

LAPORAN XIII
PRAKTIKUM DASAR PEMROGRAMAN



1-B

Luthfi Triaswangga

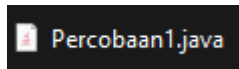
2341720208

Teknologi Informasi

Teknik Informatika

Percobaan 1

1. Pertama buatlah file terlebih dahulu dengan format **Percobaan1.java**



2. Kemudian masukkan perintah **import java.util.Scanner;** untuk memulai pemrograman.

```
import java.util.Scanner;  
public class Percobaan1{
```

3. Masukkan perintah **static int NamaFungsi (int n){** terlebih dahulu sebelum memulai pemrograman.

```
static int faktorialRekursif(int n){  
    if (n==0){  
        return (1);  
    } else {  
        return (n * faktorialRekursif(n-1));  
    }  
}
```

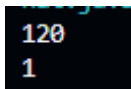
4. Lalu Masukkan perintah **public static void main(String args[]) { t**

```
public static void main(String[] args) {  
    System.out.println(faktorialRekursif(n:5));  
    System.out.println(faktorialIteratif(n:5));  
}
```

5. Kemudian mulailah pemrograman

```
import java.util.Scanner;  
public class Percobaan1{  
    static int faktorialRekursif(int n){  
        if (n==0){  
            return (1);  
        } else {  
            return (n * faktorialRekursif(n-1));  
        }  
    }  
    static int faktorialIteratif(int n){  
        int faktor = 1;  
        for (int i=n; i >= 1; i--){  
            faktor = faktor * i;  
        }  
        return faktor;  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        System.out.println(faktorialRekursif(n:5));  
        System.out.println(faktorialIteratif(n:5));  
    }  
}
```

6. Maka hasilnya akan seperti ini



120
1

Pertanyaan 1

1. Apa yang dimaksud dengan fungsi rekursif?

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri secara langsung atau tidak langsung. Fungsi rekursif dapat digunakan untuk menyelesaikan masalah tertentu yang lebih mudah diselesaikan dengan cara ini daripada dengan cara iteratif.

2. Bagaimana contoh kasus penggunaan fungsi rekursif ?

Perulangan :

Fungsi rekursif dapat digunakan untuk menggantikan perulangan iteratif. Misalnya, berikut adalah contoh fungsi rekursif untuk mencetak angka dari 1 hingga n.

Fungsi ini bekerja dengan cara mencetak n, kemudian memanggil dirinya sendiri dengan argumen $n - 1$. Hal ini dilakukan berulang kali hingga n menjadi 0.

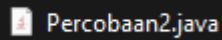
3. Pada Percobaan1, apakah hasil yang diberikan fungsi faktorialRekursif() dan fungsi faktorialIteratif() sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iteratif!

A. Fungsi faktorialRekrusif: Fungsi ini memanggil dirinya sendiri sampai mencapai kondisi dasar, yaitu jika $n == 0$, maka fungsi akan mengembalikan 1. Jika tidak, fungsi akan memanggil dirinya sendiri dengan argumen $n - 1$ dan mengalikan hasilnya dengan n. Proses ini akan berlanjut sampai n menjadi 0.

B. Fungsi Iteratif (faktorialIteratif): Fungsi ini menggunakan loop for untuk mengalikan setiap angka dari 1 sampai n. Tidak ada pemanggilan fungsi lagi setelah fungsi ini dipanggil.

Percobaan 2

1. Pertama buatlah file terlebih dahulu dengan format **Percobaan2.java**



2. Kemudian masukkan perintah **import java.util.Scanner;** untuk memulai pemrograman.

```
import java.util.Scanner;
public class Percobaan2 {
```

3. Masukkan perintah **static int NamaFungsi ()**{ terlebih dahulu sebelum memulai pemrograman.

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

4. Lalu Masukkan perintah **public static void main(String args[])** { t

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int bilangan, pangkat;
```

5. Kemudian mulailah pemrograman

