

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 5**

**REKURSIF**



Disusun Oleh:

NAMA : MARIA DWI A

NIM : 2311102228

KELAS : S1- 1F – 11 - 02

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

# I. DASAR TEORI

## 1. Definisi Procedure

Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identik dari masalah utama. Contoh :

	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	

Apabila diperhatikan subprogram **cetak()** di atas, terlihat pada baris ke-4 terdapat pemanggilan subprogram **cetak()** kembali. Misalnya apabila dieksekusi perintah **cetak()** maka akan menampilkan angka 5 6 7 8 9..dst tanpa henti. Artinya setiap pemanggilan subprogram **cetak()** nilai **x** akan selalu bertambah 1 (*increment by one*) secara **terus menerus tanpa henti**.

```
1 package main
2 import "fmt"
3 func main(){
4     cetak(5)
5 }
6 func cetak(x int){
7     fmt.Println(x)
8     cetak(x+1)
9 }
```

Oleh karena itu, biasanya ditambahkan struktur kontrol percabangan (if then) untuk menghentikan proses rekursif ini. Kondisi ini disebut juga dengan **base-case**, artinya apabila kondisi base-case bernilai true maka proses rekursif akan berhenti. Sebagai contoh misalnya base case adalah ketika **x** bernilai 10 atau **x == 10**, maka tidak perlu dilakukan rekursif.

```

1 procedure cetak(in x:integer)
2 algoritma
3   if x == 10 then
4     output(x)
5   else
6     output(x)
7     cetak(x+1)
8   endif
9 endprocedure

```

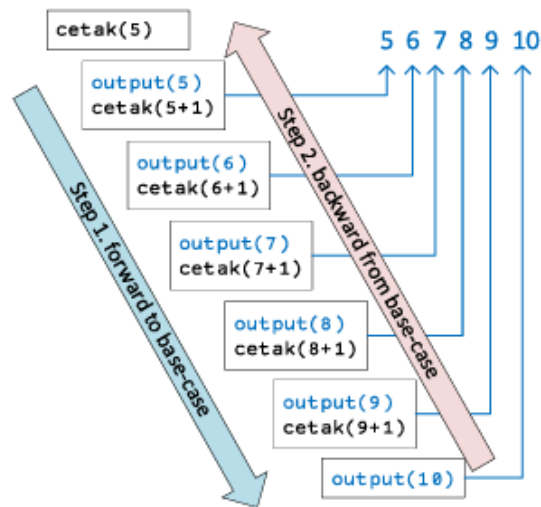
Apabila diperhatikan pada baris ke-3 di Program di atas, kita telah menambahkan **base-case** seperti penjelasan sebelumnya. Selanjutnya pada bagian aksi else di baris ke-6 dan ke-7 dinamkan **recursive-case** atau kasus pemanggilan dirinya sendiri tersebut terjadi. Kondisi dari **recursive-case** ini adalah negasi dari kondisi **base-case** atau ketika nilai  $x \neq 10$ .

```

1 package main
2 import "fmt"
3 func main(){
4   cetak(5)
5 }
6 func cetak(x int){
7   if x == 10 {
8     fmt.Println(x)
9   }else{
10    fmt.Println(x)
11    cetak(x+1)
12  }
13 }

```

Apabila program di atas ini dijalankan maka akan tampil angka 5 6 7 8 9 10. Terlihat bahwa proses rekursif berhasil dihentikan ketika  $x == 10$ .



Gambar 1. Ilustrasi proses forward dan backward pada saat rekursif.

Pada Gambar diatas meperlihatkan saat subprogram dipanggil secara rekursif, maka subprogram akan terus menerus melakukan pemanggilan (**forward**) hingga berhenti pada saat kondisi **base-case** akan terpenuhi atau **true**. Setelah itu terjadi proses **backward** atau kembali ke subprogram yang sebelumnya. Artinya setelah semua instruksi **cetak(10)** selesai dieksekusi, maka program akan kembali ke **cetak(9)** yang memanggil cetak(10) tersebut. begitu seterusnya hingga kembali ke cetak(5).

