

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL VI

Rekursif



Disusun Oleh :

Avrizal Setyo Aji Nugroho

2311102145

IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Rekursif adalah metode penyelesaian masalah di mana suatu fungsi memanggil dirinya sendiri untuk menyelesaikan submasalah yang lebih kecil hingga mencapai kondisi dasar (base case) yang tidak memerlukan pemanggilan lebih lanjut. Pendekatan ini terdiri dari dua komponen utama: base case, yang menentukan kapan rekursi berhenti, dan recursive case, yang mendefinisikan langkah untuk memecahkan masalah dengan memanggil fungsi itu sendiri. Rekursif sering digunakan dalam pemrograman untuk masalah seperti faktorial, deret Fibonacci, traversal struktur data, dan algoritma pembagian seperti merge sort. Meskipun memberikan solusi yang elegan, rekursif memerlukan pengelolaan memori yang hati-hati untuk menghindari stack overflow dan dapat dioptimalkan menggunakan teknik seperti memoization atau tail rekursif.

II. GUIDED

1. Sourcecode

```
package main

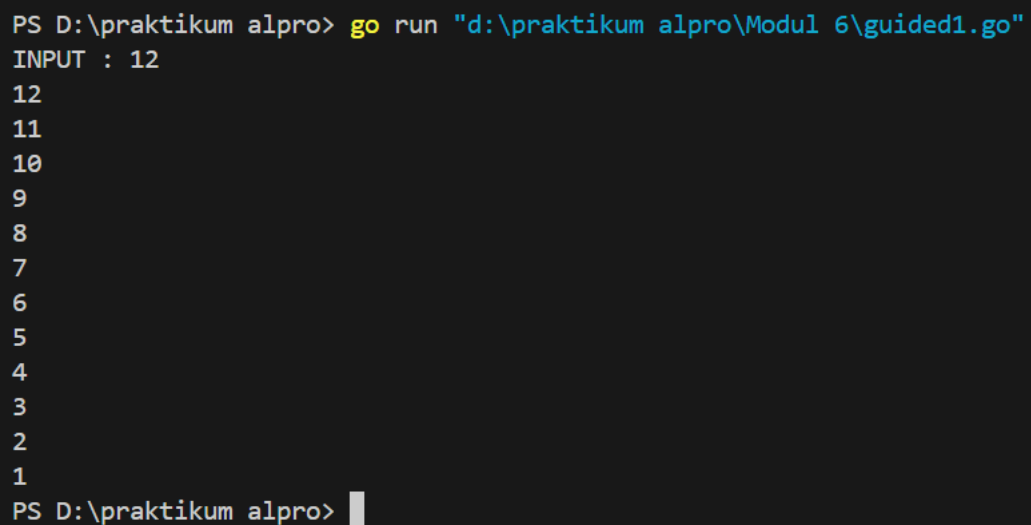
import "fmt"

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

func main() {

    var n int
    fmt.Print("INPUT : ")
    fmt.Scan(&n)
    baris(n)
}
```

Screenshoot Output



```
PS D:\praktikum alpro> go run "d:\praktikum alpro\Modul 6\guided1.go"
INPUT : 12
12
11
10
9
8
7
6
5
4
3
2
1
PS D:\praktikum alpro> 
```

Deskripsi Program

Program di atas adalah implementasi fungsi rekursif dalam bahasa Go yang mencetak bilangan secara menurun mulai dari bilangan yang dimasukkan oleh

pengguna hingga mencapai angka 1. Fungsi utama, `baris`, memeriksa apakah bilangan sama dengan 1 (base case), di mana ia mencetak angka 1 dan menghentikan rekursi. Jika bilangan lebih besar dari 1, fungsi mencetak angka tersebut, lalu memanggil dirinya sendiri dengan bilangan dikurangi 1. Program dimulai dari fungsi `main`, di mana pengguna diminta untuk memasukkan sebuah bilangan, kemudian nilai tersebut diteruskan ke fungsi `baris` untuk mencetak deretan bilangan secara rekursif

