

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 5  
REKURSIF**



**Disusun Oleh :**

**Aryo Tegar Sukarno / 2311102018  
11 – IF – 6**

**Dosen Pengampu :**

**Abednego Dwi Septiadi S. Kom. M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Rekursi adalah salah satu konsep penting dalam pemrograman yang memungkinkan sebuah fungsi untuk memanggil dirinya sendiri. Ini sering digunakan untuk menyelesaikan masalah yang kompleks dengan cara yang lebih sederhana dan elegan. Berikut adalah penjelasan mendalam mengenai dasar teori rekursif:

### 1. Pengertian Rekursi

- **Definisi:** Rekursi adalah teknik di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan sub-masalah dari masalah yang lebih besar.
- **Perbandingan dengan Iterasi:** Meskipun rekursi mirip dengan loop (iterasi), rekursi lebih cocok untuk masalah yang memiliki struktur berulang atau hierarkis.

### 2. Komponen Rekursi

- **Base Case (Kasus Dasar):**
  - Merupakan kondisi yang menghentikan pemanggilan fungsi rekursif. Tanpa base case, fungsi akan terus memanggil dirinya sendiri, yang dapat menyebabkan stack overflow.
  - Contoh: Dalam menghitung faktorial, base case adalah ketika  $n = 0$ , di mana faktorial dari 0 adalah 1.
- **Recursive Case (Kasus Rekursif):**
  - Bagian di mana fungsi memanggil dirinya sendiri dengan parameter yang dimodifikasi. Ini adalah bagian yang menyelesaikan sub-masalah.
  - Contoh: Dalam faktorial, jika  $n > 0$ , maka faktorial  $n$  adalah  $n * \text{faktorial}(n-1)$ .

## II. GUIDED I

### Soal Studi Case

XXXXXXX

### Sourcecode

```
//2311102018_ Aryo Tegar Sukarno

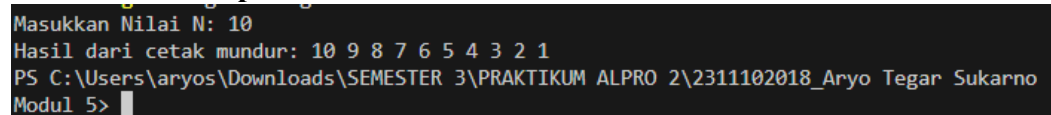
package main

import "fmt"

func cetakmundur(n int ) {
    if n == 1{
        fmt.Println(n)
        return
    }
    fmt.Println(n, " ")
    cetakmundur(n - 1)
}

func main() {
    var n int
    fmt.Print("Masukan Nilai N :")
    fmt.Scanln(&n)
    fmt.Print("Hasil dari cetak mundur :")
    cetakmundur(n)
}
```

### Screenshoot Output



```
Masukkan Nilai N: 10
Hasil dari cetak mundur: 10 9 8 7 6 5 4 3 2 1
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno
Modul 5>
```

### Deskripsi Program

Kode Program ini adalah sederhana dari penerapan rekursi untuk mencetak angka secara mundur. Konsep rekursi sangat berguna untuk menyelesaikan masalah yang bersifat rekursif, seperti traversal pada struktur data pohon atau pencarian solusi pada masalah kombinatorik.

**GUIDED II**  
**Soal Studi Case**  
**XXXXXXX**  
**Sourcecode**

```
//2311102018_Aryo Tegar Sukarno

package main

import "fmt"

func jumlahrekursif(n int) int {

    if n == 1{

        return 1

    }

    return n + jumlahrekursif(n-1)

}

func main() {

    var n int

    fmt.Print("Masukan nilai n untuk penjumlahan 1 hingga
n: ")

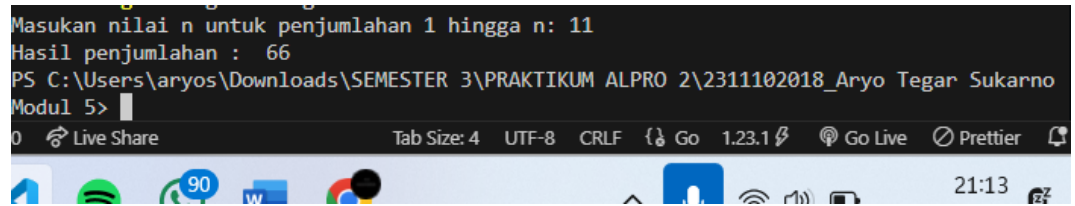
    fmt.Scanln(&n)

    fmt.Println("Hasil penjumlahan : " ,
jumlahrekursif(n))

}
```

