

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 5
REKRUSIF**



Oleh:

GALIH TRISNA

2311102050

IF-11-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

Rekursif adalah sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan tugas tertentu. Rekursif sangat berguna ketika masalah dapat dibagi menjadi sub-masalah yang lebih kecil dengan pola yang sama. Contohnya dari penggunaan rekursif yaitu untuk menghitung faktorial, di mana faktorial dari suatu bilangan n adalah hasil kali dari n dengan faktorial dari $n-1$, hingga mencapai nilai dasar (base case) yaitu 1.

Strukturnya

```
func [nama_fungsi]([parameter_list]) [tipe_pengembalian] {  
    // base case  
    if kondisi_berhenti {  
        return nilai_dasar  
    }  
    // rekursif  
    return [pemanggilan_fungsi_dengan_modifikasi_parameter]  
}
```

Contoh

```
package main  
import "fmt"  
  
// Fungsi rekursif untuk menjumlahkan angka dari 1 hingga n  
func sum(n int) int {  
    if n == 0 {  
        return 0 // Base case: jika n adalah 0, hasilnya adalah 0  
    }  
    return n + sum(n-1) // Panggil fungsi dengan n-1, dan tambahkan n  
}  
  
func main() {  
    fmt.Println("Jumlah dari 1 hingga 5 adalah:", sum(5)) // Output:  
    15  
}
```

II. GUIDED

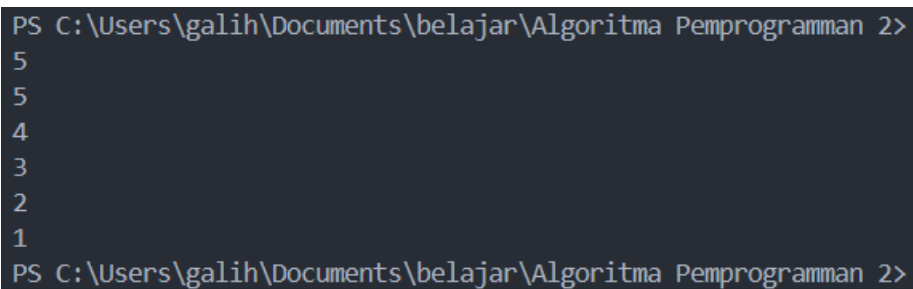
1. Source Code

```
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



```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
5
5
4
3
2
1
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas adalah program yang mencetak bilangan dari n hingga 1 secara menurun menggunakan rekursif. Program dimulai dengan meminta pengguna untuk memasukkan sebuah angka n, yang kemudian disimpan dalam variabel n. Fungsi baris dipanggil dengan parameter bilangan, yang merupakan nilai dari n. Di dalam fungsi baris, terdapat logika rekursif yang memeriksa apakah bilangan sama dengan 1. Jika ya, fungsi mencetak 1 dan berhenti, menandai kondisi dasar (base case). Namun, jika bilangan lebih besar dari 1, fungsi akan mencetak nilai bilangan saat ini,

dan kemudian memanggil dirinya sendiri dengan parameter bilangan - 1, sehingga proses ini berulang hingga mencapai 1. Sebagai contoh, jika pengguna memasukkan 5, output yang dihasilkan adalah deretan angka: 5, 4, 3, 2, 1, yang ditampilkan dalam urutan menurun.

2. Source Code

```
package main
import "fmt"

func main() {
    var n int
    fmt.Scan(&n)
    fmt.Println(penjumlah(n))
}

func penjumlah(n int) int {
    if n == 1 {
        return 1
    } else {
        return n + penjumlah(n-1)
    }
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
5
15
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas adalah program yang dapat menghitung jumlah semua bilangan bulat dari 1 hingga n menggunakan rekursif. Program dimulai dengan mendeklarasikan variabel n dan pengguna menginputkan nilai yang kemudian disimpan ke dalam variabel tersebut. Fungsi penjumlah kemudian dipanggil dengan parameter n, yang akan mengembalikan hasil penjumlahan. Di dalam fungsi penjumlah, terdapat

rekursif yang memeriksa apakah n sama dengan 1. Jika ya, fungsi mengembalikan 1, menandai kondisi dasar (base case). Jika tidak, fungsi akan mengembalikan hasil penjumlahan dari n dan memanggil dirinya sendiri dengan parameter $n-1$, sehingga proses ini berulang hingga mencapai 1.

III. UNGUIDED

1. Source Code

