

**LAPORAN PRAKTIKUM PEMROGRAMAN
BERORIENTASI OBJEK
MODUL V
REKURSIF**



Oleh:

NAMA : CHRIST DANIEL SANTOSO

NIM : 2311102305 KELAS : IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM
PURWOKERTO**

2024

I. DASAR TEORI

Rekursif adalah teknik di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah. Fungsi rekursif biasanya terdiri dari dua bagian utama: **basis kasus** (base case), yang menentukan kapan fungsi harus berhenti memanggil dirinya sendiri, dan **rekursi** (recursive case), yang memecah masalah menjadi sub-masalah yang lebih kecil dan memanggil fungsi tersebut kembali. Rekursi sangat berguna untuk masalah yang memiliki sifat berulang atau dapat dipecah menjadi versi yang lebih sederhana dari masalah yang sama, seperti perhitungan faktorial, bilangan Fibonacci, atau traversing struktur data seperti pohon atau graf.

		Algo
1	procedure (x r)	func cetak(x int){ fmt.Println(x) cetak(x+1) }
2	algoritma	
3	output(x)	
4	cetak(x+1)	
5	endprocedure	

Catatan:

1. Teknik rekursif merupakan salah satu alternatif untuk menggantikan struktur kontrol perulangan dengan memanfaatkan subprogram, baik itu fungsi maupun prosedur.
2. Untuk menghentikan proses rekursif, digunakan struktur percabangan (if-then).
3. Base-case adalah kondisi di mana proses rekursif berhenti. Base-case adalah elemen paling penting dan harus diketahui terlebih dahulu ketika membuat program rekursif. Membuat program rekursif tanpa mengetahui base-case terlebih dahulu tidak mungkin dilakukan.
4. Recursive-case adalah kondisi di mana fungsi memanggil dirinya sendiri. Recursive-case merupakan kebalikan atau negasi dari base-case.
5. Setiap algoritma rekursif selalu memiliki bentuk setara dalam algoritma iteratif.

Komponen Rekursif:

Algoritma rekursif terdiri dari dua komponen utama:

- **Base-case (Basis):** Bagian yang menghentikan proses rekursif, yang menjadi komponen paling penting dalam rekursi.

- **Recursive-case:** Bagian di mana subprogram dipanggil kembali

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL

NO 1.

Source code:

```
C:\> Users > cynzw > OneDrive > Documents > -go guided 1 modul 5.go > baris
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n int
7      fmt.Scan(&n)
8      baris(n)
9  }
10
11
12 func baris(bilangan int) {
13     if bilangan == 1 {
14         fmt.Println(1)
15     } else {
16         fmt.Println(bilangan)
17         baris(bilangan - 1)
18     }
19 }
20
21
```

Output :

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\guided 1 modul 5.go"
5
5
4
3
2
1
PS C:\Users\cynzw> |
```

Penjelasan :

Program di atas meminta pengguna untuk memasukkan bilangan bulat n, kemudian mencetak bilangan tersebut secara menurun hingga mencapai 1 dengan menggunakan fungsi rekursif bernama baris. Fungsi baris(bilangan int) akan mencetak nilai bilangan, lalu memanggil dirinya sendiri dengan argumen bilangan - 1. Rekursi ini akan berhenti ketika bilangan mencapai nilai 1, di mana angka 1 akan dicetak, dan rekursi pun berakhir. Program ini menunjukkan cara menggunakan rekursi untuk mencetak bilangan secara menurun.

NO 2.

Source code:

```

C: > Users > cynzw > OneDrive > Documents > - guided 2 modul 5.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n int
7      fmt.Scan(&n)
8      fmt.Println(penjumlahan(n))
9  }
10
11
12  func penjumlahan(n int) int {
13      if n == 1 {
14          return 1
15      } else {
16          return n + penjumlahan(n-1)
17      }
18  }
19
20

```

Output :

