

## Jobsheet 12 (Fungsi 2)

Nama	: Abdul Muid
Kelas	: TI 1D
NIM	: 244107020006
No. Absen	: 01

### Percobaan 1

#### Langkah Langkah

1. Buat repository baru dengan nama daspro-jobsheet12 Buat project baru bernama Rekursif, dan buat file Java dengan nama Percobaan1.
2. Buat fungsi static dengan nama faktorialRekrusif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya.

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

3. Buat lagi fungsi static dengan nama faktorialIteratif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya.

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

4. Buatlah fungsi main dan lakukan pemanggilan terhadap kedua fungsi yang telah dibuat sebelumnya, dan tampilkan hasil yang didapatkan.

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

5. Jalankan program dan amati hasilnya.

```
120  
120
```

6. Push dan commit ke github.

#### Pertanyaan

1. Apa yang dimaksud dengan fungsi rekursif?

Jawab: Fungsi rekursif yaitu fungsi yang memiliki perintah untuk memanggil fungsi itu sendiri (dirinya sendiri).

2. Bagaimana contoh kasus penggunaan fungsi rekursif ?

Jawab: Contoh kasus penggunaan fungsi rekursif yaitu menghitung faktorial.

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!

Jawab: Sama, alur pada fungsi rekursif yaitu menggunakan pengulangan if else fungsi ini akan memanggil dirinya sampai kondisi base case terpenuhi yaitu  $n == 0$ , sedangkan alur fungsi iterative yaitu menggunakan pengulangan for artinya pengulangan akan berhenti jika kondisi false.

## Percobaan 2

### Langkah Langkah

1. Buat file Java dengan nama Percobaan2.
2. Buat fungsi static dengan nama hitungPangkat(), dengan tipe data kembalian fungsi int dan memiliki 2 parameter dengan tipe data int berupa bilangan yang akan dihitung pangkatnya dan bilangan pangkatnya.

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

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc., Kemudian buatlah dua buah variabel bertipe int dengan nama bilangan dan pangkat.

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

4. Tambahkan kode berikut ini untuk menerima input dari keyboard.

```
System.out.print(s:"Bilangan yang dihitung: ");  
bilangan = sc.nextInt();  
System.out.print(s:"Pangkat: ");  
pangkat = sc.nextInt();
```

5. Lakukan pemanggilan fungsi hitungPangkat yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

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

6. Jalankan program dan amati hasilnya.

```
Bilangan yang dihitung: 2  
Pangkat: 3  
8
```

7. Push dan commit ke github.

### Pertanyaan

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!

Jawab: Proses pemanggilan fungsi tersebut akan terus berjalan sampai  $y == 0$  karena merupakan base case.

2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : hitungPangkat(2,5) dicetak  $2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$

Jawab:

```
if (y == 0){
    System.out.print(s:"1");
    return (1);
} else {
    System.out.print(x + "x");
    return (x * hitungPangkat(x, y -1));
}
```

```
Bilangan yang dihitung: 2
Pangkat: 5
Perhitungan: 2x2x2x2x2x1 = 32
```

## Percobaan 3

### Langkah Langkah

1. Buat file baru dengan nama Percobaan3.
2. Buat fungsi static dengan nama hitungLaba(), dengan tipe data kembalian fungsi double dan memiliki 2 parameter dengan tipe data int berupa saldo investor dan lamanya investasi. Lakukan seperti berikut.

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

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc. Kemudian buatlah sebuah variabel bertipe double dengan nama saldoAwal dan sebuah variabel bertipe int bernama tahun.

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

4. Tambahkan kode berikut untuk menerima input dari keyboard.

```
System.out.print(s:"Jumlah saldo awal: ");
saldoAwal = sc.nextDouble();
System.out.print(s:"Lamanya investasi (tahun) : ");
tahun = sc.nextInt();
```

5. Lakukan pemanggilan fungsi hitungLaba yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

```
System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
System.out.println(hitungLaba(saldoAwal, tahun));
```

6. Jalankan program dan amati hasilnya.

```
Jumlah saldo awal: 200000
Lamanya investasi (tahun) : 2
Jumlah saldo setelah 2 tahun : 246420.000000000006
```

7. Push dan commit ke github.

### Pertanyaan

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

Jawab:

**“base case”**

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

**“recursion call”**

```
(1.11 * hitungLaba(saldo, tahun-1));
```

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

Jawab:

```
Jumlah saldo awal: 100000  
Lamanya investasi (tahun) : 3  
Jumlah saldo setelah 3 tahun : 136763.10000000003
```

**Fase ekspansi;**

= 1.11 \* hitungLaba(10000, 2)

= 1.11 \* hitungLaba(10000, 1)

= 1.11 \* hitungLaba(10000, 0)

= hasil = 100000

**Fase substitusi;**

= hasil = 100000

= 1.11 \* 100000 = 111000

= 1.11 \* 111000 = 123210

= 1.11 \* 123210 = 136763.100003

# Tugas

## Nomer 1

```
import java.util.Scanner;
public class Tugas1 {

    static void deretDescendingRekursif(int n) {
        if (n < 0) {
            return;
        } else {
            System.out.print(n + " ");
            deretDescendingRekursif(n - 1);
            return;
        }
    }

    static void deretDescendingIteratif(int n) {
        for (int i = n; i >= 0; i--) {
            System.out.print(i + " ");
        }
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Masukkan bilangan: ");
        int n = sc.nextInt();

        System.out.print(s:"Deret Descending (Rekursif):");
        deretDescendingRekursif(n);
        System.out.println();
        System.out.print(s:"Deret Descending (Iteratif):");
        deretDescendingIteratif(n);

        sc.close();
    }
}
```

```
Masukkan bilangan: 9
Deret Descending (Rekursif):9 8 7 6 5 4 3 2 1 0
Deret Descending (Iteratif):9 8 7 6 5 4 3 2 1 0
```

## Nomer 2

```
import java.util.Scanner;
public class Tugas2 {

    static int hitungPenjumlahan(int n) {
        if (n == 1) {
            System.out.print(s:"1");
            return 1;
        } else {
            int hasilSebelum = hitungPenjumlahan(n - 1);
            System.out.print(" + " + n);
            return n + hasilSebelum;
        }
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s:"Masukkan bilangan: ");
        int bilangan = sc.nextInt();

        int hasil = hitungPenjumlahan(bilangan);
        System.out.println(" = " + hasil);

        sc.close();
    }
}
```

```
Masukkan bilangan: 5
1 + 2 + 3 + 4 + 5 = 15
```

### Nomer 3

```
import java.util.Scanner;
public class Tugas3 {

    public static int fibonacci(int n) {
        if (n == 0 || n == 1) {
            return n;
        } else {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s:"Masukkan bulan (1-12): ");
        int bulan = sc.nextInt();
        int jumlahPasangan = fibonacci(bulan);

        System.out.println("Jumlah pasangan marmut pada bulan ke-" + bulan + " adalah " + jumlahPasangan);

        sc.close();
    }
}
```

```
Masukkan bulan (1-12): 12
Jumlah pasangan marmut pada bulan ke-12 adalah 144
```