

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 2**  
**MODUL V**  
**REKURSIF**



Oleh:

ANISSA FAUZIA ISYANTI

2311102219

S1IF-11-02

**S1 TEKNIK INFORMATIKA**  
**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**  
**2024**

## I. DASAR TEORI

### A. Pengantar Rekursif

Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identik dari masalah utama, di mana sebuah fungsi memanggil dirinya sendiri.

	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	

#### Catatan:

- Teknik rekursif merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (fungsi atau prosedur).
- Untuk menghentikan rekursif digunakan percabangan (if-then).
- **Base-case** adalah kondisi proses rekursif berhenti. **Base-case** merupakan hal terpenting dan pertama yang harus diketahui ketika akan membuat program rekursif. **Mustahil** membuat program rekursif tanpa mengetahui **base-case** terlebih dahulu.
- Recursive-case **adalah kondisi dimana proses pemanggilan dirinya sendiri dilakukan. Kondisi** recursive-case adalah komplemen atau negasi dari **base-case**.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma rekursif.

### B. Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama:

- **Base-case (Basis)**, yaitu bagian untuk menghentikan proses rekursif dan menjadi komponen terpenting di dalam sebuah rekursif.
- **Recursive-case**, yaitu bagian pemanggilan subprogramnya.

## II. GUIDED

### 1. Guided 1

#### Source code

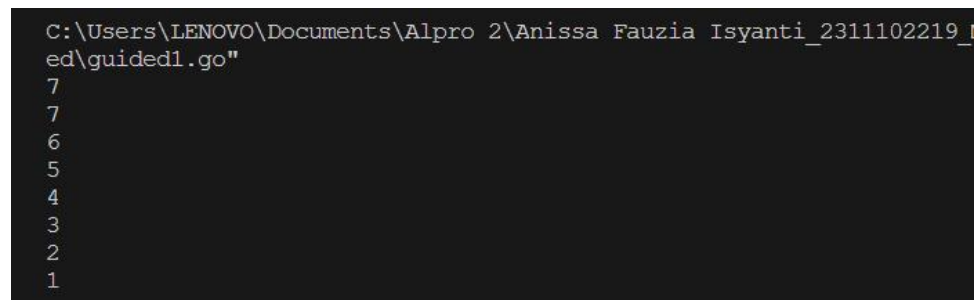
```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n) //membaca input pengguna
    baris(n)     //memanggil fungsi rekursif
    'baris'
}

func baris(bilangan int) {
    if bilangan == 1 { //base case
        fmt.Println(1) //cetak angka 1
    } else {
        fmt.Println(bilangan)
        baris(bilangan - 1)
    }
}
```

#### Screenshot Program



```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219_1\ed\guided1.go"
7
7
6
5
4
3
2
1
```

Program ini mencetak angka dari n hingga 1 secara berurutan. Pertama, program meminta pengguna untuk memasukkan sebuah bilangan bulat (n). Setelah itu, fungsi rekursif bernama **baris** dipanggil dengan nilai n. Fungsi ini memiliki kondisi dasar (**base-case**) yang memeriksa apakah nilai yang diberikan adalah 1. Jika ya, maka program akan mencetak angka 1. Jika tidak, program akan mencetak angka tersebut dan kemudian memanggil dirinya sendiri dengan nilai yang lebih kecil (bilangan - 1) atau decrement, hingga hasil akhirnya akan mencetak bilangan n hingga 1 secara berurutan..

## 2. Guided2

### Source code

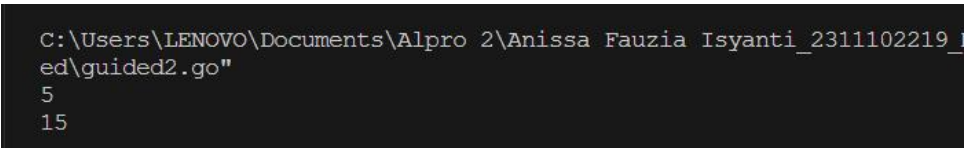
```
package main

import "fmt"

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

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

### Screenshot Program



```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219_1ed\guided2.go"
5
15
```

Program ini menghitung jumlah angka dari 1 hingga n. Pertama, program meminta pengguna untuk memasukkan sebuah bilangan bulat (n). Setelah mendapatkan input tersebut, program akan memanggil fungsi **penjumlahan**, yang menghitung total dengan cara rekursif. Jika nilai n sama dengan 1, fungsi akan mengembalikan 1. Namun, jika n lebih dari 1, fungsi akan menjumlahkan nilai n dengan hasil penjumlahan angka sebelumnya (n-1). Proses ini berlanjut hingga semua angka dari 1 hingga n dijumlahkan. Di Akhir program, akan dicetak hasil penjumlahan tersebut.

### III. UNGUIDED

#### 1. Unguided 1

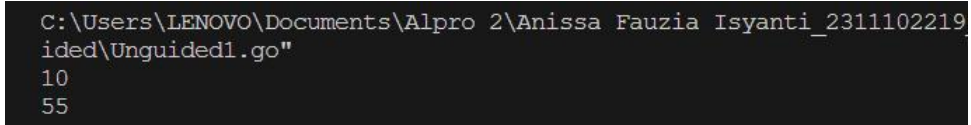
```
package main