Perhatikan modifikasi program di atas dengan menukar posisi baris 10 dan 11, mengakibatkan ketika program dijalankan maka akan menampilkan 10 9 8 7 6 5, sesuai dengan proses **backward** pada gambar.

```

1 package main
2 import "fmt"
3 func main(){
4     cetak(5)
5 }
6 func cetak(x int){
7     if x == 10 {
8         fmt.Println(x)
9     }else{
10        cetak(x+1)
11        fmt.Println(x)
12    }
13 }

```

#### Catatan :

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur)
- Untuk mengidentifikasi proses 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 **base-case**.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma iteratif.

## 2. Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama:

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

## II. GUIDED

### 1. Membuat baris bilangan dari n hingga 1

*Jawab :*

Source Code :

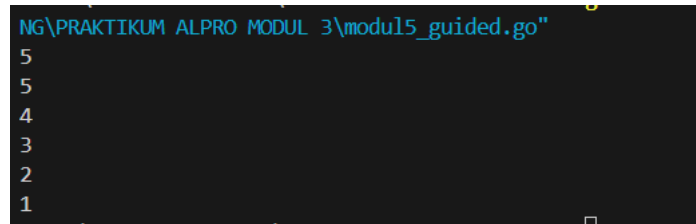
```
package main

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 Program



```
C:\PRAKTIKUM ALPRO MODUL 3\modul5_guided.go
5
5
4
3
2
1
```

## Deskripsi Program

Program diatas merupakan implementasi penggunaan rekursif untuk menampilkan urutan bilangan. Program meminta user untuk menginputkan sebuah bilangan, lalu program akan memanggil fungsi rekursif yang akan menampilkan bilangan ke-n (pada program diatas menginputkan angka 5) hingga bilangan dibawahnya sampai mencapai angka base case yang telah ditentukan yaitu 1.

## 2. Program menghitung penjumlahan 1 hingga n

*Jawab :*

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

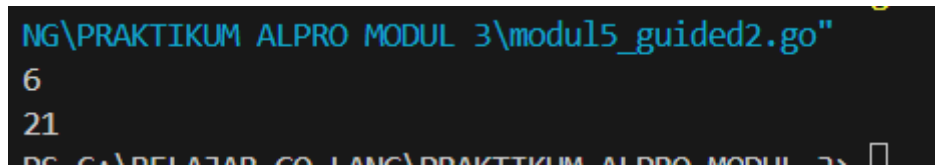
    } else {
```

```
        return n + penjumlahan(n-1)

    }

}
```

### Screenshoot Program



```
NG\PRAKTIKUM ALPRO MODUL 3\modul5_guided2.go"
6
21
PS C:\BELAJAR GO LANG\PRAKTIKUM ALPRO MODUL 3>
```

### Deskripsi Program

Program diatas merupakan implementasi penggunaan rekursif untuk menghitung penjumlahan dari sebuah bilangan. Program akan meminta user untuk menginputkan sebuah angka lalu fungsi rekursif akan dipanggil. Program akan mengeksekusi angka yang diinputkan dengan menjumlahkannya dengan angka dibawahnya sampai pada angka base yang ditentukan yaitu angka 1. Untuk program diatas user menginputkan angka 6, program akan memproses ( $6 + 5 + 4 + 3 + 2 + 1 = 21$ ) melalui fungsi rekursif.



### III. UNGUIDED Soal Modul 5

1. Buatlah program yang mengimplementasikan fungsi rekursif pada deret Fibonacci.

*Jawab :*

Source code

```
package main
import "fmt"

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

func main(){
    var n int
    j:= 0

    fmt.Println("\n*PROGRAM MENGHITUNG DERET
FIBONACCI*")

    fmt.Print("\nBatas deret: ")
    fmt.Scan(&n)
    fmt.Println()

    for i:= 0; i <= n; i++){
        fmt.Print("n", i, " : ")
        fmt.Println(fibonacci(j))
        j++
    }
    fmt.Println()
}
```

## Screenshoot Program

```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO MODUL 3> go run "c:\BELAJAR GO-LANG\PRAK
*PROGRAM MENGHITUNG DERET FIBONACCI*

Batas deret: 10

n0 : 0
n1 : 1
n2 : 1
n3 : 2
n4 : 3
n5 : 5
n6 : 8
n7 : 13
n8 : 21
n9 : 34
n10 : 55
```

## Deskripsi Program

Program tersebut merupakan program implementasi dari fungsi rekursif untuk menghitung suatu deret Fibonacci hingga suku ke-n. Program meminta user untuk menginputkan batas deret yang ingin diketahui deret fibonaccinya. Lalu program akan memanggil fungsi rekursif `func Fibonacci()` untuk menampilkan deret Fibonacci dengan batas yang diinginkan.

