

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN LANJUTAN
PERTEMUAN KE – 11



Disusun Oleh :

NAMA : TARISA DWI SEPTIA
NIM : 205410126
JURUSAN : TEKNIK INFORMATIKA
JENJANG : S1

Laboratorium Terpadu
Sekolah Tinggi Management Informatika Komputer
AKAKOM
YOGYAKARTA
2021

REKURSIF

A. Tujuan

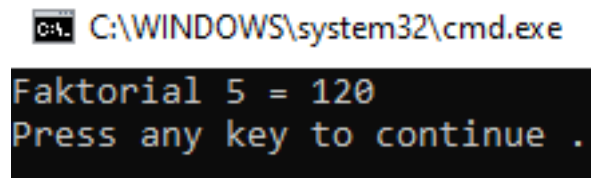
- Mahasiswa dapat memahami dan menyelesaikan kasus dengan rekursif

B. Praktik

1. Program faktorial dengan rekursif

```
public class Praktik1{
    //method rekursif
    public static long faktorial( long N ){
        if ( N <= 1 ) // kondisi terpenuhi untuk akhir rekursif
            return 1; // nilai akhir: 0! = 1 dan 1! = 1
        else // step rekursif, ada pemanggilan kembali method 'faktorial'
            return N * faktorial( N - 1 );
        }
    public static void main( String[] args ){
        // pemanggilan awal method faktorial
        System.out.println( "Faktorial 5 = "+ faktorial(5));
    }
}
```

Output :



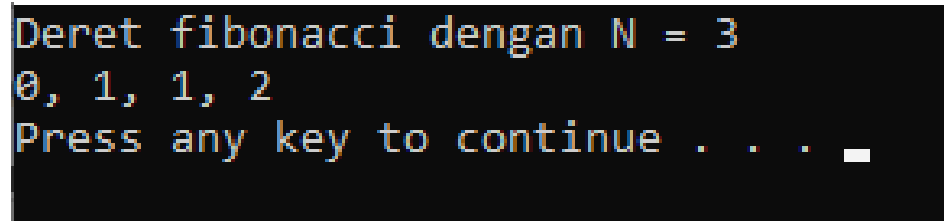
C:\WINDOWS\system32\cmd.exe

Faktorial 5 = 120
Press any key to continue .

2. Program fibonacci dengan rekursif

```
public class Praktik2{
    //method rekursif
    public static long fibonacci( long N ){
        if (( N == 0 ) || ( N == 1 )) // kondisi terpenuhi untuk akhir rekursif
            return N; // nilai akhir : N=0 atau N=1
        else // step rekursif, ada pemanggilan kembali method 'fibonacci'
            return fibonacci(N-1) + fibonacci(N-2);
        }
    public static void main( String[] args ){
        System.out.println("Deret fibonacci dengan N = 3");
        for (int i=0; i<=3;i++){
            // pemanggilan awal method fibonacci
            if (i<3)
                System.out.print(fibonacci(i)+"", " ");
            else
                System.out.println(fibonacci(i));
        }
    }
}
```

Output :



Deret fibonacci dengan N = 3
0, 1, 1, 2
Press any key to continue . . .

3. Modifikasi method faktorial diatas menjadi menampilkan semua nilai hasil faktorial dari bilangan-bilangan sebelumnya. Misalnya jika dari method main dipanggil untuk menghasilkan faktorial dari 5 maka akan ditampilkan faktorial 1, faktorial 2, faktorial 3, faktorial 4 dan terakhir faktorial 5.

```
import java.util.Scanner;
public class Praktik3{
    //method rekursif
    public static int faktorial( int N ){
        if ( N <= 1 ) // kondisi terpenuhi untuk akhir rekursif
            return 1; // nilai akhir: 0! = 1 dan 1! = 1
        else // step rekursif, ada pemanggilan kembali method 'faktorial'
            return N * faktorial( N - 1 );
        }
    public static void main( String[] args ){
        int bill;
        Scanner masuk = new Scanner(System.in);
        System.out.println("Masukan bilangan yang ingin di faktorialkan : ");
        bill = masuk.nextInt();
        for(int i = 1; bill >= i; bill--){
            int hasil = faktorial(bill);
            System.out.println("Faktorial "+bill+ " = "+hasil);
        }
    }
}
```

