

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

**MODUL 5
REKURSIF**



Oleh:

ANANDA BASKORO PUTRA

2311102187

IF 11 02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO 2024**

I. DASAR TEORI

Rekursi adalah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan suatu masalah. Dalam banyak kasus, rekursi digunakan untuk memecah masalah yang kompleks menjadi sub-masalah yang lebih kecil dan lebih sederhana. Setiap pemanggilan rekursif akan terus berlanjut hingga mencapai kondisi dasar (base case), yaitu kondisi di mana fungsi berhenti memanggil dirinya sendiri dan mulai mengembalikan hasil. Rekursi sering digunakan untuk menyelesaikan masalah yang memiliki sifat berulang, seperti pencarian dalam struktur data, pemrosesan urutan, atau perhitungan matematika seperti faktorial dan deret Fibonacci. Meskipun rekursi dapat mempermudah implementasi algoritma tertentu, ia juga memiliki potensi untuk menyebabkan konsumsi memori yang tinggi atau terjadinya infinite recursion (rekursi tak berujung) jika base case tidak didefinisikan dengan benar.

Komponen rekursif adalah elemen-elemen utama yang harus ada dalam sebuah fungsi rekursif agar dapat berjalan dengan baik. Terdapat dua komponen penting dalam rekursi, yaitu **base case** dan **recurrence relation**:

Base Case (Kondisi Dasar): Komponen ini adalah kondisi penghentian yang menentukan kapan fungsi rekursif berhenti memanggil dirinya sendiri. Base case sangat penting untuk menghindari rekursi yang berulang tanpa henti (infinite recursion). Ketika base case tercapai, fungsi akan mulai mengembalikan hasilnya dan proses rekursi berhenti. Misalnya, dalam perhitungan faktorial, base case sering kali adalah saat nilai n mencapai 0 atau 1.

Recurrence Relation (Hubungan Rekursif): Bagian ini menggambarkan bagaimana fungsi memanggil dirinya sendiri untuk mengerjakan sub-masalah yang lebih kecil. Setiap pemanggilan rekursif mengurangi ukuran masalah sampai base case tercapai. Hubungan rekursif ini adalah inti dari pemecahan masalah dalam rekursi, di mana solusi untuk masalah yang lebih besar bergantung pada solusi untuk masalah yang lebih kecil.

Dengan adanya kedua komponen ini, sebuah fungsi rekursif dapat berjalan dengan benar, memecah masalah yang kompleks, dan memberikan solusi.

I. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

Guided 1

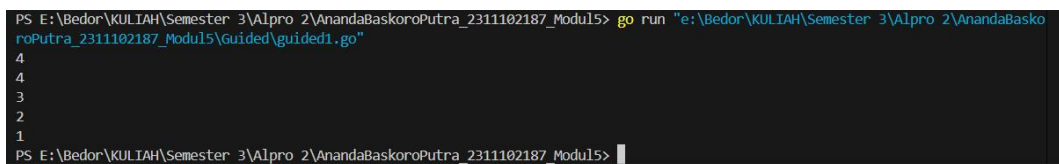
```
package main

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)
    }
}
```

Screenshot Output



```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5\guided\guided1.go"
4
4
3
2
1
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5>
```

Deskripsi Program

Program tersebut berfungsi menerima input berupa bilangan bulat n dari pengguna dan mencetak angka dari n hingga 1 secara berurutan. Fungsi rekursif baris digunakan untuk mencetak angka; jika nilai yang diterima adalah 1, program mencetak 1 dan berhenti, sedangkan jika lebih besar dari 1, program mencetak nilai saat ini dan memanggil dirinya sendiri dengan nilai yang dikurangi 1. Dengan cara ini, program menghasilkan output yang menampilkan semua angka dari n ke 1 di konsol.

Guided 2

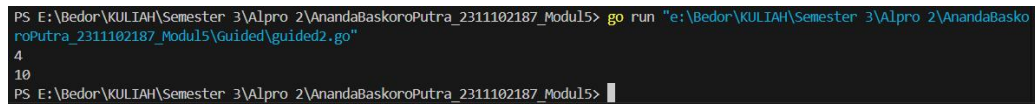
```
package main

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 Output



```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5\Guided\guided2.go"
4
10
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5>
```

Deskripsi Program

Program ini menghitung dan mencetak jumlah dari 1 hingga bilangan bulat n yang diinput oleh pengguna. Fungsi rekursif penjumlahan digunakan untuk menjumlahkan angka; jika nilai n adalah 1, fungsi mengembalikan 1. Jika tidak, fungsi mengembalikan nilai n ditambahkan dengan hasil penjumlahan dari $n-1$, sehingga menghasilkan total jumlah dari 1 hingga n . Output akhir adalah hasil penjumlahan tersebut.

II. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

Unguided 1

```
package main

import "fmt"

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

func main() {
    var n int

    fmt.Print("Masukkan nilai n: ")

    fmt.Scan(&n)

    fmt.Printf("Deret Fibonacci hingga Fibonacci(%d):\n", n)