```
import java.util.Scanner;
public class Percobaan2 {
    static int hitungPangkat(int x, int y){
        if (y==0){
            return (1);
        } else {
            return (x *hitungPangkat(x, y - 1));
        }
    }
}
Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int bilangan, pangkat;

    System.out.println(x:"Bilangan yang dihitung: ");
    bilangan = sc.nextInt();
    System.out.println(x:"Pangkat : ");
    pangkat =sc.nextInt();

    System.out.println(hitungPangkat(bilangan, pangkat));
}
```

6. Maka hasilnya akan seperti ini

```
Bilangan yang dihitung:  
5  
Pangkat :  
3  
125
```

Pertanyaan 2

1. Pada Percobaan2, terdapat pemanggilan fungsi rekursif hitungPangkat(bilangan, pangkat) pada fungsi main, kemudian dilakukan pemanggilan fungsi hitungPangkat() secara berulang kali. Jelaskan sampai kapan proses pemanggilan fungsi tersebut akan dijalankan!

Pemanggilan fungsi akan terus berlanjut hingga kondisi dasar terpenuhi, yaitu $y == 0$. Setiap kali fungsi tersebut dipanggil, nilai y dikurangi 1. Jadi, jika Anda memulai dengan y sebagai 3, fungsi tersebut akan dipanggil lagi dengan y sebagai 2, kemudian 1, dan akhirnya 0. Ketika y menjadi 0, fungsi tersebut akan mengembalikan 1 dan tidak memanggil dirinya sendiri lagi. Nilai ini kemudian dikalikan dengan x dari pemanggilan fungsi sebelumnya, dan proses ini berlanjut sampai semua pemanggilan fungsi selesai dan hasil akhir dikembalikan.

2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh :

hitungPangkat(2,5) dicetak $2 \times 2 \times 2 \times 2 \times 2 = 32$

Dibawah ini adalah kode program yang sudah saya tambahkan

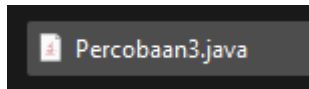
```
1  import java.util.Scanner;
2  public class Percobaan2After {
3      static int hitungPangkat(int x, int y){
4          if (y==0){
5              return (1);
6          } else {
7              return (x *hitungPangkat(x, y - 1));
8          }
9      }
10     public static void main(String[] args) {
11         Scanner sc = new Scanner(System.in);
12         int bilangan, pangkat;
13
14         System.out.println("Bilangan yang dihitung: ");
15         bilangan = sc.nextInt();
16         System.out.println("Pangkat : ");
17         pangkat =sc.nextInt();
18         System.out.println("Hasil Perhitungan : ");
19         for (int i=0; i<= pangkat; i++){
20             if (i==pangkat){
21                 System.out.print(bilangan);
22             } else {
23                 System.out.print(bilangan + "x");
24             }
25         }
26
27         System.out.println(" = " + hitungPangkat(bilangan, pangkat));
28     }
29 }
```

Dan ini adalah output dari kode program yang sudah saya tambahkan

```
Bilangan yang dihitung:
2
Pangkat :
5
Hasil Perhitungan :
2x2x2x2x2 = 32
```


Percobaan 3

1. Pertama buatlah file terlebih dahulu dengan format **Percobaan3.java**



2. Kemudian masukkan perintah **import java.util.Scanner;** untuk memulai pemrograman.

```
import java.util.Scanner;  
public class Percobaan3 {
```

3. Masukkan perintah **static int NamaFungsi ()** { terlebih dahulu sebelum memulai pemrograman.

```
static double hitungLaba(double saldo, int tahun){  
    if (tahun ==0){  
        return (saldo);  
    } else {  
        return (1.11 * hitungLaba(saldo, tahun -1));  
    }  
}
```

4. Lalu Masukkan perintah **public static void main(String args[])** { t

```
public static void main(String[] args) {  
    Scanner sc =new Scanner(System.in);  
    double saldoAwal;  
    int tahun;
```

5. Kemudian mulailah pemrograman