Output :

```
Masukan bilangan yang ingin di faktorialkan
6
Faktorial 6= 720
Faktorial 5= 120
Faktorial 4= 24
Faktorial 3= 6
Faktorial 2= 2
Faktorial 1= 1
Press any key to continue . . . _
```

C. Latihan

1. Buat program dengan method rekursif untuk menghitung nilai A^B dimana A dan B bilangan bulat yang lebih besar atau sama dengan 1. Deklarasi methodnya ada sebagai berikut pangkat(A,B)

```
public class Latihan1{
    public static int pangkat (int A, int B){
        if(B == 0){
            return 1;
        }else{
            return A * pangkat(A, B-1);
        }
    }

    public static void main (String [] args){
        System.out.println("Pangkat dari 5^4 = "+pangkat(5,4));
    }
}
```

Output :

```
Pangkat dari 5^4 = 625
Press any key to continue . . . _
```

2. Greatest Common Divisor (gcd) dari bilangan bulat x dan y adalah bilangan bulat terbesar yang terbagi rata menjadi x dan y. Buat algoritma dan program dengan method rekursif gcd yang mengembalikan reatest Common Divisor x dan y. Gcd x dan y didefinisikan secara rekursif sebagai berikut: Jika y sama dengan 0, maka gcd (x, y) adalah x; jika tidak, gcd (x, y) adalah gcd (y, x% y), di mana% adalah operator sisanya

```
import java.util.Scanner;
public class Latihan2{
    public static int FBB (int X, int Y){
        if(Y == 0){
            return X;
        }else{
            return FBB (Y, X % Y);
        }
    }

    public static void main (String [] args){
        Scanner masuk = new Scanner(System.in);
        System.out.println("Masukan bilangan pertama : ");
        int X = masuk.nextInt();
        System.out.println("Masukan bilangan kedua : ");
        int Y = masuk.nextInt();
        int hasil = FBB(X,Y);
        System.out.println("Hasil dari FBB "+X+" dan "+Y+" adalah "+hasil);
    }
}
```

Output :

```
Masukan bilangan pertama :
99
Masukan bilangan kedua :
33
Hasil dari FBB 99 dan 33 adalah 33
Press any key to continue . . . _
```

D. Tugas

1. Palindrome adalah string yang dieja dengan cara yang sama maju dan mundur. Beberapa contoh palindrom adalah "radar," "makam". Tulis method rekursif testPalindrome yang mengembalikan nilai boolean true jika string yang disimpan dalam array adalah palindrom dan false jika tidak. Metode ini harus mengabaikan spasi dan tanda baca dalam string.

```

import java.util.Scanner;

public class Tugas1 {
    private static Boolean polindrome(String kata) {
        // buat variabel untuk nampung kata baru
        String balik = "";
        // perulangan untuk buat kata baru tapi hurufnya diambil dari belakang
        // misal katanya adalah "uang"
        // maka setelah lewat ini variabel balik berisi "gnau"
        for (int i = kata.length(); i > 0; i--) {
            String a = kata.substring(i - 1, i);
            balik += a;
        }
        // cek apakah balik sama dengan kata
        // kalo sama berarti polindrome
        if (kata.equals(balik)) {
            return true;
        }
        // kalo engga berarti false
        return false;
    }

    public static void main(String[] args) {
        // deklarasi variabel yang diperlukan
        String kata;
        Scanner scan = new Scanner(System.in);

        // proses input user | memasukkan kata yang akan dicek
        System.out.print("Masukkan kata : ");
        kata = scan.nextLine();

        // menampilkan hasil
        System.out.println(polindrome(kata));
    }
}

```

output :

```

Masukkan kata : radar
true
Press any key to continue . . .

```

E. Kesimpulan

Setelah melakukan praktik di atas yang sesuai dengan tujuan modul 11 dapat disimpulkan bahwa mahasiswa dapat memahami dan menyelesaikan kasus dengan rekursif.