

LAPORAN PRAKTIKUM DASPRO JOBSHEET 12

Nama : M Nawfal Mawla A

Kelas / No. Absen: 1D-TI / 19

NIM : 244107020174

PERCOBAAN 1 :

1. Membuat Fungsi Static faktorialRekursif()

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

2. Membuat Fungsi Static faktorialIteratif()

```
static int faktorialIteratif(int n) {  
    int faktor = 1;  
    for (int i = n; i >= 1; i--){  
        faktor = faktor * i;  
    }  
    return faktor;  
}
```

3. Membuat fungsi main dan memanggil fungsi rekursif dan iteratif

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

PERTANYAAN 1 :

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri untuk menyelesaikan masalah yang dapat dipecah menjadi submasalah serupa.
2. Contoh kasus
3. Ya, hasil yang diberikan oleh fungsi faktorialRekursif() dan faktorialIteratif() akan sama jika diberi input yang sama. Kedua fungsi tersebut menghitung nilai faktorial dari suatu bilangan n , tetapi cara kerjanya berbeda. Perbedaan rekursif dengan iteratif yaitu:

Rekursif : Fungsi memanggil dirinya sendiri hingga mencapai *base case*. Menggunakan struktur *stack* untuk melacak setiap panggilan fungsi, dengan risiko *stack overflow* jika n besar.

Iteratif : Menggunakan perulangan langsung tanpa membutuhkan *stack*. Lebih efisien dalam penggunaan memori.

PERCOBAAN 2 :

1. Membuat Fungsi Hitung Pangkat

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

2. Membuat Fungsi Main

```
public static void main(String[] args) {  
    Scanner sc = new Scanner (System.in);  
  
    System.out.print(s:" Bilangan yang dihitung : ");  
    int bilangan = sc.nextInt();  
    System.out.print(s:"Pangkat: ");  
    int pangkat = sc.nextInt();  
}
```

3. Memanggil fungsi hitungPangkat untuk melakukan perhitungan di main

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

PERTANYAAN 2 :

1. Fungsi rekursif hitung pangkat akan terus berlanjut hingga memenuhi kondisi dasar yaitu $(y == 0)$.

```
public class Percobaan2 {  
    static String deret = "";  
  
    static int hitungPangkat(int x, int y) {  
        if (y == 0) {  
            deret += "1";  
            return 1;  
        } else {  
            if (deret.isEmpty()) {  
                deret += x; // Menambahkan bilangan pertama ke deret  
            } else {  
                deret += "x" + x; // Menambahkan pengali berikutnya  
            }  
            return x * hitungPangkat(x, y - 1);  
        }  
    }  
}
```

2.

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
  
    System.out.print(s:"Bilangan yang dihitung: ");  
    int bilangan = sc.nextInt();  
    System.out.print(s:"Pangkat: ");  
    int pangkat = sc.nextInt();  
  
    deret = " ";  
    int hasil = hitungPangkat(bilangan, pangkat);  
  
    System.out.println(deret + " = " + hasil);  
}
```

PERCOBAAN 3 :

1. Membuat fungsi hitungLaba

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

2. Membuat fungsi main dan memanggil fungsi hitungLaba

```
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 investasi (tahun) : ");  
    tahun = sc.nextInt();  
  
    System.out.print("Jumlah saldo setelah " + tahun + " Tahun adalah :");  
    System.out.print( hitunglaba(saldoAwal, tahun));  
}
```

PERTANYAAN 3 :

1. - Base Case (kondisi menghentikan rekursi)
- Recursion Call (bagian memanggil fungsi)

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

```
return (1.11 * hitunglaba(saldo, tahun - 1));
```

2. Pertama tama setelah input dari saldo awal dan tahun telah di masukkan maka akan terjadi pemanggilan fungsi yang berulang dengan mengurangi tahun dengan angka 1 hingga tahun menjadi 0, setelah itu menghitung laba dimulai dari tahun 1 hingga 3 dan mendapatkan hasilnya

