

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL V

REKURSIF



Disusun Oleh :

Tegar Aji Pangestu/ 2311102021

IF-11-6

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Algoritma rekursif adalah metode pemecahan masalah yang dilakukan dengan cara memanggil dirinya sendiri berulang kali hingga masalahnya terpecahkan. Algoritma ini memecah masalah menjadi submasalah yang lebih kecil dan lebih kecil lagi, hingga mencapai masalah yang cukup kecil untuk dipecahkan dengan mudah.

Algoritma rekursif dapat digunakan untuk memecahkan masalah yang dapat dipecahkan dengan menerapkan solusi pada versi yang lebih kecil dari masalah yang sama.

Contoh klasik dari fungsi rekursif adalah perhitungan faktorial dari suatu bilangan. Faktorial dari n (ditulis sebagai $n!$) adalah produk dari semua bilangan bulat positif dari 1 hingga n .

Rekursi mungkin agak sulit dipahami. Cara terbaik untuk mengetahui cara kerjanya adalah dengan bereksperimen dengannya.

3. GUIDED

Soal Studi Case

Sourcecode

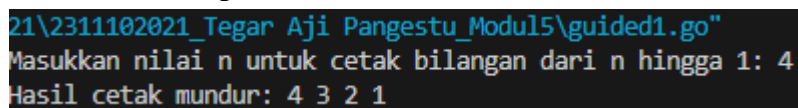
```
package main

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)
}
```

Screenshot Output



```
21\2311102021_Tegar Aji Pangestu_Modul5\guided1.go"
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 4
Hasil cetak mundur: 4 3 2 1
```

Deskripsi Program

Program ini dirancang untuk mencetak serangkaian bilangan secara mundur dari suatu nilai awal n hingga 1. Program ini memanfaatkan konsep rekursi untuk mencapai tujuan tersebut.

4I . GUIDED

Soal Studi Case

Sourcecode

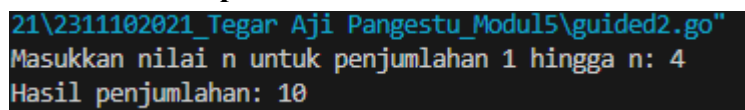
```
package main

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))
}
```

Screenshot Output



```
21\2311102021_Tegar Aji Pangestu_Modul5\guided2.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 4
Hasil penjumlahan: 10
```

Deskripsi Program

Program Go ini dirancang untuk menghitung jumlah bilangan bulat dari 1 hingga suatu nilai n yang diberikan oleh pengguna. Program ini menggunakan konsep rekursi untuk mencapai tujuan tersebut

III . GUIDED

Soal Studi Case

Sourcecode

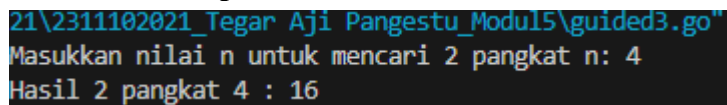
```
package main

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))
}
```

Screenshot Output



```
21\2311102021_Tegar Aji Pangestu_Modul5\guided3.go"
Masukkan nilai n untuk mencari 2 pangkat n: 4
Hasil 2 pangkat 4 : 16
```

Deskripsi Program

Program Go ini dirancang untuk menghitung nilai 2 pangkat n, di mana n adalah bilangan bulat non-negatif yang diberikan oleh pengguna. Program ini menggunakan konsep rekursi untuk mencapai tujuan tersebut.

