

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMOGRAMAN II**  
**MODUL V**  
**REKURSIF**



Oleh:

NAMA : DWI HESTI ARIANI

NIM : 2311102094

KELAS : 11- IF -02

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## I. DASAR TEORI

### a. Rekursif

Recursive function adalah sebuah function yang memanggil/mengeksekusi dirinya sendiri. Recursive function bisa dikatakan salah satu yang bisa kita gunakan untuk melakukan perulangan. Ketika menulis kode aplikasi, terkadang ada kasus dimana akan lebih mudah jika dilakukan dengan recursive function. Sebagai contoh perhatikan prosedur cetak berikut ini

	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() diatas , terlihat pada baris ke-4 terdapat pemanggilan subprogram cetak() kembali. Misalnya apabila kita eksekusi perintah cetak(5) 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.

Oleh karena itu biasanya ditambahkan struktur control percabangan (if-then) untuk menghentikan proses rekursif ini, kondisi ini disebut dengan base-case , artinya apabila kondisi base-case bernilai true maka proses rekursif akan berhenti.

### b. Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama yaitu :

- 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

### Guided 1

#### Source Code

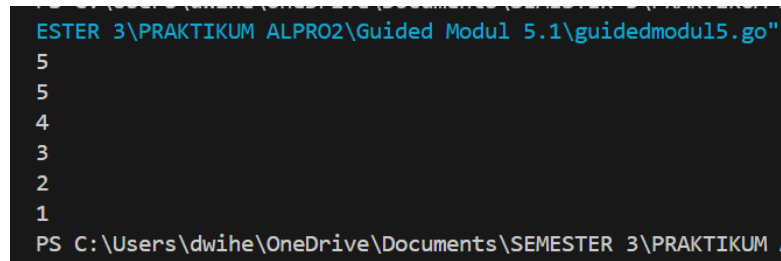
```
package main

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 : jika bilangan sama dengan 1
        fmt.Println(1) // cetak angka 1
    } else { // jika bilangan lebih besar dari 1
        fmt.Println(bilangan) // cetak bilangan saat ini
        baris(bilangan - 1) // panggil fungsi baris dengan
bilangan
    }
}
```

#### Output Program



```
ESTER 3\PRAKTIKUM ALPRO2\Guided Modul 5.1\guidedmodul5.go"
5
4
3
2
1
PS C:\Users\dwihe\OneDrive\Documents\SEMESTER 3\PRAKTIKUM
```

#### Deskripsi Program

Program diatas ialah program sederhana dalam bahasa Go yang berfungsi untuk mencetak angka dari bilangan input pengguna hingga 1 secara menurun menggunakan rekursif. User di minta untuk memasukkan bilangan bulat positif lalu program akan membaca dan menyimpan input dari user , kemudian program akan melakukan proses mencetak angka secara rekursif yang dimana jika bilangan sama dengan 1 , maka fungsi akan langsung mencetak 1 dan proses rekursif berhenti, namun jika bilangan lebih besar dari 1 maka program akan

mencetak bilangan itu sendiri dan mengekursi bilangan lainnya yang bilangan itu dikurangi 1, sehingga nilai bilangan akan semakin kecil.

## Guided 2

### 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)
    }
}
```

### Output Program

```
PS C:\Users\dwihe\OneDrive\Documents\SEMESTER 3\
ESTER 3\PRAKTIKUM ALPRO2\Guided Modul 5.1\Guided
7
28
PS C:\Users\dwihe\OneDrive\Documents\SEMESTER 3\
```

### Deskripsi Program

Program ini merupakan program sederhana Bahasa Golang yang berfungsi untuk melakukan penjumlahan dari bilangan 1 hingga n menggunakan rekursif. User diminta untuk memasukkan sebuah bilangan bulat positif lalu program akan memproses penjumlahan dari input tersebut. Jika  $n = 1$  maka program akan

mencetak 1, namun jika  $n > 1$  maka program akan mengembalikan nilai  $n$  kemudian ditambah dengan penjumlahan  $(n-1)$ .

### III. UNGUIDED

#### Source Code

#### Unguided 1

