

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSIF**



Disusun Oleh :

Rafi Bintang Maulana / 2311102327

Kelas IF-11-06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Rekursif dalam pemrograman komputer adalah teknik di mana sebuah fungsi memanggil dirinya sendiri secara berulang untuk menyelesaikan suatu tugas. Teknik ini sangat berguna dalam pemecahan masalah yang dapat diuraikan menjadi sub-masalah yang mirip dengan masalah aslinya.

Rekursif memiliki keunggulan dalam menulis kode yang lebih sederhana, terutama untuk masalah-masalah yang melibatkan struktur data yang berulang, seperti pohon atau graf.

II. GUIDED

1. Membuat baris bilangan dari n hingga 1
Base-case: bilangan == 1

Sourcecode

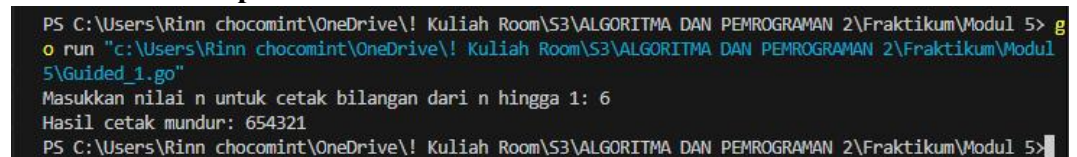
```
package main

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 C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Guided_1.go"
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 6
Hasil cetak mundur: 654321
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>
```

Deskripsi Program

Menggunakan fungsi rekursif untuk mencetak bilangan secara mundur dari nilai yang diberikan pengguna hingga 1, Program meminta input dari pengguna berupa bilangan bulat n. Kemudian, melalui fungsi “**cetakMundur**”, program mencetak setiap bilangan dari n hingga 1.

2. Menghitung hasil penjumlahan 1 hingga n
Base-case: n == 1

Sourcecode

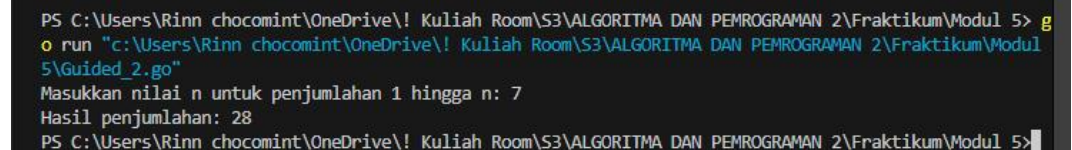
```
package main
import "fmt"
```

```
// Fungsi untuk menghitung penjumlahan 1 hingga n

func jumlahRekursif(n int) int {
    if n == 1 {
        return 1
    }
    return n + jumlahRekursif(n-1)
}

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

Screenshoot Output



```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Guided_2.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 7
Hasil penjumlahan: 28
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>
```

Deskripsi Program

Sama seperti sebelumnya, Program ini menggunakan fungsi rekursif untuk menghitung penjumlahan. Fungsi “**jumlahRekursif**” menerima parameter “**n**” dan memiliki basis kasus yang mengembalikan 1 jika “**n**” sama dengan 1. Jika “**n**” lebih besar dari 1, fungsi akan menjumlahkan nilai n dengan hasil pemanggilan fungsi “**jumlahRekursif**” dengan parameter “**n-1**”.

3. Mencari dua pangkat n atau 2"
Base-case: $n == 0$

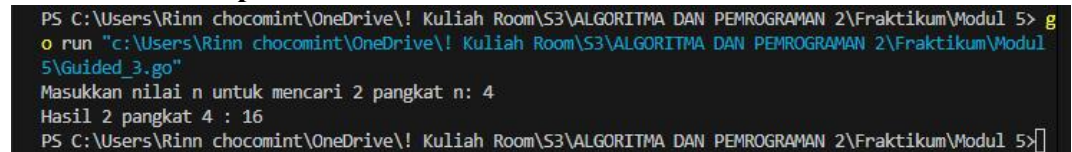
Sourcecode

```
package main
import "fmt"

// Fungsi untuk mencari 2 pangkat 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 C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Guided_3.go"
Masukkan nilai n untuk mencari 2 pangkat n: 4
Hasil 2 pangkat 4 : 16
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>
```

Deskripsi Program

Pada program, fungsi “**pangkatDua**” memiliki basis kasus ketika $n = 0$, di mana fungsi akan mengembalikan nilai 1, sesuai dengan sifat bilangan bahwa setiap bilangan berpangkat 0 adalah 1. Jika n lebih besar dari 0, fungsi mengembalikan hasil perkalian 2 dengan code “**pangkatDua (n-1)**”, yang akan terus menghitung hingga mencapai basis kasus.

4. Mencari nilai faktorial atau $n!$
Base-case: $n == 0$ atau $n == 1$

Sourcecode

```
package main

import "fmt"