import "fmt"

func Fibonacci_219(n int) int {
    if n <= 1 {
        return n
    }
    return Fibonacci_219(n-1) + Fibonacci_219(n-2)
}

func main() {
    var n int
    fmt.Scan(&n)
    fmt.Println(Fibonacci_219(n))
}
```

#### Screenshot Program



```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219\Unguided1.go
10
55
```

Program ini menghitung angka Fibonacci ke-n. Dalam deret Fibonacci, setiap angka adalah jumlah dari dua angka sebelumnya, dimulai dengan 0 dan 1. Program dimulai dengan mendefinisikan fungsi Fibonacci\_219, yang menerima satu parameter input, yaitu n. Jika n kurang dari atau sama dengan 1, fungsi akan mengembalikan nilai n itu sendiri (0 atau 1). Jika tidak, fungsi akan memanggil dirinya sendiri secara rekursif untuk menghitung angka Fibonacci ke-(n-1) dan ke-(n-2), lalu menjumlahkannya.

#### 2. Unguided 2

```
package main

import "fmt"

func cetak_bintang219(n int) {
    if n == 0 {
        return
    } else {
        fmt.Print("* ")
        cetak_bintang219(n - 1)
    }
}
```

```

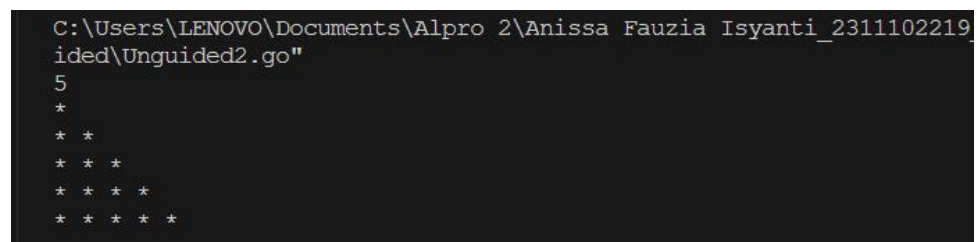
}

func pola(n, i int) {
    if i > n {
        return
    } else {
        cetak_bintang219(i)
        fmt.Println()
        pola(n, i+1)
    }
}

func main() {
    var n int
    fmt.Scan(&n)
    pola(n, 1)
}

```

#### Screenshot Program



The screenshot shows a terminal window with the following output:

```

C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219_
ided\Unguided2.go"
5
*
* *
* * *
* * * *
* * * * *

```

Program ini mencetak pola bintang segitiga. Pengguna diminta untuk memasukkan sebuah angka bulat (n), yang menentukan jumlah baris dalam pola bintang. Fungsi **cetak\_bintang219** bertugas untuk mencetak bintang, di mana jika nilai n adalah 0, fungsi akan berhenti; jika tidak, fungsi akan mencetak bintang dan memanggil dirinya sendiri dengan n yang berkurang satu (decrement), sehingga bintang dicetak dalam jumlah sesuai dengan nilai n. Fungsi **pola** mengatur bagaimana baris pola dicetak. Fungsi ini memeriksa apakah nilai i lebih besar dari n; jika ya, fungsi berhenti; jika tidak, fungsi akan memanggil **cetak\_bintang219(i)** untuk mencetak bintang sebanyak i pada baris tersebut, kemudian melanjutkan dengan memanggil dirinya sendiri dengan nilai i yang ditambah satu. Sehingga program akan mencetak bintang dalam bentuk segitiga, di mana setiap baris memiliki jumlah bintang yang bertambah satu (increment) dari baris sebelumnya.

### 3. Unguided 3

```
package main

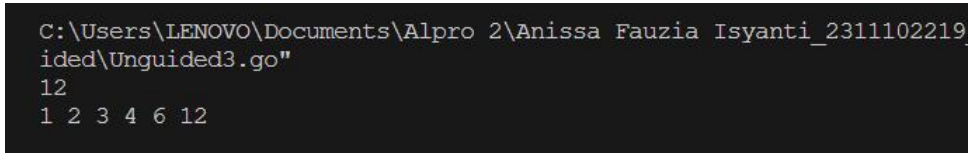
import (
    "fmt"
)

func faktor_219(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    faktor_219(n, i+1)
}

