

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSIF**



Disusun Oleh :

FATTAH RIZQY ADHIPRATAMA / 2311102019

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Rekursi adalah teknik dalam pemrograman di mana suatu fungsi memanggil dirinya sendiri untuk menyelesaikan suatu masalah yang lebih kompleks dengan cara memecahnya menjadi sub-masalah yang lebih kecil. Dalam konteks algoritma pemrograman, rekursi digunakan untuk menyelesaikan masalah yang memiliki struktur yang berulang, atau yang dapat dipecah menjadi masalah yang serupa namun lebih sederhana.

Fungsi rekursif akan terus memanggil dirinya sendiri hingga mencapai kondisi tertentu yang disebut sebagai base case (kasus dasar), di mana pemanggilan rekursif berhenti. Base case adalah kondisi di mana masalah sudah cukup sederhana sehingga dapat diselesaikan langsung tanpa perlu melakukan pemanggilan rekursif lebih lanjut.

Fungsi rekursif biasanya terdiri dari dua bagian:

1. Base Case (Kasus Dasar): Bagian ini mengakhiri proses rekursif dengan kondisi tertentu. Jika base case tidak ada atau tidak dapat tercapai, fungsi rekursif akan terus berlanjut dan mengakibatkan stack overflow atau memori penuh.
2. Recurrence Relation (Hubungan Rekursif): Bagian ini adalah logika atau aturan yang mengatur pemanggilan fungsi terhadap dirinya sendiri. Biasanya, setiap kali fungsi dipanggil, masalah yang dikerjakan akan diperkecil atau dipecah.

II. GUIDED

1.

Sourcecode

```
package main

import "fmt"

// Fungsi untuk mencetak bilangan dari n hingga 1
func cetakMundur(n int) {
    if n == 1 {
        fmt.Println(n)
        return
    }
    fmt.Print(n, " ")
    cetakMundur(n - 1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk cetak bilangan dari n
hingga 1: ")
    fmt.Scanln(&n)
    fmt.Print("Hasil cetak mundur: ")
    cetakMundur(n)
}
```

Screenshoot Output

```
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\guided1.go"
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 6
Hasil cetak mundur: 6 5 4 3 2 1
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> █
```

Deskripsi Program

Program ini menggunakan Bahasa pemrograman Go. Program ini dirancang untuk mencetak serangkaian bilangan secara mundur, mulai dari nilai n yang diinputkan pengguna hingga angka 1. Program ini menggunakan konsep rekursif untuk mencapai tujuan tersebut.

2.

Source Code

```
package main

import "fmt"

// Fungsi untuk menghitung penjumlahan 1 hingga n
func jumlahRekursi(n int) int {
    if n == 1 {
        return 1
    }
    return n + jumlahRekursi(n-1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk penjumlahan 1 hingga n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil penjumlahan:", jumlahRekursi(n))
}
```

Screenshoot Output

```
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\guided2.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 7
Hasil penjumlahan: 28
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> █
```

Deskripsi Program

Program ini merupakan program sederhana dalam bahasa Go. Program ini menghitung penjumlahan berurutan dari 1 hingga n dengan cara menambah n dan hasil penjumlahan dari semua bilangan sebelumnya (melalui pemanggilan rekursif).

3.

Source Code

```
package main

import "fmt"

// Fungsi untuk mencari 2 pangkal n
```

```

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

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk mencari 2 pangkat n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil 2 pangkat", n, ":", pangkatDua(n))
}

```

Screenshoot Output

```

PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\guided3.go"
Masukkan nilai n untuk mencari 2 pangkat n: 8
Hasil 2 pangkat 8 : 256
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5>

```

Deskripsi Program

Program ini merupakan program sederhana dalam bahasa Go. Program ini menggunakan rekursif untuk menghitung nilai dua pangkat n , di mana n adalah bilangan bulat yang dimasukkan oleh pengguna. Program ini terdiri dari dua bagian utama, yaitu fungsi pangkatDua dan fungsi main.

4.