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

*Jawab :*

Source code

```
package main

import "fmt"

func cetakBintang(jumlah int){
```

```
        if jumlah <= 0 {
            return
        }
        fmt.Print("*")
        cetakBintang(jumlah - 1)
    }

func baris(n, i int) { // n untuk menentukan jumlah
    baris yang akan dicetak.
        if i > n {
            return
        }
        cetakBintang(i) // akan mencetak bintang sebanyak
i kali
        fmt.Println()
        baris(n, i + 1)
    }

func main(){
    var n int

        fmt.Println("\n*PROGRAM    MENAMPILKAN    POLA
BINTANG*")
        fmt.Print("\nMasukkan jumlah baris(n) : ")
        fmt.Scan(&n)
        fmt.Println("\nPola : \n")

        baris(n, 1)

        fmt.Println()
    }
```

## Screenshoot Program

```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO MODUL 3> go run "c:\BELAJAR GO-LANG\
*PROGRAM MENAMPILKAN POLA BINTANG*

Masukkan jumlah baris(n) : 5

Pola :

*
**
***
****
*****
```

## Deskripsi program

Program diatas merupakan implementasi dari fungsi rekursif yang digunakan untuk menampilkan pola bintang dengan jumlah n yang diinputkan oleh user. Setelah jumlah baris (n) diinputkan maka program akan memanggil fungsi rekursif cetakBintang untuk menampilkan pola bintang dari i (1 sampai ke-n), dan fungsi rekursif baris untuk menampilkan baris yang akan dicetak sesuai dengan (n) kali. Untuk inputan diatas user memasukkan jumlah baris(n) 5 maka program akan menampilkan bintang dari pola ke-1 sampai ke-5.

3. 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 yang menjadi faktor dari N (terurut 1 hingga N)

*Jawab :*

Source code

```
package main

import "fmt"

func cariFaktor(n, i int) {
    if i > n {
        return
    }
    if n % i == 0 {
        fmt.Print(i, " ") // jika n dibagi habis, maka
i adalah faktor dari
    }
    cariFaktor(n, i+1)
    fmt.Println()
}

func main(){
    var n int

    fmt.Println("\n*PROGRAM    MENAMPILKAN    FAKTOR
BILANGAN*")
    fmt.Print("\nMasukkan bilangan : ")
    fmt.Scan(&n)

    fmt.Println("\nFaktor dari", n, " adalah:")
    cariFaktor(n, 1)
    fmt.Println()
}
```

### Screenshoot Program

```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO MODUL 3> go run "c:\BELAJAR GO-L
*PROGRAM MENAMPILKAN FAKTOR BILANGAN*

Masukkan bilangan : 12

Faktor dari 12 adalah:
1 2 3 4 6 12
```

### Deskripsi Program

Program diatas merupakan program implementasi dari fungsi rekursif yang digunakan untuk menampilkan faktor dari sebuah bilangan. Program akan meminta user untuk menginputkan sebuah bilangan lalu fungsi rekursif `cariFaktor` akan dipanggil untuk menentukan faktor dari bilangan yang diinputkan dengan syarat rekursif akan berjalan selama  $i > n$ , jika bilangan yang diinputkan ( $n$ ) habis membagi  $i$  maka ( $i$  dimulai dari 1) tersebut termasuk faktor dari bilangan  $n$ . Untuk inputan program diatas user menginputkan bilangan 12, lalu program akan memanggil fungsi rekursif `cariFaktor` lalu menampilkan faktor dari 12 yaitu 1 2 3 4 6 12.

4. Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu.

*Jawab :*

Source code

```
package main
import "fmt"

func turun(n int) {
    if n < 1 {
        return
```

```

    }

    fmt.Print(n, " ")
    turun(n - 1)
}

func naik(n, i int) {
    if i > n {
        return
    }
    fmt.Print(i, " ")
    naik(n, i + 1)

    fmt.Println()
}

func main(){

    var n int

    fmt.Println("\n*PROGRAM    MENGHITUNG    BARIS
BILANGAN*")

    fmt.Print("\nMasukkan angka : ")
    fmt.Scan(&n)

    fmt.Println()

    fmt.Println("Barisan bilangan : \n")
    turun(n)
    naik(n, 2)

    fmt.Println()