```
import java.util.Scanner;  
public class Percobaan3 {  
    static double hitungLaba(double saldo, int tahun){  
        if (tahun ==0){  
            return (saldo);  
        } else {  
            return (1.11 * hitungLaba(saldo, tahun -1));  
        }  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        Scanner sc =new Scanner(System.in);  
        double saldoAwal;  
        int tahun;  
  
        System.out.print(s:"Jumlah saldo awal : ");  
        saldoAwal = sc.nextInt();  
        System.out.print(s:"Lamanya invetasu (tahun) : ");  
        tahun = sc.nextInt();  
  
        System.out.print("Jumlah Saldo Setelah "+tahun+" ");  
        System.out.print(hitungLaba(saldoAwal, tahun));  
    }  
}
```

6. Maka hasilnya akan seperti ini

```
Jumlah saldo awal : 1000000
Lamanya invetasu (tahun) : 2
Jumlah Saldo Setelah 2 Tahun : 1232100.0
```

Pertanyaan 3

1. Pada Percobaan3, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!

Dibawah ini adalah blok kode dari BASE CASE

```
if (tahun ==0){  
    return (saldo);  
}
```

Ini adalah kondisi di mana fungsi berhenti melakukan pemanggilan rekursif dan langsung mengembalikan nilai (saldo). Dalam konteks ini, ketika tahun mencapai 0, fungsi mengembalikan nilai saldo saat itu sebagai hasil akhir.

Dibawah ini adalah blok kode dari Recursion Call

```
} else {  
    return (1.11 * hitungLaba(saldo, tahun -1));  
}
```

Pada setiap pemanggilan fungsi, nilai saldo dikalikan dengan 1.11 dan kemudian fungsi dipanggil lagi dengan nilai tahun yang dikurangi 1. Proses ini terus berlanjut hingga mencapai base case (tahun = 0), di mana pemanggilan rekursif dihentikan dan hasil akhirnya dikembalikan.

2. Jabarkan trace fase ekspansi dan fase substitusi algoritma perhitungan laba di atas jika diberikan nilai hitungLaba(100000,3)

Fase Ekspansi:

- A. Fungsi hitungLaba dipanggil dengan saldo awal 100,000 dan tahun 3.
- B. Fungsi memanggil dirinya sendiri dengan saldo 100,000 dan tahun 2.
- C. Fungsi kedua memanggil dirinya sendiri dengan saldo 100,000 dan tahun 1.
- D. Fungsi ketiga mengembalikan saldo karena tahun telah mencapai 0.

Fase Substitusi:

Panggilan pertama: Saldo = 100.000. Panggilan kedua: Saldo = $100.000 * 1,11 = 111.000$. Panggilan ketiga: Saldo = 111.000.

Hasil akhir: Saldo akhir = 111.000.

Tugas

1. Buatlah program untuk menampilkan bilangan n sampai 0 dengan menggunakan fungsi rekursif dan fungsi iteratif. (DeretDescendingRekursif).

Dibawah ini adalah Kode Program Rekursif

```
public class DeretDescendingRekursif {  
    public static void tampilDeretRekursif(int n) {  
        if (n >= 0) {  
            System.out.println(n);  
            tampilDeretRekursif(n - 1);  
        }  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 10;  
        tampilDeretRekursif(n);  
    }  
}
```

Dan ini adalah Output dari Kode Program Rekursif

```
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0
```

Dibawah ini adalah Kode Program Iteratif

```
public class DeretDescendingIteratif {  
    public static void tampilDeretIteratif(int n) {  
        for (int i = n; i >= 0; i--) {  
            System.out.println(i);  
        }  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 10;  
        tampilDeretIteratif(n);  
    }  
}
```

Dan ini adalah Output dari Kode Program Iteratif

```
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0
```

2. Buatlah program yang di dalamnya terdapat fungsi rekursif untuk menghitung penjumlahan bilangan. Misalnya $f = 8$, maka akan dihasilkan $1+2+3+4+5+6+7+8 = 36$ (PenjumlahanRekursif).

Dibawah ini adalah Kode Program PenjumlahanRekursif