Source Code

```

package main

import "fmt"

// Fungsi untuk menghitung faktorial n!
func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * faktorial(n-1)
}

func main() {
    var n int

```

```
    fmt.Print("Masukkan nilai n untuk mencari faktorial n!:")
    ")
    fmt.Scanln(&n)
    fmt.Println("Hasil faktorial dari", n, ":", faktorial(n))
}
```

Screenshoot Output

```
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\guided4.go"
Masukkan nilai n untuk mencari faktorial n!: 5
Hasil faktorial dari 5 : 120
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> |
```

Deskripsi Program

Program ini merupakan program sederhana dalam bahasa Go. Program ini menggunakan rekursif untuk menghitung faktorial dari suatu bilangan n . Faktorial dari n , ditulis sebagai $n!$, adalah hasil perkalian berurutan dari semua bilangan bulat positif dari 1 hingga n .

III. UNGUIDED

Soal Studi Case 1

Deret fibonacci adalah sebuah deret dengan nilai suku ke-0 dan ke-1 adalah 0 dan 1, dan nilai suku ke-n selanjutnya adalah hasil penjumlahan dua suku sebelumnya. Secara umum dapat diformulasikan $S_n = S_{n-1} + S_{n-2}$. Berikut ini adalah contoh nilai deret fibonacci hingga suku ke-10. Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci tersebut.

n	0	1	2	3	4	5	6	7	8	9	10
S_n	0	1	1	2	3	5	8	13	21	34	55

Source Code

```
package main

import "fmt"

// Fungsi rekursif untuk menghitung deret Fibonacci
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 untuk mencari suku ke-n\n")
    deret Fibonacci: "
    fmt.Scanln(&n)

    result := fibonacci(n)
    fmt.Println("Suku ke-", n, "deret Fibonacci\n")
    adalah:", result)
}
```

Screenshoot Output

```
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 0
Suku ke- 0 deret Fibonacci adalah: 0
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 1
Suku ke- 1 deret Fibonacci adalah: 1
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 2
Suku ke- 2 deret Fibonacci adalah: 1
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 3
Suku ke- 3 deret Fibonacci adalah: 2
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 4
Suku ke- 4 deret Fibonacci adalah: 3
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 5
Suku ke- 5 deret Fibonacci adalah: 5
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
m Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 8
Suku ke- 8 deret Fibonacci adalah: 21
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 9
Suku ke- 9 deret Fibonacci adalah: 34
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided1.go"
Masukkan nilai n untuk mencari suku ke-n deret Fibonacci: 10
Suku ke- 10 deret Fibonacci adalah: 55
```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini menggunakan rekursif untuk menghitung suku ke- n dalam deret Fibonacci. Deret Fibonacci adalah deret bilangan di mana setiap bilangan adalah hasil penjumlahan dari dua bilangan sebelumnya, dengan dua bilangan pertama dalam deret tersebut adalah 0 dan 1.

Soal Studi Case 2

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

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	* ** *** **** *****
2	1	*
3	3	* ** ***

Source Code

```
package main
package main

import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan jumlah baris bintang: ")
    fmt.Scanln(&n)
    printStars(n)
}
```

Screenshoot Output

```
Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided2.go"
Masukkan jumlah baris bintang: 6
*
**
***
****
*****
*****
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> |
```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini menggunakan rekursif untuk mencetak pola bintang (*) dalam bentuk segitiga terbalik yang semakin kecil ke atas. Setiap baris dalam pola ini berisi jumlah bintang yang berkurang dari bawah ke atas.

Soal Studi Case 3

Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu N atau bilangan yang apa saja habis membagi N. Masukan terdiri dari sebuah bilangan bulat positif N Keluaran terdiri dari barisan bilangan yang menjadi faktor dari N (terurut dari 1 hingga N ya)

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	1 5
2	12	1 2 3 4 6 12

Source Code

```
package main

import "fmt"

func findFactors(num, divisor int) {
    if divisor > num {
        return
    }
    if num%divisor == 0 {
        fmt.Print(divisor, " ")
    }
    findFactors(num, divisor+1)
}

func main() {
    var number int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&number)
    fmt.Println("Faktor dari", number, "adalah:")
    findFactors(number, 1)
}
```

Screenshoot Program

```
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided3.go"
Masukkan bilangan: 8
Faktor dari 8 adalah:
1 2 4 8
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> |
```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini menggunakan rekursif untuk menemukan dan mencetak semua faktor dari suatu bilangan bulat yang diberikan oleh pengguna. Faktor dari suatu bilangan adalah angka-angka yang dapat membagi bilangan tersebut secara tepat tanpa sisa. Program ini mencari faktor-faktor dari bilangan dengan cara memeriksa pembagi satu per satu, mulai dari angka 1 hingga bilangan itu sendiri.