```
package main
import "fmt"

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

func deretFobonacci(n int, i int) {
    if i > n {
        return
    } else {
        fmt.Print(" ", fibonacci(i))
        deretFobonacci(n, i+1)
    }
}

func main() {
    var n int
    fmt.Scan(&n)
    deretFobonacci(n, 0)
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
10
0 1 1 2 3 5 8 13 21 34 55
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas adalah program yang akan mencetak deret Fibonacci hingga angka ke-n menggunakan rekursif. Program dimulai dengan mendeklarasikan fungsi fibonacci,

yang menghitung angka Fibonacci ke-n. Fungsi ini memeriksa apakah n kurang dari atau sama dengan 1. jika ya, maka mengembalikan n sebagai hasilnya (yaitu 0 untuk n = 0 dan 1 untuk n = 1). Jika tidak, fungsi akan memanggil dirinya sendiri dua kali dengan parameter n-1 dan n-2, lalu menjumlahkannya. Fungsi deretFobonacci berfungsi untuk mencetak deret Fibonacci dari 0 hingga n, di mana fungsi ini memeriksa apakah i lebih besar dari n. Jika tidak, ia mencetak nilai Fibonacci untuk indeks i dan memanggil dirinya sendiri dengan i yang meningkat. Pada fungsi main, program meminta input pengguna untuk nilai n dan kemudian memanggil deretFobonacci untuk menghasilkan deret Fibonacci.

2. Source Code

```
package main
import "fmt"

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

func baris(n int, i int) {
    if i > n {
        return
    }
    bintang(i)
    fmt.Println()
    baris(n, i+1)
}

func main() {
    var n int
    fmt.Scan(&n)
    baris(n, 1)
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
6
*
**
***
****
*****
*****
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas adalah program yang mencetak pola bintang berbentuk segitiga dengan jumlah baris sesuai input pengguna menggunakan rekursif. Program dimulai dengan mendeklarasikan fungsi bintang, yang bertugas mencetak bintang (*) sebanyak n kali. Jika nilai n sama dengan 0, fungsi akan berhenti (base case). Jika tidak, fungsi mencetak satu bintang dan kemudian memanggil dirinya sendiri dengan parameter n - 1 untuk mencetak bintang berikutnya. Fungsi baris yaitu mengatur pencetakan baris, di mana ia memeriksa apakah i lebih besar dari n. Jika tidak, fungsi memanggil bintang(i) untuk mencetak i bintang. Kemudian menggunakan `fmt.Println()` untuk berpindah ke baris yang baru. Setelah itu, baris memanggil dirinya sendiri dengan i yang increment untuk melanjutkan ke baris berikutnya. Pada fungsi main, program meminta input pengguna untuk menentukan jumlah baris n, kemudian memanggil fungsi baris untuk menghasilkan pola bintang.

3. Source Code

```
package main
import "fmt"

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

func main() {
    var n int
    fmt.Scan(&n)
    faktor(n, 1)
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
10
1 2 5 10
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
5
1 5
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas adalah program yang mencetak semua faktor dari sebuah bilangan n menggunakan rekursif. Program dimulai dengan mendeklarasikan fungsi faktor, yang menerima dua parameter yaitu n (bilangan yang akan dicari faktornya) dan i (indeks saat ini untuk memeriksa apakah i adalah faktor dari n). Fungsi ini memeriksa apakah i lebih besar dari n. jika ya, fungsi berhenti (base case). Selanjutnya, jika n dapat dibagi habis oleh i ($n \% i == 0$), maka i dicetak sebagai faktor. Setelah itu, fungsi faktor memanggil

dirinya sendiri dengan i yang ditambah satu, untuk melanjutkan pemeriksaan faktor berikutnya. Pada fungsi main, program meminta input pengguna untuk bilangan n , kemudian memanggil fungsi faktor untuk menghasilkan deret faktor dari n .