```
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() {
    for i := 0; i <= 10; i++ {
        fmt.Printf("S%d = %d\n", i, fibonacci(i))
    }
}
```

#### Output Program

```
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5> .\unguided1.go
S0 = 0
S1 = 1
S2 = 1
S3 = 2
S4 = 3
S5 = 5
S6 = 8
S7 = 13
S8 = 21
S9 = 34
S10 = 55
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5>
```

## Deskripsi Program

Program ini merupakan program sederhana Bahasa Golang yang berfungsi untuk menghitung dan mencetak deret Fibonacci hingga  $n$ . Jika  $n = 0$  program akan mengembalikan 0, begitu pun dengan  $n = 1$  program akan mengembalikan nilai 1, jika  $n$  lebih besar 1 program akan menjumlahkan nilai  $n$  dengan Fibonacci ( $n-1$ ) dan Fibonacci ( $n-2$ ).

## Unguided 2

### Source Code

```
Package main

import "fmt"

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

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

## Output Program

```
Masukkan nilai N: 5
*
**
***
****
*****
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5> go r
ESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\unguided Modul 5.2\unguided2.go"
Masukkan nilai N: 1
*
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5> go r
ESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\unguided Modul 5.2\unguided2.go"
Masukkan nilai N: 3
*
**
***
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5> |
```

## Deskripsi Program

Program ini merupakan program sederhana Bahasa Golang yang berfungsi untuk mencetak pola Bintang berdasarkan nilai n yang di inputkan oleh user , jika  $n = 0$  program akan mengembalikan nilai n tanpa melakukan apa-apa ( tidak mencetak Bintang) , namun jika  $n > 0$  maka program akan mencetak pola dengan fungsi n-1 dimana program akan mencetak pola yang paling sedikit terlebih dahulu.

## Unguided 3

### Source Code

```
package main

import (
    "fmt"
)

func printFactors(N, i int) {

    if i > N {
        return
    }

    if N%i == 0 {
        fmt.Println(i)
    }
    printFactors(N, i+1)
}
```

```
func main() {
    var N int

    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)
    fmt.Println("Faktor-faktor dari", N, "adalah:")
    printFactors(N, 1)
}
```

### Output Program

```
Masukkan bilangan bulat positif N: 5
Faktor-faktor dari 5 adalah:
1
5
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5> go run "c:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\unguided Modul 5.2\Unguided Modul 5.3\unguided3.go"
Masukkan bilangan bulat positif N: 12
Faktor-faktor dari 12 adalah:
1
2
3
4
6
12
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5> |
```

### Deskripsi program

Program diatas ialah program sederhana dalam bahasa Go yang berfungsi untuk mencetak factor-faktor dari bilangan bulat positif yang dimasukkan oleh user. Program mempunyai dua parameter yaitu n ( bilangan bulat positif ) dan I (bilangan yang digunakan untuk iterasi ) , jika i lebih besar dari n program tidak mencetak factor . Program akan memeriksa apakah n habis dibagi jika iya i adalah factor dari n dan program akan mencetak hasil i . Kemudian program akan melanjutkan ke bilangan berikutnya dengan i + 1.

### Unguided 4

#### Source Code

```
package main

import "fmt"

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



```

        fmt.Println("Keluaran:", generateSequence(n))
    }

    func generateSequence(n int) string {
        if n == 1 {
            return "1"
        }

        return fmt.Sprintf("%d %s %d", n, generateSequence(n-1), n)
    }

```

### Output Program

