

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V**

**REKURSIF**



Oleh:

Muhammad Rifki Fadhillah

2311102032

IF 11 02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024/2025**

## I. DASAR TEORI

Rekursi adalah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan submasalah dari masalah yang lebih besar. Fungsi rekursif biasanya terdiri dari dua komponen:

1. Basis Kasus (Base Case): Kondisi yang menghentikan pemanggilan rekursif. Ini mencegah fungsi berjalan tanpa henti.
2. Kasus Rekursif (Recursive Case): Bagian di mana fungsi memanggil dirinya sendiri dengan argumen yang dimodifikasi, mendekati basis kasus.

Contoh codingannya:

```
func faktorial(n int) int {  
    if n == 0 { // Basis kasus  
        return 1  
    }  
    return n * faktorial(n-1) // Kasus rekursif  
}
```

## II. GUIDED

### 1. SOURCE CODE

```
package main

import "fmt"

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

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

### OUTPUT

```
golang\Alpro\Modul 5\guided\1\main.go"
5
5
4
3
2
1
```

### DESKRIPSI PROGRAM

Program ini menampilkan urutan bilangan yang diinputkan secara descending. Program ini diawali dengan program meminta inputan berupa bilangan bulat n dari pengguna. Setelah itu, program memanggil prosedur baris dengan parameter bilangan bertipe dasar integer. Di prosedur baris, terdapat percabangan menggunakan kondisi if jika nilai sama dengan satu maka program akan mengeluarkan satu di layar. Jika bilangan lebih dari satu maka program akan mencetak nilai bilangan saat ini dan rekursif dengan parameter bilangan dikurangi satu.

## 2. SOURCE CODE

```
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))
}
```

## OUTPUT

```
golang\Alpro\Modul 5\guided\2\main.go"
5
15
PS D:\Project VS Code\golang\Alpro\Modul
```

## DESKRIPSI PROGRAM

Program ini menampilkan hasil penjumlahan dari n sampai 1. Program ini diawali dengan meminta inputan dari pengguna berupa bilangan bulat n. Program kemudian memanggil fungsi penjumlahan dengan parameter n. Di dalam fungsi penjumlahan, terdapat kondisi dasar (base case) yang memeriksa apakah nilai n sama dengan 1. Jika kondisi ini terpenuhi, fungsi akan mengembalikan nilai 1. Jika n lebih besar dari 1, fungsi akan memanggil dirinya sendiri secara rekursif dengan nilai n dikurangi satu dan menjumlahkannya dengan n saat ini.

### III. UNGUIDED

#### 1. SOURCE CODE

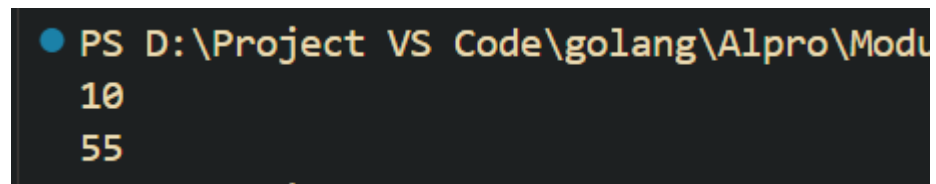
```
package main

import "fmt"

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

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

#### OUTPUT



```
PS D:\Project VS Code\golang\Alpro\Modu
10
55
```

#### DESKRIPSI PROGRAM

Program ini berfungsi untuk menghitung dan menampilkan bilangan Fibonacci ke-n, di mana n merupakan input yang dimasukkan oleh pengguna. Program ini dimulai dengan membaca input pengguna berupa bilangan bulat n. Setelah itu, program memanggil fungsi deretFibonacci dengan parameter n untuk menghitung nilai Fibonacci yang sesuai. Di dalam fungsi deretFibonacci, terdapat percabangan menggunakan kondisi if. Jika nilai n kurang dari atau sama dengan 1, maka program akan mengembalikan nilai n tersebut sebagai hasil (kondisi dasar atau base case). Jika nilai n lebih besar dari 1, program akan rekursif dua kali, yaitu dengan parameter n-1 dan n-2, untuk menghitung dua bilangan Fibonacci sebelumnya, kemudian menjumlahkannya.