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

Screenshoot Output

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> go run "e:\Bedor\AnandaBaskoroPutra_2311102187_Modul5\Unguided\Unguided1.go"
Masukkan nilai n: 10
Deret Fibonacci hingga Fibonacci(10):
Fibonacci(0) = 0
Fibonacci(1) = 1
Fibonacci(2) = 1
Fibonacci(3) = 2
Fibonacci(4) = 3
Fibonacci(5) = 5
Fibonacci(6) = 8
Fibonacci(7) = 13
Fibonacci(8) = 21
Fibonacci(9) = 34
Fibonacci(10) = 55
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> |
```

Deskripsi Program

Program ini berfungsi untuk menghitung dan menampilkan deret Fibonacci hingga nilai Fibonacci ke-n, yang diinput oleh pengguna. Fungsi fibonacci menggunakan metode rekursif untuk menghitung nilai Fibonacci dari bilangan yang diberikan. Di dalam fungsi main, program meminta input dari pengguna berupa bilangan bulat n, kemudian mencetak hasil deret Fibonacci dari Fibonacci(0) hingga Fibonacci(n). Nilai-nilai Fibonacci dihitung dan dicetak dalam bentuk "Fibonacci(i) = hasil".

Unguided 2

```
package main

import "fmt"

func cetakBintang(n int) {
    if n == 0 {
        return
    }

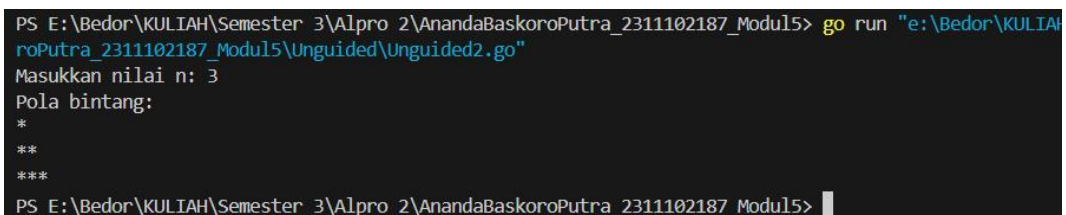
    fmt.Print("*")
}
```

```

        cetakBintang(n - 1)
    }
    func cetakPola(n, current int) {
        if current > n {
            return
        }
        cetakBintang(current)
        fmt.Println()
        cetakPola(n, current+1)
    }
    func main() {
        var n int
        fmt.Print("Masukkan nilai n: ")
        fmt.Scan(&n)
        fmt.Println("Pola bintang:")
        cetakPola(n, 1)
    }
}

```

Screenshoot Output



```

PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> go run "e:\Bedor\KULIAH\roPutra_2311102187_Modul5\Unguided\Unguided2.go"
Masukkan nilai n: 3
Pola bintang:
*
**
***
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5>

```


Deskripsi Program

Program ini bertujuan untuk mencetak pola bintang segitiga menggunakan rekursi. Fungsi cetakBintang bertugas mencetak sejumlah bintang (*) sesuai dengan nilai n yang diberikan, dengan setiap panggilan rekursif mengurangi nilai n hingga mencapai 0. Fungsi cetakPola mencetak setiap baris pola bintang dari 1 bintang hingga n bintang, di mana setiap kali ia memanggil cetakBintang untuk mencetak bintang pada baris tertentu, dan kemudian memanggil dirinya sendiri untuk mencetak baris berikutnya dengan jumlah bintang yang lebih banyak. Di dalam main, program meminta input dari pengguna berupa nilai n, kemudian mencetak pola bintang tersebut dalam bentuk segitiga dengan jumlah baris n.

Unguided 3

```
package main

import "fmt"

func cetakFaktor(n, i int) {
    if i > n {
        return
    }

    if n%i == 0 {
        fmt.Printf("%d ", i)
    }

    cetakFaktor(n, i+1)
}

func main() {
    var n int

    fmt.Print("Masukkan nilai N: ")

    fmt.Scan(&n)

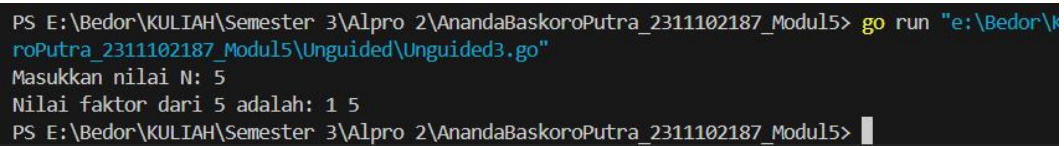
    fmt.Printf("Nilai faktor dari %d adalah: ", n)
```

```
        cetakFaktor(n, 1)

        fmt.Println()