**Screenshoot Output**

```
Masukan nilai n untuk penjumlahan 1 hingga n: 11
Hasil penjumlahan : 66
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno
Modul 5> |
```



## Deskripsi Program

Kode ini dirancang untuk menghitung jumlah bilangan bulat dari 1 hingga n menggunakan rekursi. Program sederhana bagaimana rekursi dapat digunakan untuk menyelesaikan masalah penjumlahan secara elegan. Meskipun masalah ini bisa diselesaikan dengan perulangan, rekursi seringkali memberikan solusi yang lebih intuitif dan mudah dipahami untuk masalah-masalah tertentu.

### GUIDED III

#### Soal Studi Case

XXXXXX

#### Sourcecode

```
//2311102018_ Aryo Tegar Sukarno

package main

import "fmt"

func pangkatdua(n int) int {

    if n == 0 {

        return 1

    }

    return 2 * pangkatdua(n-1)

}

func main() {

    var n int

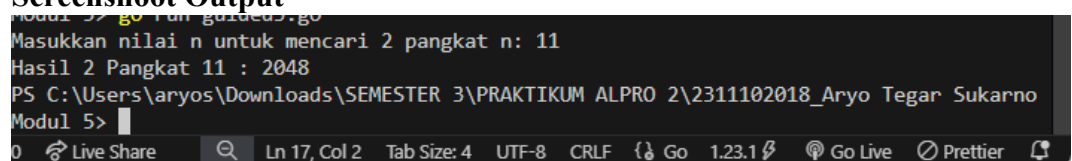
    fmt.Print("Masukkan nilai n untuk mencari 2 pangkat n: ")

    fmt.Scan(&n) // Perbaiki 'fmt.scan' menjadi 'fmt.Scan'

    fmt.Println("Hasil 2 Pangkat", n, ":", pangkatdua(n))

}
```

#### Screenshoot Output



```
Modul 5> go run guided3.go
Masukkan nilai n untuk mencari 2 pangkat n: 11
Hasil 2 Pangkat 11 : 2048
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno
Modul 5>
0 Live Share Ln 17, Col 2 Tab Size: 4 UTF-8 CRLF Go 1.23.1 Go Live Prettier
```

#### Deskripsi Program

Kode ini menggunakan konsep rekursi untuk menyelesaikan masalah perhitungan pangkat secara efisien. Meskipun masalah ini bisa diselesaikan dengan perulangan, rekursi seringkali memberikan solusi yang lebih elegan dan mudah dipahami untuk masalah-masalah tertentu.

## GUIDED IV

### Soal Studi Case

**Xxxxx**

### Sourcecode

```
//2311102018_ Aryo Tegar Sukarno

package main

import "fmt"

func faktorial(n int ) int {

    if n == 0 || n == 1{

        return 1

    }

    return n * faktorial(n-1)

}

func main(){

    var n int

    fmt.Print("Masukan n mencari Faktorial n!: ")

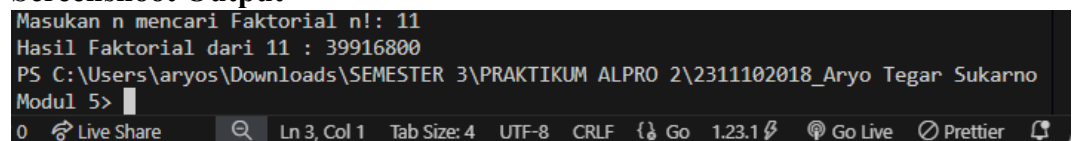
    fmt.Scanln(&n)

    fmt.Println("Hasil Faktorial dari",n,

":",faktorial(n))

}
```