Proses rekursif ini terus berlangsung hingga mencapai kondisi dasar, dan nilai akhir bilangan Fibonacci ke-n akan dihitung dan ditampilkan oleh program.

## 2. SOURCE CODE

```
package main

import (
    "fmt"
)

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

func polaBintang(n, i int) {
    if i > n {
        return
    }
    cetakBintang(i)
    fmt.Println()
    polaBintang(n, i+1)
}

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

## OUTPUT

```
● PS D:\Project VS Code\golang\Alpro\
5
*
* *
* * *
* * * *
* * * * *
```

## DESKRIPSI PROGRAM

Program ini berfungsi untuk menampilkan pola bintang bertingkat berdasarkan bilangan  $n$  yang diinputkan oleh pengguna. Pola tersebut menampilkan baris-baris bintang yang jumlahnya bertambah pada setiap baris.

Program dimulai dengan membaca input bilangan bulat  $n$  dari pengguna. Setelah itu, prosedur `polaBintang` dipanggil dengan dua parameter:  $n$  sebagai batas pola dan  $i$  yang dimulai dari 1 untuk menunjukkan baris ke berapa yang sedang dicetak.

Di dalam prosedur `polaBintang`, terdapat percabangan untuk menghentikan rekursi ketika nilai  $i$  lebih besar dari  $n$ . Pada setiap panggilan, prosedur `cetakBintang` dipanggil untuk mencetak  $i$  buah bintang secara horizontal dalam satu baris, di mana nilai  $i$  bertambah di setiap langkah rekursif.

Prosedur `cetakBintang` bekerja dengan cara rekursif juga, mencetak satu bintang per pemanggilan dan mengurangi nilai  $n$  hingga mencapai nol, yang merupakan kondisi dasar untuk menghentikan pencetakan.

Proses ini berulang sampai semua baris bintang tercetak dari baris pertama hingga baris ke- $n$ , dengan setiap baris menampilkan jumlah bintang yang sesuai dengan nomor barisnya.

### 3. SOURCE CODE

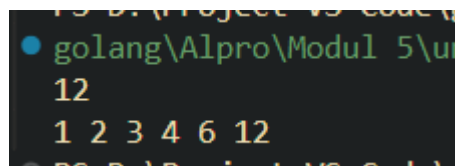
```
package main

import "fmt"

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

func main() {
    var n int
    fmt.Scan(&n)
    faktorBilangan(n,1)
    fmt.Print()
}
```

### OUTPUT



```
golang\Alpro\Modul 5\ur
12
1 2 3 4 6 12
```

### DESKRIPSI PROGRAM

Program ini berfungsi untuk menampilkan faktor-faktor dari bilangan bulat positif yang diinputkan oleh pengguna. Program dimulai dengan membaca input pengguna berupa bilangan bulat n. Setelah itu, program memanggil prosedur faktorBilangan dengan parameter n dan i, di mana i diinisialisasi dengan nilai 1.

Di dalam prosedur faktorBilangan, terdapat kondisi if untuk memeriksa apakah nilai i lebih besar dari n. Jika i lebih besar dari n, prosedur akan berhenti dengan



perintah return, yang merupakan kondisi dasar (base case) dari rekursi.

Jika  $i$  tidak lebih besar dari  $n$ , program akan memeriksa apakah  $n$  habis dibagi  $i$  (yaitu  $n \% i == 0$ ). Jika kondisi ini terpenuhi, program akan mencetak nilai  $i$ , yang merupakan faktor dari  $n$ . Setelah itu, prosedur faktorBilangan akan memanggil dirinya sendiri (rekursif) dengan parameter  $n$  dan  $i$  yang telah ditingkatkan satu (yaitu  $i + 1$ ).

Proses rekursif ini akan terus berlanjut hingga semua faktor dari bilangan  $n$  ditampilkan.

#### 4. SOURCE CODE

```
package main

import "fmt"

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

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

#### OUTPUT

```
● golang\Alpro\Modul 5\unguide
5
5 4 3 2 1 2 3 4 5
```

## DESKRIPSI PROGRAM

Program ini digunakan untuk menampilkan urutan bilangan yang diinputkan secara menurun (descending) dan kemudian kembali lagi ke urutan menaik (ascending), tanpa menduplikasi angka satu. Program dimulai dengan membaca input dari pengguna berupa bilangan bulat  $n$ . Setelah itu, program memanggil fungsi `urutanBilangan` dengan parameter  $n$ .