2. Sourcecode

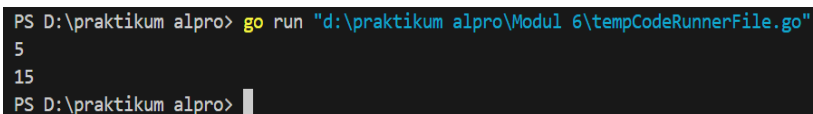
```
package main

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))
}
```

Screenshoot Output



```
PS D:\praktikum alpro> go run "d:\praktikum alpro\Modul 6\tempCodeRunnerFile.go"
5
15
PS D:\praktikum alpro> 
```

Deskripsi Program

Program di atas adalah implementasi fungsi rekursif dalam bahasa Go untuk menghitung jumlah angka dari 1 hingga n. Fungsi `penjumlahan` menerima parameter `n` dan memeriksa apakah `n` sama dengan 1 (base case). Jika iya, fungsi mengembalikan nilai 1, menandakan akhir dari rekursi. Jika tidak, fungsi akan menambahkan nilai `n` dengan hasil rekursif dari `penjumlahan(n-1)`, sehingga menghasilkan penjumlahan angka dari `n` hingga 1. Fungsi `main` meminta pengguna untuk memasukkan nilai `n`, kemudian mencetak hasil dari fungsi `penjumlahan(n)`, yang menghitung jumlah semua angka dari 1 hingga `n` secara rekursif.

3. Sourcecode

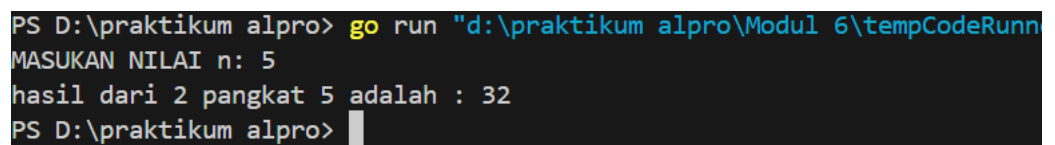
```
package main

import "fmt"

func pangkat(n int) int {
    if n == 0 {
        return 1
    } else {
        return 2 * pangkat(n-1)
    }
}

func main() {
    var n int
    fmt.Print("MASUKAN NILAI n: ")
    fmt.Scan(&n)
    fmt.Println("hasil dari 2 pangkat", n, "adalah :",
    pangkat(n))
}
```

Screenshoot Output



```
PS D:\praktikum alpro> go run "d:\praktikum alpro\Modul 6\tempCodeRunn
MASUKAN NILAI n: 5
hasil dari 2 pangkat 5 adalah : 32
PS D:\praktikum alpro> █
```

Deskripsi Program

Program diatas menghitung nilai 2^n menggunakan fungsi rekursif. Fungsi pangkat memiliki base case $n=0$, yang mengembalikan 1 (karena $2^0=1$). Untuk nilai $n>0$, fungsi memanggil dirinya sendiri dengan $n-1$ dan mengalikan hasilnya dengan 2. Program utama meminta input nilai n dari pengguna, lalu mencetak hasil perhitungan tersebut.

4. Sourcecode

```
package main

import "fmt"

var n int
func faktorial(n int)int{
    if n == 0 || n == 1{
        return 1
    } else {
        return n*faktorial(n-1)
    }
}
func main(){
    fmt.Scan(&n)
    fmt.Println(faktorial(n))
}
```

Screenshoot Output

```
PS D:\praktikum alpro> go run "d:\praktikum alpro\Modul 6\ter
6
720
PS D:\praktikum alpro> |
```

Deskripsi Program

Program berikut menghitung nilai 2^n menggunakan fungsi rekursif.

Fungsi pangkat memiliki base case $n = 0$, yang mengembalikan

1 (karena $2^0 = 1$). Untuk nilai $n > 0$, fungsi memanggil dirinya sendiri

dengan $n-1$ dan mengalikan hasilnya dengan 2. Program utama meminta

input nilai n dari pengguna, lalu mencetak hasil perhitungan tersebut