### Screenshoot Output



```
Masukan n mencari Faktorial n!: 11
Hasil Faktorial dari 11 : 39916800
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno
Modul 5> 
```

### Deskripsi Program

Kode ini memberikan contoh sederhana bagaimana rekursi dapat digunakan untuk menghitung faktorial. Konsep rekursi adalah konsep penting dalam ilmu komputer dan sering digunakan dalam berbagai algoritma dan struktur data.

### III. UNGUIDED I

#### 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  $S_2 = S_{n-1} + S_{n-2}$ . Berikut ini adalah contoh nilai deret fibonacci hingga suku ke-10. Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci tersebut.

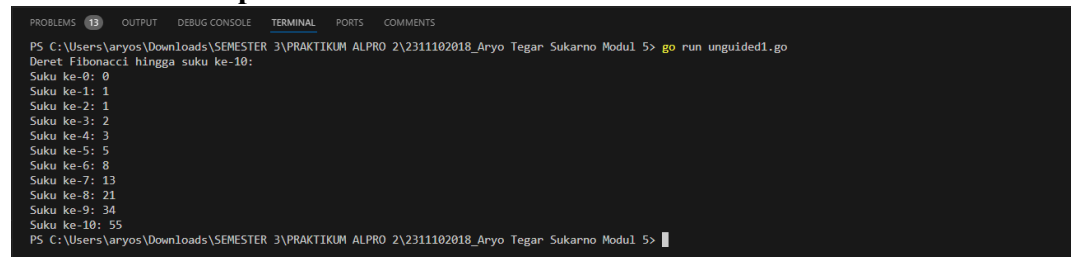
#### Sourcecode

```
//2311102018-Aryo Tegar Sukarno
package main
import (
    "fmt"
)

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

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

#### Screenshoot Output



```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5> go run unguided1.go
Deret Fibonacci hingga suku ke-10:
Suku ke-0: 0
Suku ke-1: 1
Suku ke-2: 1
Suku ke-3: 2
Suku ke-4: 3
Suku ke-5: 5
Suku ke-6: 8
Suku ke-7: 13
Suku ke-8: 21
Suku ke-9: 34
Suku ke-10: 55
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5> |
```



**Deskripsi Program**

Fungsi ini menggunakan rekursi untuk menghitung suku Fibonacci ke-n. Kode ini dirancang untuk menghitung dan mencetak deret Fibonacci hingga suku ke-n yang ditentukan. implementasi rekursif untuk menghitung deret Fibonacci. Konsep rekursi ini sering digunakan dalam pemrograman untuk menyelesaikan masalah yang memiliki struktur berulang.

## UNGUIDED II

## Soal Studi Case

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

## Sourcecode

```
//2311102018 Aryo Tegar Sukarno
package main
import "fmt"

func cetakbintang(n int) {
    if n <= 0 {
        return
    }
    cetakbintang(n - 1)
    for i := 0; i < n; i++ {
        fmt.Print("* ")
    }
    fmt.Println()
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan n: ")
    fmt.Scan(&n)
    fmt.Print("\n")
    cetakbintang(n)
}
```

## Screenshot Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
```

```
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5> go run .  
go: go.mod file not found in current directory or any parent directory; see 'go help modules'  
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5> go run unguided2.go  
Masukkan bilangan n: 11  
*  
**  
***  
****  
*****  
******  
*******  
********  
*********  
**********  
***********  
****  
***  
**  
*
```

```
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5>
```

0 Live Share

## Deskripsi Program

Fungsi cetakbintang( $n - 1$ ) dipanggil terlebih dahulu, sehingga bagian pencetakan bintang terjadi setelah rekursi mencapai base case ( $n \leq 0$ ). Pola ini memastikan baris dengan jumlah bintang yang lebih kecil dicetak lebih dahulu, diikuti oleh baris dengan lebih banyak bintang.