```

PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\guided 2 modul 5.go"
10
55
PS C:\Users\cynzw>

```

Penjelasan :

Program di atas berfungsi untuk menghitung total penjumlahan dari angka 1 hingga n menggunakan metode rekursif. Pertama, pengguna diminta untuk memasukkan sebuah bilangan bulat positif n sebagai input. Setelah itu, fungsi rekursif penjumlahan(n) akan dipanggil untuk menghitung hasil penjumlahan. Fungsi ini bekerja dengan menjumlahkan nilai n saat ini dengan hasil dari pemanggilan rekursif fungsi yang sama, namun dengan argumen n-1. Proses rekursi ini terus berlanjut hingga mencapai kondisi dasar, yaitu ketika n sama dengan 1. Pada titik tersebut, fungsi akan mengembalikan nilai 1 dan proses rekursi berhenti. Setelah semua nilai dijumlahkan, hasil akhir penjumlahan total dari 1 hingga n akan dicetak sebagai output. Program ini menunjukkan bagaimana rekursi digunakan untuk menyelesaikan operasi penjumlahan bertingkat.

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL 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.

Source code :

```
C: > Users > cynzw > OneDrive > Documents > -go unguided 1 modul 5.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("Masukkan nilai n: ")
8     fmt.Scan(&n)
9     fibonacciSeries(n)
10 }
11
12 func fibonacciSeries(n int) {
13     a, b := 0, 1
14     for i := 0; i <= n; i++ {
15         fmt.Print(a, " ")
16         a, b = b, a+b
17     }
18     fmt.Println()
19 }
20
```

Output :

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 1 modul 5.go"
Masukkan nilai n: 10
0 1 1 2 3 5 8 13 21 34 55
PS C:\Users\cynzw>
```

Penjelasan :

Program di atas menerima input dari pengguna berupa bilangan bulat n dan menampilkan deret Fibonacci hingga elemen ke-n. Fungsi fibonacciSeries(n) mencetak setiap angka Fibonacci dari 0 hingga n, dengan memanggil fungsi rekursif fibonacci(n) yang menghitung nilai Fibonacci berdasarkan rumus dasar: jika n sama dengan 0 atau 1, maka hasilnya adalah n ; sedangkan untuk nilai yang lebih besar, hasilnya diperoleh dari penjumlahan dua angka Fibonacci sebelumnya (fibonacci(n-1) + fibonacci(n-2)).

SOAL 2

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

Source code :

```
C: > Users > cynzw > OneDrive > Documents > -go unguided 2 modul 5.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n int
7      fmt.Print("Masukkan nilai N: ")
8      fmt.Scan(&n)
9      printBintang(n)
10 }
11
12 func printBintang(n int) {
13     for i := 1; i <= n; i++ {
14         for j := 0; j < i; j++ {
15             fmt.Print("*")
16         }
17         fmt.Println()
18     }
19 }
20
```

Output :

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 2 modul 5.go"
Masukkan nilai N: 5
*
**
***
****
*****
PS C:\Users\cynzw>
```

Penjelasan :

Program ini bertujuan untuk mencetak pola bintang bertingkat berdasarkan input yang diberikan oleh pengguna. Setelah pengguna memasukkan nilai *n*, program memanfaatkan fungsi rekursif `printbintang(n)` untuk membentuk pola bintang. Fungsi `printbintang(n)` bekerja dengan memanggil dirinya sendiri untuk mencetak baris bintang sebelumnya, kemudian menambahkan baris bintang dengan panjang sesuai nilai *n*, yang dihasilkan oleh fungsi `cetakBintang(n)`. Proses ini berlanjut hingga mencapai kondisi dasar, yaitu ketika $n == 1$, di mana fungsi tersebut mengembalikan satu bintang.

SOAL 3

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

Source code :

```
C: > Users > cynzw > OneDrive > Documents > -go unguided 3 modul 5.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n int
7      fmt.Scan(&n)
8      fmt.Println("Faktor dari", n, "adalah:")
9      cetakFaktor(n)
10 }
11
12 func cetakFaktor(n int) {
13     for i := 1; i <= n; i++ {
14         if n%i == 0 {
15             fmt.Print(i, " ")
16         }
17     }
18     fmt.Println()
19 }
20
```

Output :

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 3 modul 5.go"
12
Faktor dari 12 adalah:
1 2 3 4 6 12
PS C:\Users\cynzw>
```

Penjelasan :

Program di atas mencari dan mencetak faktor-faktor dari bilangan bulat n yang dimasukkan oleh pengguna. Fungsi utama `main` menerima input n dan memanggil fungsi rekursif cetakFaktor, yang secara berulang memeriksa apakah bilangan i (dimulai dari 1) adalah faktor dari n (jika $n \% i == 0$). Jika ya, i dicetak. Proses ini berlanjut hingga semua faktor dari 1 hingga n ditemukan, kemudian program berhenti.