4. Source Code

```
package main
import "fmt"

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

func main() {
    var n int
    fmt.Scan(&n)
    barisan(n)
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
5
5 4 3 2 1 2 3 4 5
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas adalah program yang mencetak deret angka dari n hingga 1 dan kemudian kembali lagi ke n menggunakan rekursif. Program dimulai dengan mendeklarasikan fungsi barisan, yang menerima satu parameter, n . Jika nilai n sama dengan 1, fungsi mencetak 1 dan berhenti (base case). Jika tidak, fungsi akan mencetak nilai n dan kemudian memanggil dirinya sendiri dengan parameter $n - 1$. Setelah mencapai base case, fungsi mencetak nilai n lagi setelah panggilan rekursif, sehingga

menghasilkan deret yang naik kembali ke nilai semula. Pada fungsi main pengguna diminta untuk menginputkan nilai n, lalu memanggil fungsi barisan untuk mencetak deret tersebut

5. Source Code

```
package main
import "fmt"

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

func main() {
    var n int
    fmt.Scan(&n)
    ganjil(n, 1)
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
10
1 3 5 7 9
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
5
1 3 5
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas adalah program yang mencetak semua bilangan ganjil dari 1 hingga n menggunakan rekursif. Program dimulai dengan mendeklarasikan fungsi ganjil, yang menerima dua parameter yaitu n (batas atas bilangan yang akan diperiksa) dan i (bilangan saat ini yang sedang diperiksa). Fungsi ganjil memeriksa apakah i lebih besar dari n; jika ya, fungsi berhenti (base case). Jika tidak, fungsi akan memeriksa apakah i adalah bilangan ganjil dengan $i\%2 \neq 0$. Jika iya, i dicetak. Setelah itu, fungsi ganjil memanggil dirinya sendiri dengan parameter $i+1$. Pada fungsi main, program meminta pengguna untuk menginputkan nilai n, lalu memanggil fungsi ganjil untuk mencetak semua bilangan ganjil dari 1 sampai n.

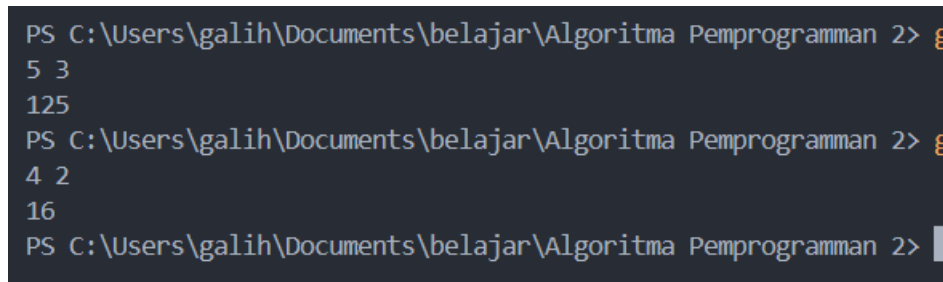
6. Source Code

```
package main
import "fmt"

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

func main() {
    var x, y int
    fmt.Scanln(&x, &y)
    println(pangkat(x,y))
}
```

Screenshot



```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> g
5 3
125
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> g
4 2
16
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas adalah program yang menghitung dan mencetak hasil dari x pangkat y menggunakan rekursif. Program dimulai dengan mendeklarasikan fungsi pangkat, yang menerima dua parameter yaitu x (angka yang akan dipangkatkan) dan y (nilai pangkat). Fungsi memeriksa apakah y sama dengan 1. jika ya, fungsi mengembalikan nilai x . Jika tidak, fungsi akan mengalikan x dengan hasil dari pemanggilan fungsi pangkat dengan parameter x dan $y-1$. Pada fungsi main, program meminta penggunaan untuk menginputkan nilai x dan y . Kemudian, fungsi pangkat dipanggil untuk menghitung hasil perpangkatan.