IV . GUIDED

Soal Studi Case

Sourcecode

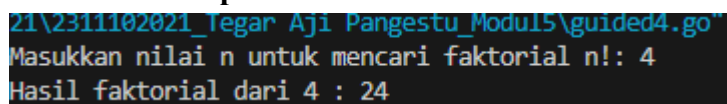
```
package main

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))
}
```

Screenshot Output



```
21\2311102021_Tegar Aji Pangestu_Modul5\guided4.go"
Masukkan nilai n untuk mencari faktorial n!: 4
Hasil faktorial dari 4 : 24
```

Deskripsi Program

Program Go ini dirancang untuk menghitung faktorial dari suatu bilangan bulat non-negatif n yang diberikan oleh pengguna. Faktorial dari suatu bilangan n (ditulis $n!$) adalah hasil perkalian semua bilangan bulat positif dari 1 sampai n . Program ini menggunakan konsep rekursi untuk mencapai tujuan tersebut

V. UNGUIDED

Soal Studi Case

Sourcecode

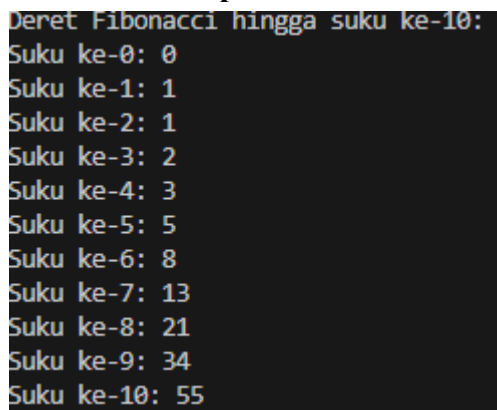
```
package main

import (
    "fmt"
)

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

func main() {
    n := 10
    fmt.Printf("Deret Fibonacci hingga suku ke-%d:\n", n)
    for i := 0; i <= n; i++ {
        fmt.Printf("Suku ke-%d: %d\n", i, fibonacci(i))
    }
}
```

Screenshot Output



```
Deret Fibonacci hingga suku ke-10:
Suku ke-0: 0
Suku ke-1: 1
Suku ke-2: 1
Suku ke-3: 2
Suku ke-4: 3
Suku ke-5: 5
Suku ke-6: 8
Suku ke-7: 13
Suku ke-8: 21
Suku ke-9: 34
Suku ke-10: 55
```

Deskripsi Program

Program Go ini dirancang untuk menghasilkan deret Fibonacci hingga suku ke-n, di mana n adalah bilangan bulat non-negatif yang ditentukan dalam kode. Deret Fibonacci adalah urutan bilangan di mana setiap bilangan adalah jumlah dari dua bilangan sebelumnya. Program ini menggunakan rekursi untuk menghitung suku-suku deret Fibonacci.

VI. UNGUIDED

Soal Studi Case

Sourcecode

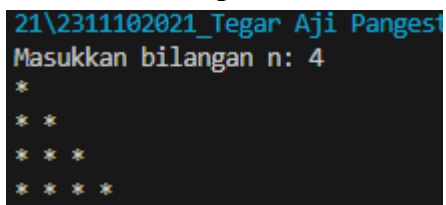
```
package main
import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan bilangan n: ")
    fmt.Scan(&n)

    fmt.Print("\n")
    cetakbintang(n)
}
```

Screenshot Output



```
21\2311102021_Tegar Aji Pangesti
Masukkan bilangan n: 4
*
* *
* * *
* * * *
```

Deskripsi Program

Program Go ini dirancang untuk mencetak pola bintang segitiga siku-siku dengan menggunakan rekursi. Pola bintang ini akan memiliki n baris, dengan setiap baris berisi n bintang.

VII . UNGUIDED

Soal Studi Case

Sourcecode

```
package main
import (
    "fmt"
)

func findFactors(N, divisor int) {

    if divisor > N {
        return
    }

    if N%divisor == 0 {
        fmt.Printf("%d ", divisor)
    }

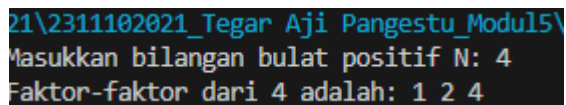
    findFactors(N, divisor+1)
}

func main() {
    var N int

    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    fmt.Printf("Faktor-faktor dari %d adalah: ", N)
    findFactors(N, 1)
    fmt.Println()
}
```