```

```
}
```

### Screenshoot Program

```
*PROGRAM MENGHITUNG BARIS BILANGAN*

Masukkan angka : 9

Barisan bilangan :

9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
```

### Deskripsi Program

Program diatas merupakan implementasi dari fungsi rekursif yang digunakan untuk menghitung/menampilkan deret dari bilangan tertentu. Program meminta user untuk menginputkan sebuah bilangan/angka untuk dihitung deret bilangannya. Terdapat dua fungsi rekursif pada program diatas. Fungsi rekursif pertama adalah func turun() , fungsi ini digunakan untuk menghitung mundur bilangan yang diinputkan. Misal user memasukkan angka 9 maka pada fungsi ini akan mencetak bilangan 9 8 7 6 5 4 3 2 1. Lalu fungsi rekursif yang kedua adalah func naik(), pada fungsi rekursif ini akan melanjutkan mencetak deret bilangan secara naik melanjutkan urutan sebelumnya yaitu 1 2 3 4 5 6 7 8 9.

5. 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.

*Jawab :*

Source code

```
package main

import "fmt"

func ganjil(n, i int){ // i batas maksimum bilangan
yang akan diperiksa dan dicetak
    if n > i { // jika n > i maka rekursif akan
berhenti
        return
    }
    if n % 2 != 0{ // jika hasil bagi n dengan 2
adalah 0, maka bilangan tsb adalah ganjil, lalu
tampilkan pada layar
        fmt.Print(n, " ")
    }
    ganjil(n+1, i) // n + 1, n dimulai dari 0,
    fmt.Println()
}

func main(){
    var i int

    fmt.Println("\n\n*PROGRAM MENCARI BILANGAN
GANJIL*")
    fmt.Print("\nMasukkan jumlah bilangan: ")
    fmt.Scan(&i)

    fmt.Println("\nBilangan yang ganjil : \n")
    ganjil(0, i)

}
```

### Screenshoot Program

```
PS C:\DESKTOP-G0-LAN\PRAKTIKUM ALPRO MODUL 3> go run
NG\PRAKTIKUM ALPRO MODUL 3\modul5_unguided5.go"

*PROGRAM MENCARI BILANGAN GANJIL*

Masukkan jumlah bilangan: 20

Bilangan yang ganjil :

1 3 5 7 9 11 13 15 17 19
```

### Deskripsi Program

Program diatas merupakan program implementasi dari rekursif untuk mencari bilangan yang ganjil dari jumlah bilangan yang diinputkan. Program akan meminta user untuk menginputkan bilangan. Lalu fungsi rekursif `func ganjil()` akan dipanggil yang menerima parameter inputan `n`, dan `i`. `n` dimulai dari 0 dan `i` adalah bilangan yang diinputkan user(pada gambar diatas 20). Lalu fungsi rekursif akan berjalan selama `i < n`, lalu akan memeriksa apakah `i % 2 != 0` (syarat bilangan ganjil, jika dibagi 2 hasilnya tidak sama dengan 0) jika true maka bilangan yang ganjil akan disimpan untuk nanti ditampilkan setelah proses perulangan dengan rekursif selesai. Jika diinputkan bilangan 20 maka didapat bilangan yang ganjil adalah 1 3 5 7 9 11 13 15 17 19.

6. 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`.

*Jawab :*

Source code

```
package main

import "fmt"

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

func main(){

    var a, b int
    fmt.Println("\n*PROGRAM MENGHITUNG PANGKAT*")
    fmt.Print("\nMasukkan basis : ")
    fmt.Scan(&a)
    fmt.Print("Masukkan pangkat : ")
    fmt.Scan(&b)

    fmt.Println("\nHasil pangkat : ", pangkat(a, b))
    fmt.Println()
}
```

### Screenshoot Program

```
NG\PRAKTIKUM ALPRO MODUL 3\modul3_unguided6.go

*PROGRAM MENGHITUNG PANGKAT*

Masukkan basis : 5
Masukkan pangkat : 3

Hasil pangkat : 125
```

### Deskripsi Program

Program diatas merupakan implementasi dari fungsi rekursif untuk mencari hasil pangkat dari dua buah bilangan. Program akan meminta user untuk menginputkan bilangan basis dan bilangan pangkat untuk dihitung pangkatnya. Lalu fungsi `func pangkat()` untuk menampilkan hasil dari perhitungan.