhenti jika n kurang dari 1. Jika n lebih besar atau sama dengan 1, fungsi akan memanggil dirinya sendiri dengan mengurangi n satu per satu hingga mencapai 0. Setelah itu, fungsi memeriksa apakah n adalah bilangan ganjil. Jika ya, maka n akan dicetak. Di bagian utama program, nilai N dibaca dari input pengguna, dan kemudian fungsi cetakGanjil dipanggil untuk mencetak semua bilangan ganjil dari 1 hingga N. Program juga menambahkan baris baru setelah mencetak hasil agar tampilan lebih rapi. Dengan cara ini, program menggunakan rekursi untuk menghasilkan dan menampilkan urutan bilangan ganjil.

#### Unguided 6

##### Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan. Masukan terdiri dari bilangan bulat x dan y. Keluaran terdiri dari hasil x dipangkatkan y. Catatan: diperbolehkan menggunakan asterik "\*", tapi dilarang menggunakan import "math".

No	Masukan	Keluaran
1	2 2	4
2	5 3	125

##### Sourcecode

```
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.Scanln(&x)  
    fmt.Scanln(&y)  
    hasil := pangkat(x, y)  
    fmt.Println(hasil)  
}
```

### Screenshoot Output

```
Masukkan bilangan y: 0 dipangkatkan 0 adalah 1  
PS D:\semester 3\Praktikum Alpro 2\modul1 6> go run "d:\semester 3\Praktikum Alpro 2\modul1 6\unguided6.go"  
5  
3  
125  
PS D:\semester 3\Praktikum Alpro 2\modul1 6> |
```

### Deskripsi Program

Program ini bertujuan untuk menghitung hasil pangkat dari dua bilangan bulat menggunakan pendekatan rekursif. Di awal, program mengimpor paket `fmt` yang diperlukan untuk melakukan input dan output. Fungsi utama yang didefinisikan dalam program adalah `pangkat`, yang menerima dua parameter: `x`, yaitu bilangan yang akan dipangkatkan, dan `y`, yaitu pangkatnya. Di dalam fungsi ini, terdapat logika untuk menangani kasus dasar, di mana jika `y` sama dengan 0, maka hasilnya adalah 1, sesuai dengan aturan bahwa setiap bilangan pangkat 0 adalah 1. Jika `y` lebih besar dari 0, fungsi akan memanggil dirinya sendiri dengan `y` dikurangi 1, sehingga secara bertahap menghitung hasil pangkat dengan mengalikan `x` dengan hasil pangkat dari `x` dan `y-1`.