Screenshot Output



```
21\2311102021_Tegar Aji Pangestu_Modul5\
Masukkan bilangan bulat positif N: 4
Faktor-faktor dari 4 adalah: 1 2 4
```

Deskripsi Program

Program Go ini dirancang untuk menentukan dan mencetak faktor-faktor dari bilangan bulat positif yang diberikan N. Program ini menggunakan pendekatan rekursif untuk menghitung dan menampilkan faktor-faktor ini secara efisien

VIII . UNGUIDED

Soal Studi Case

Sourcecode

```
package main
import (
    "fmt"
)

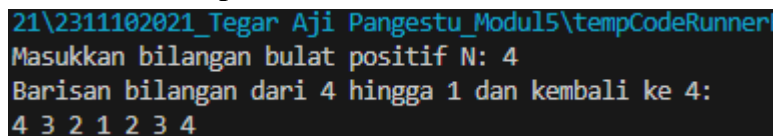
func printSeries(N, current int) {
    if current == 1 {
        fmt.Printf("%d ", current)
        return
    }

    fmt.Printf("%d ", current)
    printSeries(N, current-1)
    fmt.Printf("%d ", current)
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    fmt.Printf("Barisan bilangan dari %d hingga 1 dan kembali
ke %d:\n", N, N)
    printSeries(N, N)
    fmt.Println()
}
```

Screenshot Output



```
21\2311102021_Tegar Aji Pangestu_Modul5\tempCodeRunner\
Masukkan bilangan bulat positif N: 4
Barisan bilangan dari 4 hingga 1 dan kembali ke 4:
4 3 2 1 2 3 4
```

Deskripsi Program

Program Go ini dirancang untuk mencetak serangkaian angka dari bilangan bulat positif yang diberikan N hingga 1, lalu kembali ke N. Program ini menggunakan pendekatan rekursif untuk mencapai pola ini.

IX. UNGUIDED

Soal Studi Case

Sourcecode

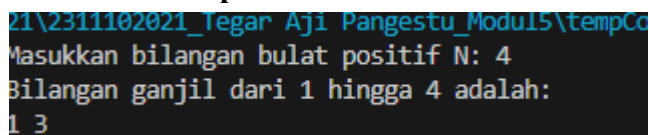
```
package main
import (
    "fmt"
)

func printOddNumbers(N, current int) {
    if current > N {
        return
    }
    if current%2 != 0 {
        fmt.Printf("%d ", current)
    }
    printOddNumbers(N, current+1)
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    fmt.Printf("Bilangan ganjil dari 1 hingga %d adalah:\n", N)
    printOddNumbers(N, 1)
    fmt.Println()
}
```

Screenshot Output



```
21\2311102021_Tegar Aji Pangestu_Modul5\tempCo
Masukkan bilangan bulat positif N: 4
Bilangan ganjil dari 1 hingga 4 adalah:
1 3
```

Deskripsi Program

Program Go ini dirancang untuk mencetak angka ganjil dari 1 hingga bilangan bulat positif tertentu N. Program ini menggunakan pendekatan rekursif untuk mengulang angka secara efisien dan mengidentifikasi angka ganjil.

X. UNGUIDED

Soal Studi Case

Sourcecode

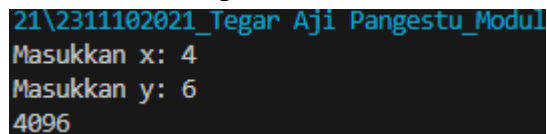
```
package main
import "fmt"

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

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

    fmt.Println(pangkat(x, y))
}
```

Screenshot Output



21\2311102021_Tegar Aji Pangestu_Modul
Masukkan x: 4
Masukkan y: 6
4096

Deskripsi Program

Program Go ini menghitung pangkat suatu bilangan x dipangkatkan y menggunakan pendekatan rekursif.