

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 6

REKURSIF



Disusun Oleh :

Natasya Intan Sukma Jiwanti / 2311102279

S1-IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

DASAR TEORI

Rekursif merupakan sebuah teknik dalam pemrograman dimana fungsi memanggil dirinya sendiri untuk menyelesaikan tugas tertentu. Pada rekursif terdapat dua bagian penting, yakni base case yang merupakan kondisi persyaratan untuk fungsi berhenti memanggil dirinya sendiri dan rekursif case yang merupakan langkah – langkah atau cara agar mencapai base case.

Perulangan menggunakan rekursif dengan perulangan menggunakan looping merupakan dua hal yang berbeda. Berikut merupakan perbandingan antara keduanya:

Perulangan	Kekurangan	Kelebihan
Looping	Mebutuhkan variable tambahan untuk iterasi	Mudah dimengerti dan implementasinya lebih langsung
Rekursif	Mebutuhkan pemahaman lebih dalam dan berpotensi memakan banyak memori jika tidak dikelola dengan baik	Solusi yang elegan untuk beberapa masalah kompleks, terutama yang memiliki struktur berulang.

Bagian – bagian pada rekursif:

- Base Case: Kondisi di mana fungsi rekursif berhenti memanggil dirinya sendiri. Ini mencegah fungsi dari pemanggilan tak terbatas. Biasanya pada kondisi base case kita akan menuliskan return untuk kembali ke nilai sebelumnya dan memberhentikan pemanggilan berulang.
- Recursive Case: Langkah-langkah yang diperlukan untuk mencapai base case. Fungsi rekursif memanggil dirinya sendiri dengan argumen yang berbeda untuk mendekati base case.

GUIDED

1. Guided 1

Soal Studi Case

Buatlah program yang digunakan untuk membuat baris bilangan dari n hingga 1

Sourcecode

```
package main
import "fmt"

func baris(bilangan int) {
    if bilangan == 1 {
        fmt.Println(1)
    } else {
        fmt.Println(bilangan)
        baris(bilangan - 1)
    }
}

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

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\guided1.go"
5
5
4
3
2
1
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas merupakan program yang mengimplementasikan fungsi rekursif untuk mencetak bilangan – bilangan sebelum n dengan memanggil fungsi baris dengan kondisi parameter (n-1) dan batas perulangan atau base case nya adalah nilai $n = 1$.

2. Guided 2

Soal Studi Case

Buatlah program yang digunakan untuk menghitung hasil penjumlahan 1 hingga n

Sourcecode

```
package main
import "fmt"

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

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

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\guided2.go"
7
28
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas merupakan program yang mengimplementasikan fungsi rekursif yang digunakan untuk melakukan penjumlahan berulang kali antara n dengan n-1 hingga n bernilai sama dengan 1.

3. Guided 3

Soal Studi Case

Buatlah program yang digunakan untuk mencari dua pangkat n

Sourcecode

```
package main
import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan Nilai n : ")
    fmt.Scan(&n)
    fmt.Println("Hasil dari 2 pangkat", n, "adalah",
    pangkat(n))
}
```

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\guided3.go"
Masukkan Nilai n : 5
Hasil dari 2 pangkat 5 adalah 32
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas merupakan program yang menggunakan implementasi rekursif untuk mengalikan bilangan 2 dengan bilangan 2 sebanyak n kali atau untuk mencari hasil dari 2 pangkat n. Base case dari rekursif pada program diatas adalah apabila n bernilai sama dengan 0.

4. Guided 4

Soal Studi Case

Buatlah program yang digunakan untuk mencari nilai faktorial dari n!

Sourcecode

```
package main
import "fmt"

func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    } else {
        return n * faktorial(n-1)
    }
}

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

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\tempCodeRunnerFile.go"
9
362880
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas merupakan program yang menggunakan fungsi rekursif untuk mencari nilai faktorial dari bilangan n dengan cara mengalikan bilangan n dengan n-1 hingga nilai n bernilai sama dengan 1.

I. UNGUIDED

1. Unguided 1

Soal Study Case

Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci dengan formula $S(n) = S(n-1) + S(n-2)$

Sourcecode

```
package main
import "fmt"

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

func main(){
    var n int
    fmt.Print("Masukkan Bilangan: ")
    fmt.Scan(&n)
    fmt.Printf("Fibonacci dari bilangan %d adalah: %d\n", n
, deretFibonacci(n))
}
```

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\unguided1.go"
Masukkan Bilangan: 10
Fibonacci dari bilangan 10 adalah: 55
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\unguided1.go"
Masukkan Bilangan: 7
Fibonacci dari bilangan 7 adalah: 13
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas merupakan yang menggunakan fungsi rekursif yang memiliki dua base case yakni apabila n bernilai 0 akan mengembalikan nilainya 0, lalu apabila n bernilai 1 maka akan mengembalikan nilainya ke 1. Kedua basecase ini penting supaya fungsi dapat berhenti melakukan

rekursi. Hal tersebut juga dikarenakan rekursif case dalam fungsi tersebut mengembalikan nilai dari deret fibonacci (n-1) ditambah nilai dari deret fibonacci (n-2).

2. Unguided 2

Soal Studi Case

Buatlah sebuah program yang digunakan untuk menampilkan pola bintang dengan menggunakan fungsi rekursif.

Sourcecode