## UNGUIDED III

### Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu  $N$ , atau bilangan yang apa saja yang habis membagi  $N$ .

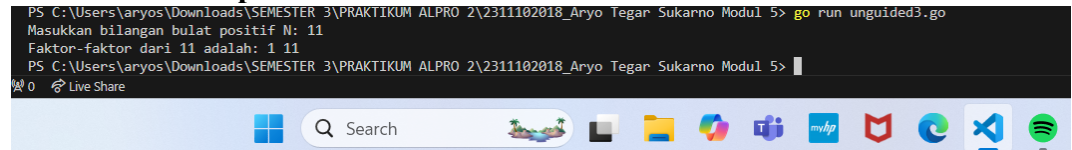
Masukan terdiri dari sebuah bilangan bulat positif  $N$ . Keluaran terdiri dari barisan bilangan yang menjadi faktor dari  $N$  (terurut dari 1 hingga  $N$  ya).

### Sourcecode

```
//Aryo Tegar Sukarno 2311102018
package main
import (
    "fmt"
)
func findFactors(N, divisor int) {
    if divisor > N {
        return
    }
    if N%divisor == 0 {
        fmt.Printf("%d ", divisor)
    }
    findFactors(N, divisor+1)
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)
    fmt.Printf("Faktor-faktor dari %d adalah: ", N)
    findFactors(N, 1)
    fmt.Println()
}
}
```

## Screenshoot Output



## Deskripsi Program

Program ini ditulis dalam bahasa Go (Golang) dan bertujuan untuk menemukan serta mencetak semua faktor dari bilangan bulat positif N yang dimasukkan oleh pengguna. Program menggunakan rekursi untuk mencari faktor-faktor tersebut.

## UNGUIDED IV

### Soal Studi Case

Buatlah program yang menimplementasikan 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.

### Sourcecode

```
package main

import (
    "fmt"
)

func printSeries(N, current int) {
    if current == 1 {
        fmt.Printf("%d ", current)
        return
    }
    fmt.Printf("%d ", current)
    printSeries(N, current-1)
    fmt.Printf("%d ", current)
}

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

```

    fmt.Printf("Barisan bilangan dari %d hingga 1 dan kembali ke
%d:\n", N, N)
    printSeries(N, N)
    fmt.Println()
}

```

## Screenshoot Output

```

Masukkan bilangan bulat positif N: 11
Barisan bilangan dari 11 hingga 1 dan kembali ke 11:
11 10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10 11
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5>

```

## Deskripsi Program

Program ini menggunakan rekursi untuk mencetak urutan bilangan, yang menunjukkan bagaimana pemanggilan fungsi dapat digunakan untuk menyelesaikan masalah dengan cara yang elegan. Namun, variabel **N** dalam fungsi **printSeries** tidak digunakan, sehingga bisa dihapus untuk meningkatkan kejelasan kode.

## UNGUIDED V

### Soal Studi Case

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

### Sourcecode

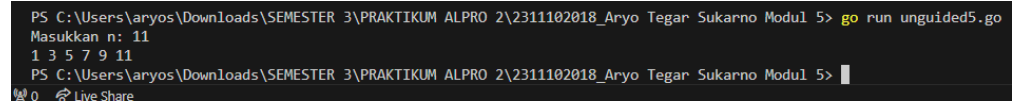
```

//2311102018-Aryo TegarSukarno
package main
import "fmt"
func ganjil(n int) {
    if n < 1 {
        return
    }
    if n%2 != 0 {
        ganjil(n - 2)
        fmt.Print(n, " ")
    } else {
        ganjil(n - 1)
    }
}
func main() {
    var n int

```

```
    fmt.Print("Masukkan n: ")
    fmt.Scanln(&n)
    ganjil(n)
}
```

### Screenshoot Output