// Fungsi untuk menghitung faktorial ni
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 C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Guided_3.go"
Masukkan nilai n untuk mencari 2 pangkat n: 4
Hasil 2 pangkat 4 : 16
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>
```

Deskripsi Program

Fungsi “**faktorial**” pada program tersebut memiliki basis kasus di mana jika $n = 0$ atau $n = 1$, fungsi akan mengembalikan nilai 1, sesuai dengan definisi faktorial bahwa $0!$ dan $1!$ adalah 1. Jika n lebih besar dari 1, fungsi akan mengembalikan hasil perkalian n dengan code “**faktorial(n-1)**”, yang akan terus menghitung hingga mencapai basis kasus.

III. UNGUIDED

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

Sourcecode

```
package main
import "fmt"

func DeretFibonacci(n int) int {
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    }
    return DeretFibonacci(n-1) + DeretFibonacci(n-2)
}

func main() {
    var n int
    fmt.Print("Cari deret Fibonacci, Masukkan nilai n : ")
    fmt.Scanln(&n)
    fmt.Println("Fibonacci ke", n, ":", DeretFibonacci(n))
}
```

Screenshoot Output

```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Unguided_1.go"
Cari deret Fibonacci, Masukkan nilai n : 10
Fibonacci ke 10 : 55
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> |
```

Deskripsi Program

Program ini juga menggunakan rekursif untuk menghitung nilai suku N. Fungsi “**DeretFibonacci**” menerima input bilangan bulat n dan mengembalikan nilai Fibonacci ke-n dengan aturan dasar. Jika $n = 0$, hasilnya 0. Dan Jika $n = 1$, hasilnya 1. Untuk nilai n lebih dari 1, fungsi akan memanggil dirinya sendiri dengan nilai $n - 1$ dan $n - 2$, menjumlahkan hasilnya untuk menghasilkan bilangan Fibonacci ke-n.

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

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

Sourcecode

```
package main
import "fmt"

func main() {
    var n int
    fmt.Print("Masukan N, Untuk pola bintang: ")
    fmt.Scanln(&n)

    for i := 1; i <= n; i++ {
        for j := 1; j <= i; j++ {
            fmt.Print("*")
        }
        fmt.Println()
    }
}
```

```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Unguided_2.go"
Masukan N, Untuk pola bintang: 8
*
**
***
****
*****
*****
*****
*****
*****
*****
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>
```

Program itu mencetak pola bintang yang tingginya ditentukan nilai n, Nilai N itu sendiri merupakan nilai yang di inputkan pengguna. Program menggunakan dua loop untuk mencetak pola. Loop pertama mengontrol jumlah baris dari 1 hingga n, lalu loop kedua mencetak bintang sebanyak nomor baris.

- Masukan** terdiri dari sebuah bilangan bulat positif N.

Contoh masukan dan keluaran:

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

```
package main
import "fmt"

func FaktorBilanganDari_N(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
}
```



```

    }
    FaktorBilanganDari_N(n, i+1)
}

func main() {
    var n int
    fmt.Print("Input bilangan bulat positif N: ")
    fmt.Scanln(&n)
    fmt.Print("Bilangan yang jadi faktor dari N = ", n )
    fmt.Println()

    fmt.Print("Yaitu : ")
    FaktorBilanganDari_N(n, 1)
    fmt.Println()
}

```

Screenshoot Output

```

PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Unguided 3.go"
Input bilangan bulat positif N: 12
Bilangan yang jadi faktor dari N = 12
Yaitu : 1 2 3 4 6 12
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>

```

Deskripsi Program

Program mencari dan mencetak semua faktor dari bilangan bulat positif n yang dimasukkan oleh pengguna. Setelah menerima input n, fungsi “**FaktorBilanganDari_N**” digunakan untuk memeriksa setiap bilangan dari 1 hingga n, Untuk menentukan apakah bilangan tersebut merupakan faktor dari n.

4. Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu.

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan dari N hingga 1 dan kembali ke N.

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

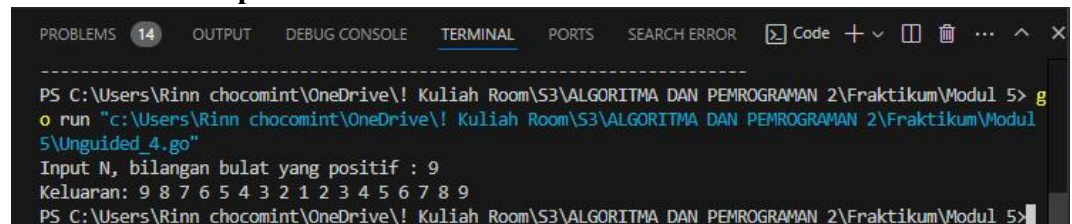
Sourcecode