Di dalam fungsi `urutanBilangan`, terdapat kondisi untuk memeriksa nilai  $n$ . Jika  $n$  kurang dari 1, fungsi akan menghentikan proses rekursif dengan menggunakan pernyataan `return`. Jika  $n$  lebih besar atau sama dengan 1, program akan mencetak nilai  $n$  di layar diikuti dengan spasi. Selanjutnya, fungsi akan memanggil dirinya sendiri (rekursif) dengan parameter  $n$  dikurangi satu.

Setelah proses rekursif selesai, terdapat percabangan yang memeriksa apakah  $n$  lebih besar dari 1. Jika iya, maka program akan mencetak nilai  $n$  kembali di layar. Proses ini memastikan bahwa urutan bilangan yang dicetak adalah menurun dari  $n$  hingga 1, dan kemudian kembali menaik dari 2 hingga  $n$ , tanpa mencetak angka satu dua kali.

## 5. SOURCE CODE

```
package main

import "fmt"

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

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

## OUTPUT

```
PS D:\Project VS Code\golang\Alpro\Modul 5>
golang\Alpro\Modul 5\unguided\5\main.go"
5
1 3 5
```

## DESKRIPSI PROGRAM

Program ini berfungsi untuk menampilkan urutan bilangan ganjil dari 1 hingga bilangan bulat n yang diinputkan oleh pengguna. Program dimulai dengan membaca input pengguna berupa bilangan bulat n. Setelah itu, program memanggil fungsi `urutanBilanganGanjil` dengan parameter n.

Di dalam fungsi `urutanBilanganGanjil`, terdapat percabangan yang menggunakan kondisi if. Jika nilai n kurang dari 1, maka fungsi akan berhenti (return). Jika n

lebih besar atau sama dengan 1, fungsi akan memanggil dirinya sendiri (rekursif) dengan parameter n dikurangi satu.

Setelah memanggil fungsi rekursif, program kemudian memeriksa apakah nilai n adalah bilangan ganjil dengan menggunakan operator modulus. Jika n adalah bilangan ganjil (yaitu  $n \% 2 == 1$ ), maka program akan mencetak nilai n ke layar diikuti dengan spasi.

Proses ini berlanjut hingga nilai n mencapai 1, dan hasil akhirnya adalah urutan bilangan ganjil dari 1 hingga n.

## 6. SOURCE CODE

```
package main

import "fmt"

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

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

## OUTPUT

```
PS D:\Project VS Code\golang\Alpro\Modul 5\unguided\6\main.go> go run main.go
5 3
125
PS D:\Project VS Code\golang\Alpro\Modul 5\unguided\6\main.go>
```

## **DESKRIPSI PROGRAM**

Program ini berfungsi untuk menghitung nilai pangkat dari bilangan  $x$  yang dipangkatkan dengan  $y$ . Program dimulai dengan membaca dua input pengguna berupa bilangan bulat  $x$  dan  $y$ . Setelah itu, program memanggil fungsi `bilanganPangkat` dengan parameter  $x$  dan  $y$ , dan hasilnya dicetak ke layar.

Di dalam fungsi `bilanganPangkat`, terdapat percabangan menggunakan kondisi `if`. Jika nilai  $y$  sama dengan nol, maka fungsi akan mengembalikan nilai 1, karena setiap bilangan yang dipangkatkan dengan nol hasilnya adalah satu (kondisi dasar atau base case). Jika  $y$  lebih besar dari nol, fungsi akan mengembalikan hasil perkalian  $x$  dengan hasil dari pemanggilan rekursif `bilanganPangkat` dengan parameter  $x$  dan  $y$  dikurangi satu.

Proses rekursif ini berlanjut hingga nilai  $y$  mencapai nol, yang merupakan kondisi dasar, setelah itu fungsi akan mengembalikan nilai hasil pangkat yang diinginkan