```
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5> go run unguided5.go
Masukkan n: 11
1 3 5 7 9 11
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5>
```

### Deskripsi Program

Program ini menggunakan rekursi untuk mencetak bilangan ganjil dari n hingga 1. Pendekatan ini menunjukkan bagaimana pemanggilan fungsi dapat digunakan untuk menyelesaikan masalah secara elegan. Penggunaan kondisi untuk memeriksa apakah angka ganjil atau genap juga merupakan teknik yang baik untuk mengontrol alur logika dalam fungsi. Program ini bekerja dengan baik untuk bilangan bulat positif, tetapi tidak ada penanganan untuk bilangan negatif atau nol, yang dapat menjadi pertimbangan untuk pengembangan lebih lanjut.

## UNGUIDED IV

### Soal Studi Case

Buatlah program yang menimplementasikan 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.

### Sourcecode

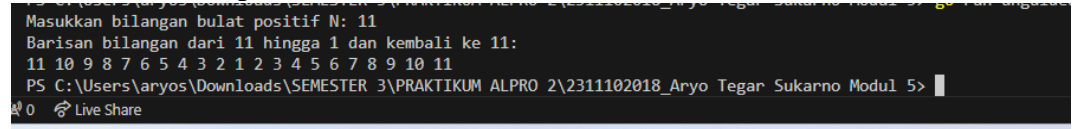
```
//2311102018-Aryo TegarSukarno
package main
import "fmt"
func ganjil(n int) {
    if n < 1 {
        return
    }
    if n%2 != 0 {
        ganjil(n - 2)
        fmt.Print(n, " ")
    }
}
```

```

    } else {
        ganjil(n - 1)
    }
}
func main() {
    var n int
    fmt.Print("Masukkan n: ")
    fmt.Scanln(&n)
    ganjil(n)
}

```

### Screenshoot Output



```

Masukkan bilangan bulat positif N: 11
Barisan bilangan dari 11 hingga 1 dan kembali ke 11:
11 10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10 11
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5>

```

### Deskripsi Program

Fungsi printOddNumbers Fungsi ini menerima dua parameter: N (batas atas) dan current (bilangan saat ini yang sedang diperiksa). Jika current lebih besar dari N, fungsi akan berhenti (base case). Jika current adalah bilangan ganjil (diperiksa dengan  $\text{current} \% 2 \neq 0$ ), maka bilangan tersebut dicetak. Fungsi kemudian memanggil dirinya sendiri dengan  $\text{current}+1$  untuk memeriksa bilangan berikutnya.

## UNGUIDED VI

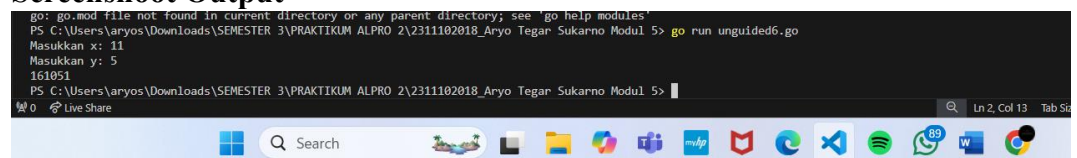
### Soal Studi Case

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

### Sourcecode

```
//231110202018 Aryo Tegar Sukarno
```

### Screenshoot Output



```

go: mod file not found in current directory or any parent directory; see 'go help modules'
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5> go run unguided6.go
Masukkan x: 11
Masukkan y: 5
161051
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Aryo Tegar Sukarno Modul 5>

```

### Deskripsi Program

Program ini menggunakan pendekatan rekursif untuk menghitung pangkat dari suatu bilangan bulat. Pendekatan ini menunjukkan bagaimana fungsi dapat memanggil dirinya sendiri untuk menyelesaikan masalah yang lebih besar dengan membaginya menjadi sub-masalah yang lebih kecil. Program ini berfungsi dengan baik untuk eksponen positif dan nol, tetapi tidak menangani eksponen negatif, yang dapat menjadi area untuk pengembangan lebih lanjut.