```

PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5>
ESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\unguided Modul 5.2\Unguided Modul 5.3\Ung
Masukkan bilangan bulat positif: 5
Keluaran: 5 4 3 2 1 2 3 4 5
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5>
ESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\unguided Modul 5.2\Unguided Modul 5.3\Ung
Masukkan bilangan bulat positif: 9
Keluaran: 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM ALPRO2\unguided modul 5>

```

### Deskripsi Program

Program diatas ialah program sederhana dalam bahasa Go yang berfungsi untuk mengurutkan hasil bilangan positif tertentu. Program akan meminta user untuk memasukkan bilangan bulat positif lalu program akan memproses menggunakan fungsi generateSequence untuk menampilkan hasil urutan bilangan positif .

### Unguided 5

#### Source Code

```

package main

import (
    "fmt"
)

func printOddNumbers(N, i int) {

    if i > N {
        return
    }

```

```

    }

    if i%2 != 0 {
        fmt.Print(i, " ")
    }

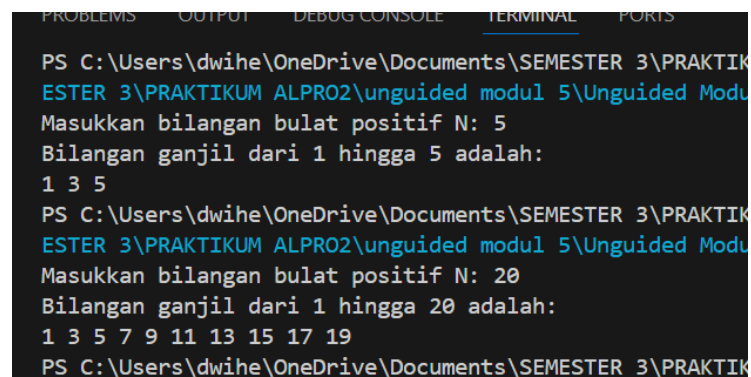
    printOddNumbers(N, i+1)
}

func main() {
    var N int

    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)
    fmt.Println("Bilangan ganjil dari 1 hingga", N, "adalah:")
    printOddNumbers(N, 1)
}

```

### Output Program



```

PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIK
ESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\Unguided Modu
Masukkan bilangan bulat positif N: 5
Bilangan ganjil dari 1 hingga 5 adalah:
1 3 5
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIK
ESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\Unguided Modu
Masukkan bilangan bulat positif N: 20
Bilangan ganjil dari 1 hingga 20 adalah:
1 3 5 7 9 11 13 15 17 19
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIK

```

### Deskripsi Program

Program diatas ialah program sederhana dalam bahasa Go yang berfungsi untuk mencetak semua bilangan ganjil dari hingga n, dimana n adalah nilai yang diinputkan oleh user . Parameter i ( bilangan bulat yang digunakan untuk iterasi). Jika i bilangan ganjil . kondisi  $i \% 2 \neq 0$  maka hasil i akan di cetak , untuk melanjutkan pemeriksaan menggunakan fungsi i+1

## Unguided 6

### Source Code

```
package main

import "fmt"

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

func main() {
    var x, y int
    fmt.Print("Masukkan bilangan bulat x: ")
    fmt.Scanln(&x)
    fmt.Print("Masukkan bilangan bulat y: ")
    fmt.Scanln(&y)

    result := power(x, y)
    fmt.Printf("%d pangkat %d adalah %d\n", x, y, result)
}
```

### Output Program

```
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM
ESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\Unguided Modul 5
Masukkan bilangan bulat x: 2
Masukkan bilangan bulat y: 2
2 pangkat 2 adalah 4
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM
ESTER 3\PRAKTIKUM ALPRO2\unguided modul 5\Unguided Modul 5
Masukkan bilangan bulat x: 5
Masukkan bilangan bulat y: 3
5 pangkat 3 adalah 125
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM
```

### Deskripsi Program

Program diatas ialah program sederhana dalam bahasa Go yang berfungsi untuk menghitung dari bilangan bulat x dipangkatkan y , dimana x dan y di inputkan oleh user . Jika y sama dengan 0, fungsi mengembalikan 1, karena setiap bilangan dipangkatkan 0 hasilnya adalah 1.