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

    faktor_219(n, 1)
    fmt.Println()
}
```

#### Screenshot Program



```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219\Unguided3.go
12
1 2 3 4 6 12
```

Program ini untuk mencetak semua faktor dari bilangan positif  $n$  yang diinput oleh pengguna. Fungsi `faktor_219` bekerja secara rekursif, di mana fungsi menerima dua parameter:  $n$  sebagai bilangan yang akan dicari faktornya, dan  $i$  sebagai angka pembagi yang dimulai dari 1. Jika  $i$  membagi  $n$  dengan sisa 0 (artinya  $i$  adalah faktor dari  $n$ ), maka  $i$  dicetak. Fungsi terus memanggil dirinya sendiri dengan  $i$  yang bertambah 1 (increment), rekursi akan berhenti ketika  $i$  melebihi nilai  $n$ .

#### 4. Unguided 4

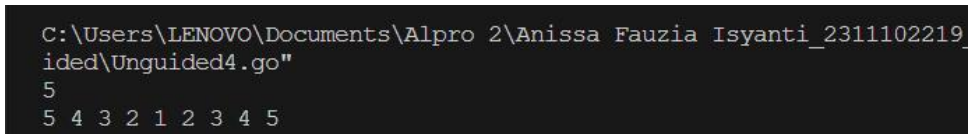
```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)
    barisan_219(n)
    fmt.Print()
}

func barisan_219(n int) {
    if n < 1 {
        return
    }
    fmt.Print(n, " ")
    barisan_219(n - 1)
    if n > 1 {
        fmt.Print(n, " ")
    }
}
```

Screenshot Program



```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219\
ided\Unguided4.go"
5
5 4 3 2 1 2 3 4 5
```

Program ini mencetak barisan bilangan dari n hingga 1, kemudian kembali lagi ke n. Setelah pengguna memasukkan nilai n, fungsi barisan\_219(n) akan dipanggil. Fungsi ini bekerja secara rekursif. Pertama, program mencetak nilai n, kemudian memanggil dirinya sendiri dengan nilai n-1 hingga mencapai kondisi dasar (**base-case**) di mana n kurang dari 1, yang menghentikan rekursi. Setelah mencapai nilai 1, program mencetak kembali nilai n secara bertahap (increment).

#### 5. Unguided 5

```
package main

import "fmt"

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



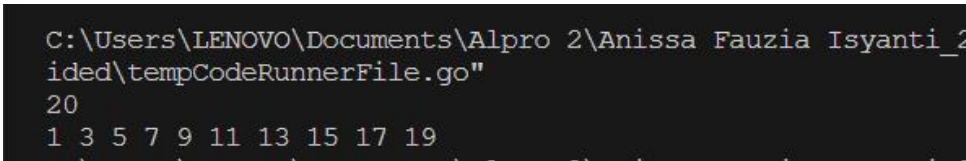
```

        baris_219(n, 1)
    }

    func baris_219(n, bilangan int) {
        if n >= bilangan {
            if bilangan%2 != 0 {
                fmt.Print(bilangan, " ")
            }
            baris_219(n, bilangan+1)
        }
    }
}

```

#### Screenshot Program



```

C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2\
ided\tempCodeRunnerFile.go"
20
1 3 5 7 9 11 13 15 17 19

```

Program ini mencetak bilangan ganjil dari 1 hingga n yang diinput oleh pengguna. Setelah pengguna memasukkan nilai n, fungsi **baris\_219** akan dipanggil dengan parameter awal bilangan = 1. Fungsi baris\_219 menggunakan rekursi untuk memeriksa setiap bilangan dari 1 hingga n. Jika bilangan tersebut ganjil (bilangan mod 2  $\neq$  0), bilangan tersebut dicetak. Proses ini berulang sampai bilangan mencapai n. Program akan berhenti ketika seluruh bilangan ganjil dari 1 hingga n sudah dicetak.

#### 6. Unguided 6

```

package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x, &y)
    fmt.Print(pangkat(x, y))
}

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

```

### Screenshot Program

```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219  
ided\Unguided6.go"  
5 3  
125
```

Program ini untuk menghitung hasil pemangkatan dari dua bilangan bulat, yaitu  $x$  sebagai basis dan  $y$  sebagai eksponen. Program pertama-tama meminta input dua angka dari pengguna. Fungsi pangkat( $x$ ,  $y$ ) bekerja secara rekursif, di mana jika eksponen  $y$  sama dengan 0, fungsi mengembalikan nilai 1 (sebab setiap bilangan dipangkatkan 0 adalah 1). Jika  $y$  lebih besar dari 0, fungsi akan mengalikan nilai  $x$  dengan hasil pemanggilan fungsi pangkat( $x$ ,  $y-1$ ) secara berulang sampai eksponen habis. Dan di akhir program dicetak hasil pemangkatannya.