    }
}
```

Screenshoot Output



```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5\Unguided\Unguided3.go"
Masukkan nilai N: 5
Nilai faktor dari 5 adalah: 1 5
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5>
```

Deskripsi Program

Program ini berfungsi untuk mencetak semua faktor dari sebuah bilangan bulat n yang dimasukkan oleh pengguna. Fungsi `cetakFaktor` menggunakan rekursi untuk mencari dan mencetak faktor-faktor dari n . Fungsi ini memulai dengan nilai $i = 1$, kemudian memeriksa apakah n habis dibagi oleh i . Jika ya, maka i dicetak sebagai faktor, lalu fungsi memanggil dirinya sendiri dengan nilai $i+1$ hingga nilai i melebihi n . Di dalam fungsi `main`, program meminta input bilangan n dari pengguna, lalu mencetak semua faktor dari bilangan tersebut dengan memanggil fungsi `cetakFaktor`.

Unguided 4

```
package main

import "fmt"

func cetakUrutan(jumlah int) {
    if jumlah == 1 {
        fmt.Print(jumlah, " ")
        return
    }
}
```

```

        fmt.Print(jumlah, " ")

        cetakUrutan(jumlah - 1)

        fmt.Print(jumlah, " ")
    }

func main() {

    var jumlah int

    fmt.Print("Masukkan jumlah bilangan: ")

    fmt.Scanln(&jumlah)

    fmt.Print("Keluaran : ")

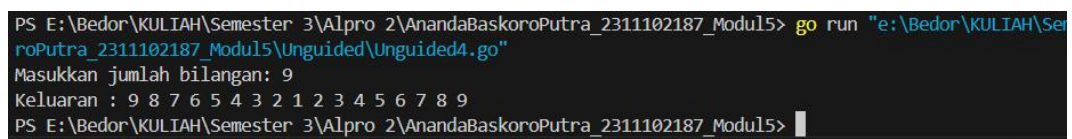
    cetakUrutan(jumlah)

    fmt.Println()

}

```

Screenshot Output



```

PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5\Unguided\Unguided4.go"
Masukkan jumlah bilangan: 9
Keluaran : 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5>

```

Deskripsi Program

Program ini bertujuan untuk mencetak urutan bilangan secara menurun dan kemudian kembali naik. Fungsi cetakUrutan menggunakan rekursi untuk mencetak bilangan dari jumlah ke 1, kemudian mencetak kembali bilangan tersebut saat fungsi keluar dari rekursi.

Unguided 5

```
package main

import "fmt"

func cetakBilanganGanjil(n int) {
    if n < 1 {
        return
    }
    cetakBilanganGanjil(n - 2)
    if n%2 != 0 {
        fmt.Print(n, " ")
    }
}

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

    fmt.Print("Keluaran : ")
    cetakBilanganGanjil(N)
    fmt.Println()
}
```

Screenshoot Output

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5\Unguided\Unguided5.go"
Masukkan nilai N: 5
Keluaran : 1 3 5
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> |
```

Deskripsi Program

Program ini berfungsi untuk mencetak bilangan-bilangan ganjil secara menaik dari 1 hingga nilai N yang diinput oleh pengguna. Fungsi cetakBilanganGanjil menggunakan rekursi untuk mencapai bilangan ganjil terkecil terlebih dahulu (base case), kemudian mencetak bilangan ganjil secara bertahap ketika rekursi "kembali" (naik).

Unguided 6

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung pangkat
func hitungPangkat(x int, y int) int {
    if y < 0 {
        return 0 // Mengembalikan 0 untuk pangkat negatif, karena hasil pangkat bilangan bulat tidak terdefinisi
    }

    hasil := 1

    for i := 0; i < y; i++ {
```

```
        hasil *= x // Hitung pangkat dengan iterasi

    }

    return hasil
}

func main() {

    var bilangan int

    var pangkat int


    // Input dari pengguna

    fmt.Print("Masukkan bilangan: ")

    fmt.Scanln(&bilangan)

    fmt.Print("Masukkan pangkat: ")

    fmt.Scanln(&pangkat)


    // Menghitung hasil pangkat

    hasil := hitungPangkat(bilangan, pangkat)


    // Menampilkan hasil

    fmt.Printf("Hasil %d pangkat %d adalah %d\n", bilangan, pangkat, hasil)
}
```

Screenshoot Output

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> go run "e:\Bedor\roPutra_2311102187_Modul5\Unguided\Unguided6.go"
Masukkan bilangan: 2
Masukkan pangkat: 2
Hasil 2 pangkat 2 adalah 4
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\AnandaBaskoroPutra_2311102187_Modul5> █
```

Deskripsi Program

Program ini berfungsi untuk menghitung pangkat dari sebuah bilangan bulat x yang dipangkatkan ke y menggunakan iterasi. Fungsi `hitungPangkat` menerima dua parameter, yaitu bilangan x dan pangkat y , dan mengembalikan hasil dari operasi tersebut. Jika nilai y negatif, fungsi akan mengembalikan 0 karena program tidak menangani pangkat negatif untuk bilangan bulat. Jika y positif, hasil dihitung menggunakan loop `for`, yang mengalikan nilai x sebanyak y kali.