```
package main
import "fmt"

func cetakBilanganDari_N_hingga_1_danKembaliKe_N(n int) {
    if n <= 0 {
        return
    }
    fmt.Print(n, " ")
    cetakBilanganDari_N_hingga_1_danKembaliKe_N(n - 1)
    if n > 1 {
        fmt.Print(n, " ")
    }
}

func main() {
    var n int
    fmt.Print("Input N, bilangan bulat yang positif : ")
    fmt.Scanln(&n)
    fmt.Print("Keluaran: ")
    cetakBilanganDari_N_hingga_1_danKembaliKe_N(n)
    fmt.Println()
}
```

Screenshoot Output



```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5\Unguided 4.go"
Input N, bilangan bulat yang positif : 9
Keluaran: 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>
```

Deskripsi Program

Program ini mencetak bilangan secara mundur dari N yang merupakan sebuah nilai yang di inputkan pengguna, hingga ke 1. Lalu kembali naik dari 1 hingga N lagi. Setelah menerima input bilangan bulat positif N, program akan memanggil fungsi “**cetakBilanganDari_N_hingga_1_danKembaliKe_N**” yang bertugas mencetak deretan angka tersebut secara rekursif.

5. Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan ganjil dari 1 hingga N.

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

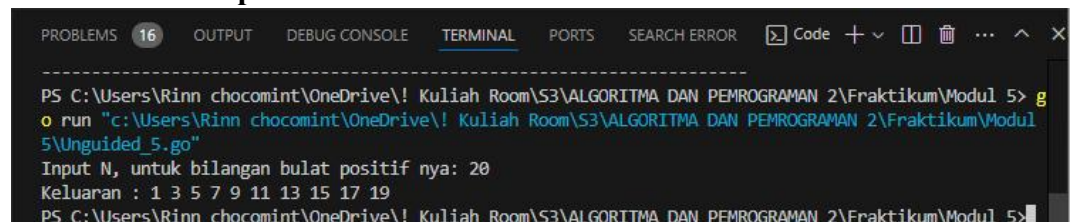
Sourcecode

```
package main
import "fmt"

func printBarisanBilanganGanjil_dari_1_hingga_N(n, i int) {
    if i > n {
        return
    }
    if i%2 != 0 {
        fmt.Print(i, " ")
    }
    printBarisanBilanganGanjil_dari_1_hingga_N(n, i+1)
}

func main() {
    var n int
    fmt.Print("Input N, untuk bilangan bulat positif nya: ")
    fmt.Scanln(&n)
    fmt.Print("Keluaran : ")
    printBarisanBilanganGanjil_dari_1_hingga_N(n, 1)
    fmt.Println()
}
```

Screenshoot Output



```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Unguided_5.go"
Input N, untuk bilangan bulat positif nya: 20
Keluaran : 1 3 5 7 9 11 13 15 17 19
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>
```

Deskripsi Program

Program itu mencetak deret bilangan ganjil dari 1 hingga n, n itu bilangan bulat positif yang nanti di input oleh pengguna diawal. Fungsi

“**printBarisanBilanganGanjil_dari_1_hingga_N**” menerima parameter variabel n dan i, dimulai dari 1, lalu memeriksa apakah i bilangan ganjil. Jika iya, maka i dicetak. Hasilnya deret bilangan ganjil dari 1 hingga n, yang dicetak dalam satu baris.

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

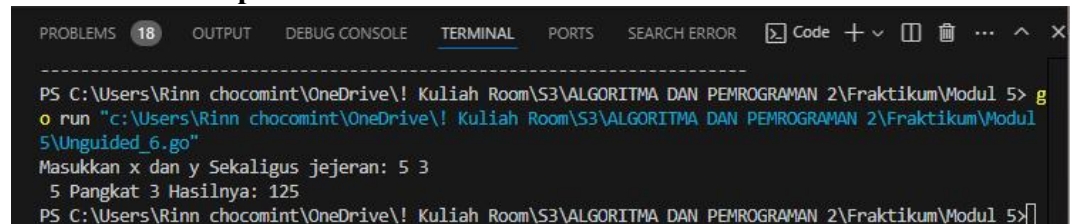
Sourcecode

```
package main
import "fmt"

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

func main() {
    var x, y int
    fmt.Println("Masukkan x dan y Sekaligus, Spasi jejeran: ")
    fmt.Scanf("%d %d", &x, &y)
    fmt.Printf(" %d Pangkat %d Hasilnya: %d\n", x, y, FungsiPangkat_y(x, y))
}
```

Screenshot Output



```
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5> g
o run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul
5\Unguided_6.go"
Masukkan x dan y Sekaligus jejeran: 5 3
5 Pangkat 3 Hasilnya: 125
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 5>
```

Deskripsi Program

Program tersebut untuk menghitung nilai pangkat dari bilangan bulat. Fungsi “**FungsiPangkat_y**” terima parameter x untuk bilangan dasar nya dan y untuk yang pangkat. Jika y sama dengan 0, fungsi mengembalikan 1, Jika tidak, fungsi mengalikan x dengan hasil pangkat dari x dengan y-1.