Soal Studi Case 4

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu. Masukan terdiri dari sebuah bilangan bulat positif N. Keluaran terdiri dari barisan dari N hingga 1 dan kembali ke N.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	5 4 3 2 1 2 3 4 5
2	9	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9

Source Code

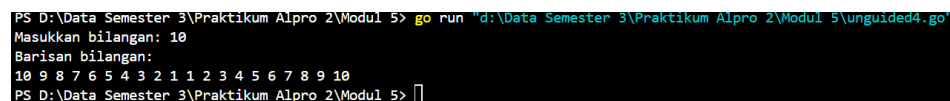
```
package main

import "fmt"

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

func main() {
    var num int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&num)
    fmt.Println("Barisan bilangan:")
    printSequence(num)
}
```

Screenshoot Program



Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini menggunakan rekursif untuk mencetak sebuah barisan angka dengan pola tertentu. Pola yang dihasilkan akan menampilkan angka secara menurun dari n hingga 1, dan kemudian kembali meningkat dari 1 hingga n. Dengan

kata lain, angka akan ditampilkan dalam urutan menurun, kemudian setelah mencapai angka 1, urutan menaik hingga kembali ke n.

Soal Studi Case 5

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil. Masukan terdiri dari sebuah bilangan bulat positif. Keluaran terdiri dari barisan bilangan ganjil dari 1 hingga N

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	1 3 5
2	20	1 3 5 7 9 11 13 15 17 19

Source Code

```
package main

import "fmt"

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

func main() {
    var num int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&num)
    fmt.Println("Barisan bilangan ganjil:")
    printOddNumbers(num)
}
```

Screenshoot Program

```
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided5.go"
Masukkan bilangan: 9
Barisan bilangan ganjil:
1 3 5 7 9
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> |
```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini menggunakan rekursif untuk mencetak barisan bilangan ganjil dari angka 1 hingga nilai n yang diberikan oleh pengguna. Bilangan ganjil adalah bilangan bulat yang tidak habis dibagi oleh 2, seperti 1, 3, 5, dan seterusnya.

Program ini akan mencetak bilangan ganjil dalam urutan menaik menggunakan pendekatan rekursif.

Soal Studi Case 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. Catatan: diperbolehkan menggunakan asterik "*", tapi dilarang menggunakan import "math".

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	2 2	4
2	5 3	125

Source Code

```
package main

import "fmt"

func power(x, y int) int {
    if y == 0 {
        return 1
    } else if y == 1 {
        return x
    } else if y%2 == 0 {
        // Jika pangkat genap, pangkatkan setengah dari
        pangkat lalu kuadratkan
        temp := power(x, y/2)
        return temp * temp
    } else {
        // Jika pangkat ganjil, kalikan x dengan hasil
        pangkat (y-1)
        return x * power(x, y-1)
    }
}

func main() {
    var base, exponent int
    fmt.Print("Masukkan bilangan dasar: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exponent)
```

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

Screenshoot Program

```
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 5\unguided6.go"
Masukkan bilangan dasar: 5
Masukkan pangkat: 4
5 pangkat 4 adalah 625
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 5> █
```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini menggunakan rekursif untuk menghitung nilai pangkat dari sebuah bilangan x yang dipangkatkan dengan y . Program ini memanfaatkan teknik rekursi dengan optimasi berdasarkan paritas (genap/ganjil) dari pangkat untuk mempercepat perhitungan. Metode ini dikenal sebagai exponentiation by squaring atau pemangkatan dengan kuadrat, yang memungkinkan komputasi pangkat menjadi lebih efisien daripada perhitungan berulang biasa.