```
package main
import "fmt"

func polaBintang(n int){
    for i:= 0; i < n; i++){
        fmt.Print("* ")
    }
    fmt.Println()
}

func printBintang(n, i int){
    if i > n{
        return
    }
    polaBintang(i)
    printBintang(n, i + 1)
}

func main() {
    var n int
    fmt.Print("Masukkan Bilangan: ")
    fmt.Scan(&n)
    printBintang(n, 1)
}
```

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\unguided2.go"
Masukkan Bilangan: 7
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
PS D:\Praktikum Alpro\praktikum 4> █
```


Deskripsi Program

Program diatas memiliki dua fungsi yakni fungsi pola bintang yang akan mencetak banyak bintang dalam satu baris menggunakan looping, dan fungsi print bintang yang akan mencetak bintang n baris menggunakan fungsi rekursif dengan batas rekursif nya adalah variabel i bernilai lebih dari n.

3. Unguided 3

Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu N, atau bilangan apa saja yang habis membagi N.

Sourcecode

```
package main
import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan Bilangan: ")
    fmt.Scan(&n)
    fmt.Print("Faktor dari bilangan ", n, " adalah ")
    carifaktor(n, 1)
    fmt.Println()
}
```

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\unguided3.go"
Masukkan Bilangan: 7
Faktor dari bilangan 7 adalah 1 7
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\unguided3.go"
Masukkan Bilangan: 9
Faktor dari bilangan 9 adalah 1 3 9
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas menggunakan fungsi rekursif untuk mencetak faktor dari sebuah bilangan n dengan cara membagi n secara berulang kali dengan nilai dari variabel i , yang nilainya akan terus bertambah 1 setiap kali rekursif. Rekursif akan berhenti memanggil dirinya sendiri apabila nilai variabel i sudah melebihi nilai n .

4. Unguided 4

Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu, yakni bilangan N hingga 1 lalu kembali ke N

Sourcecode

```
package main
import "fmt"

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

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

```
func main() {
    var n int
    fmt.Print("Masukkan Bilangan: ")
    fmt.Scan(&n)
    fmt.Println()
    printmundur(n)
    printmaju(n, 2)
    fmt.Println()
}
```

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\tempCodeRunnerFile.go"
Masukkan Bilangan: 7

7 6 5 4 3 2 1 2 3 4 5 6 7
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas menggunakan dua fungsi rekursif, yakni:

- 1) Print Mundur
 - a. Basecasenya adalah nilai n sama dengan 1
 - b. Rekursif case nya adalah nilai n dikurangi 1 pada setiap iterasi
- 2) Print Maju
 - a. Basecasenya adalah nilai variabel i lebih besar dari n
 - b. Rekursif casenya adalah nilai variabel i bertambah 1 setiap iterasi, dimana nilai i bernilai 2 yang dideklarasikan pada fungsi utama. Hal tersebut dikarenakan program harus dapat mencetak deret dari bilangan n ke 1 lalu kembali ke n lagi tanpa ada perulangan bilangan 1. Jadi dalam deret tersebut jumlah bilangan 1 hanya 1.

5. Unguided 5

Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

Sourcecode

```
package main
import "fmt"

func cariganjil(a int, b int){
    if a > b{
        return
    }else{
        if a%2 == 1{
            fmt.Print(a, " ")
        }
        cariganjil(a + 1, b)
    }
}

func main(){
    var a, b int
    fmt.Print("Masukkan Bilangan A: ")
    fmt.Scan(&a)
    fmt.Print("Masukkan Bilangan B: ")
    fmt.Scan(&b)
    fmt.Print("Bilangan Ganjil dari", a, " hingga ", b , "
adalah: ")
    cariganjil(a, b)
}
```

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\unguided5.go"
Masukkan Bilangan A: 22
Masukkan Bilangan B: 79
Bilangan Ganjil dari 22 hingga 79 adalah: 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas menggunakan fungsi rekursif untuk mencari dan mencetak bilangan ganjil dari suatu deret yang dimulai dari bilangan a hingga bilangan b. Oleh karena itu, basecase pada fungsi rekursif adalah nilai a lebih besar dari nilai b, dan rekursif casenya adalah a+1.

6. Unguided 6

Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan.

Sourcecode

```
package main
import "fmt"

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

func main() {
    var x, y int
    fmt.Print("Masukkan X: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan Y: ")
    fmt.Scan(&y)
    hasil := hitungPangkat(x, y)
    fmt.Printf("Hasil Pangkat %d dengan %d adalah %d\n", x,
y, hasil)
}
```

Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 4> go run "d:\Praktikum Alpro\praktikum 4\tempCodeRunnerFile.go"
Masukkan X: 7
Masukkan Y: 9
Hasil Pangkat 7 dengan 9 adalah 40353607
PS D:\Praktikum Alpro\praktikum 4> █
```

Deskripsi Program

Program diatas menggunakan fungsi rekursif dimana basecasenya adalah y bernilai sama dengan 0 akan mengembalikan nilai nya menjadi 1 karena bilangan apapun berpangkat 0 akan menghasilkan nilai 1. Sedangkan

rekursif case pada fungsi diatas adalah memanggil fungsi hitung pangkat dengan parameter y berkurang 1 setiap kali beriterasi.