Source code :

SOAL 4

```
C: > Users > cynzw > OneDrive > Documents > -go unguided 4 modul 5.go > baris
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n int
7      fmt.Scan(&n)
8      baris(n)
9      fmt.Println()
10 }
11
12 func baris(bilangan int) {
13     for i := bilangan; i >= 1; i-- {
14         fmt.Print(i, " ")
15     }
16
17     for i := 2; i <= bilangan; i++ {
18         fmt.Print(i, " ")
19     }
20 }
21
```

Output :

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 4 modul 5.go"
5
5 4 3 2 1 2 3 4 5
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 4 modul 5.go"
10
10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10
PS C:\Users\cynzw>
```

Penjelasan :

Program ini menggunakan rekursi untuk mencetak pola angka berdasarkan input dari pengguna. Fungsi `baris(bilangan int)` mencetak angka mulai dari bilangan input hingga 1, kemudian kembali ke angka input. Ketika bilangan mencapai 1, program mencetak 1 dan melanjutkan untuk mencetak angka yang lebih besar hingga kembali ke bilangan awal. Misalnya, jika pengguna memasukkan angka 3, outputnya akan menjadi: 3 2 1 2 3.

SOAL 5

Source code :

```
C: > Users > cynzw > OneDrive > Documents > -go unguided 5 modul 5.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n int
7      fmt.Scan(&n)
8      barisGanjil(n)
9      fmt.Println()
10 }
11
12 func barisGanjil(bilangan int) {
13     for i := 1; i <= bilangan; i++ {
14         if i%2 != 0 {
15             fmt.Print(i, " ")
16         }
17     }
18 }
19
```

Output :

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 5 modul 5.go"
5
1 3 5
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 5 modul 5.go"
18
1 3 5 7 9 11 13 15 17
PS C:\Users\cynzw>
```

Penjelasan :

Program ini mencetak bilangan ganjil dari 1 hingga n menggunakan pendekatan rekursif. Pertama, pengguna diminta untuk memasukkan nilai n, dan fungsi baris Ganjil dipanggil dengan n sebagai argumen. Di dalam fungsi, jika bilangan kurang dari atau sama dengan 0, fungsi akan berhenti. Jika tidak, fungsi akan memanggil dirinya sendiri dengan argumen bilangan - 1, membentuk tumpukan rekursi. Setelah kembali dari pemanggilan rekursif, program memeriksa apakah bilangan adalah ganjil. Jika ya, bilangan tersebut dicetak. Dengan cara ini, bilangan ganjil dari 1 hingga n dicetak dalam urutan meningkat.

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".

SOAL 6

Source code :

```
C: > Users > cynzw > OneDrive > Documents > unguided 6 modul 5.go > ...  
1  package main  
2  
3  import "fmt"  
4  
5  func main() {  
6      var x, y int  
7      fmt.Scan(&x)  
8      fmt.Scan(&y)  
9  
10     fmt.Println(pangkat(x, y))  
11 }  
12  
13 func pangkat(x, y int) int {  
14     result := 1  
15     for i := 0; i < y; i++ {  
16         result *= x  
17     }  
18     return result  
19 }  
20
```

Output :

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 6 modul 5.go"  
3 4  
81  
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\unguided 6 modul 5.go"  
10 3  
1000  
PS C:\Users\cynzw>
```

Penjelasan :

Program ini merupakan implementasi sederhana dari fungsi rekursif untuk menghitung hasil dari bilangan x dipangkatkan dengan y. Di dalam fungsi main, program meminta pengguna untuk memasukkan dua bilangan bulat, x dan y. Setelah menerima input, program memanggil fungsi pangkat, yang melakukan perhitungan pangkat secara rekursif. Jika y sama dengan 0, fungsi akan mengembalikan 1 (karena setiap bilangan yang dipangkatkan 0 adalah 1). Jika tidak, fungsi akan mengalikan x dengan hasil dari pemanggilan pangkat(x, y-1) hingga mencapai kondisi dasar. Setelah itu, hasil dari perhitungan pangkat tersebut dicetak ke layar.