```
public class PenjumlahanRekursif {  
    public static int jumlahRekursif(int n) {  
        if (n == 0) {  
            return 0;  
        } else {  
            return n + jumlahRekursif(n - 1);  
        }  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 8;  
        int hasil = jumlahRekursif(n);  
        System.out.println("Hasil penjumlahan = " + hasil);  
    }  
}
```

Dan ini adalah Output dari Kode Program PenjumlahanRekursif

```
Hasil penjumlahan = 36  
PS C:\Users\UPI-PC\Documents\Belajar\Dasar Pemrograman\Praktik\Semester-1\Jobsheet 14>
```

3. Buat program yang di dalamnya terdapat fungsi rekursif untuk mengecek apakah suatu bilangan n merupakan bilangan prima atau bukan. n dikatakan bukan bilangan prima jika ia habis dibagi dengan bilangan kurang dari n. (CekPrimaRekursif).

Dibawah ini adalah Kode Program CekPrimaRekursif

```
public class CekPrimaRekursif {  
    public static boolean cekPrimaRekursif(int n) {  
        if (n <= 1) {  
            return false;  
        }  
        for (int i = 2; i <= Math.sqrt(n); i++) {  
            if (n % i == 0) {  
                return false;  
            }  
        }  
        return true;  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 13;  
        boolean hasil = cekPrimaRekursif(n);  
        if (hasil) {  
            System.out.println(n + " adalah bilangan prima");  
        } else {  
            System.out.println(n + " bukan bilangan prima");  
        }  
    }  
}
```

Dan ini adalah Output dari Kode Program CekPrimaRekursif

```
13 adalah bilangan prima  
PS C:\Users\UPI-PC\Documents\Belajar\Dasar Pemrograman\Praktik\Semester-1\Jobsheet 14>
```

4. Sepasang marmut yang baru lahir (jantan dan betina) ditempatkan pada suatu pembiakan. Setelah dua bulan pasangan marmut tersebut melahirkan sepasang marmut kembar (jantan dan betina). Setiap pasangan marmut yang lahir juga akan melahirkan sepasang marmut juga setiap 2 bulan. Berapa pasangan marmut yang ada pada akhir bulan ke-12? Buatlah programnya menggunakan fungsi rekursif! (Fibonacci). Berikut ini adalah ilustrasinya dalam bentuk tabel.


Dibawah ini adalah Kode Program Fibonacci



```
public class Fibonacci17 {  
    // Fungsi rekursif untuk menghitung jumlah pasangan marmut  
    public static int hitungMarmut(int bulan) {  
        if (bulan <= 2) {  
            return 1; // Pada dua bulan pertama, hanya ada satu pasangan  
        } else {  
            return hitungMarmut(bulan - 1) + hitungMarmut(bulan - 2);  
        }  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        int bulan = 12;  
        int jumlahMarmut = hitungMarmut(bulan);  
        System.out.println("Jumlah pasangan marmut pada akhir bulan ke-" + bulan + " adalah: " + jumlahMarmut);  
    }  
}
```




Dan ini adalah Output dari Kode Program Fibonacci

```
Jumlah pasangan marmut pada akhir bulan ke-12 adalah: 144
```



GitHub Push :










 **Jobsheet-14** Public

 Pin  Unwatch 1

 main  1 branch  0 tags

[Go to file](#) [Add file](#) [Code](#)

 **Luthfi27Upi** Dasar Pemrograman 6036ad4 3 minutes ago  5 commits

 CekPrimaRekursif.java	Dasar Pemrograman	13 minutes ago
 DeretDescendingIteratif.java	Dasar Pemrograman	13 minutes ago
 DeretDescendingRekursif.java	Dasar Pemrograman	13 minutes ago
 Fibonacci17.java	Dasar Pemrograman	3 minutes ago
 PenjumlahanRekursif.java	Dasar Pemrograman	13 minutes ago
 Percobaan1.java	Dasar Pemrograman	1 hour ago
 Percobaan2.java	Dasar Pemrograman	1 hour ago
 Percobaan2After.java	Dasar Pemrograman	32 minutes ago
 Percobaan3.java	Dasar Pemrograman	1 hour ago