

JOB SHEET 14

Fungsi Rekursif

1. Tujuan

- Mahasiswa memahami konsep fungsi rekursif
- Mahasiswa mampu mengimplementasikan fungsi rekursif dalam kode program

2. Praktikum

2.1 Percobaan 1

Waktu Percobaan: 60

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung nilai faktorial dari suatu bilangan dengan menggunakan fungsi **rekursif**. Selain itu, akan dibuat juga fungsi untuk menghitung nilai faktorial dengan menggunakan algoritma **iteratif** sebagai pembandingnya.

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

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

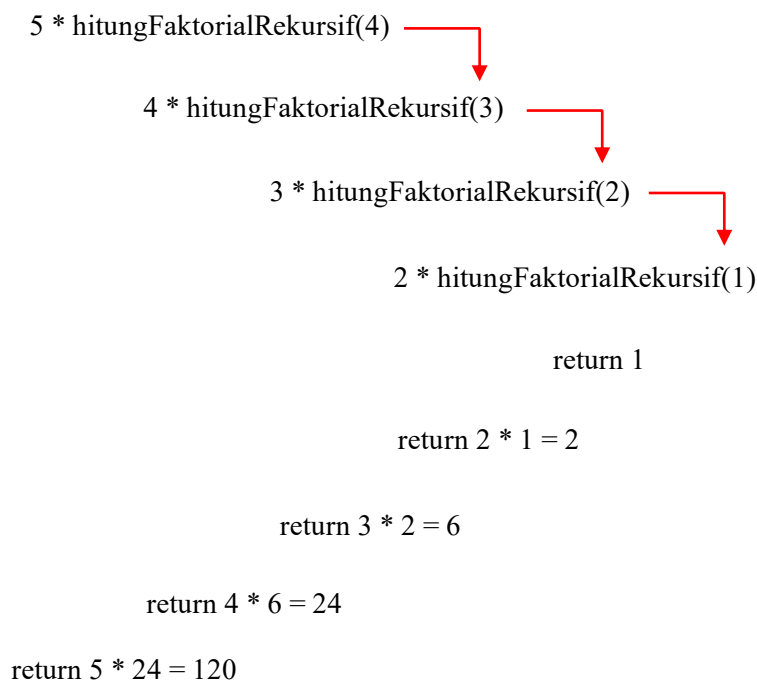
```
static int faktorialIteratif(int n) {  
    int faktor = 1;  
    for (int i = n; i >= 1; i--) {  
        faktor = faktor * i;  
    }  
    return faktor;  
}
```



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(faktorialRekursif(5));
    System.out.println(faktorialIteratif(5));
}
```

5. Jalankan program tersebut. Amati apa yang terjadi!
6. Jika ditelusuri, pada saat pemanggilan fungsi **faktorialRekursif(5)**, maka proses yang terjadi dapat diilustrasikan sebagai berikut:



Pertanyaan

1. Apa yang dimaksud dengan fungsi rekursif?
2. Pada **Percobaan1**, apakah hasil yang diberikan fungsi **faktorialRekursif()** dan fungsi **faktorialIteratif()** sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iteratif!

2.2 Percobaan 2

Waktu Percobaan : 60

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung pangkat sebuah bilangan dengan menggunakan fungsi rekursif.

1. Pada project **Rekursif**, dan 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**
4. Buatlah dua buah variabel bertipe int dengan nama **bilangan** dan **pangkat**
5. Tambahkan kode berikut ini untuk menerima input dari keyboard

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

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

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

7. Jalankan program tersebut. Amati apa yang terjadi!

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



2.3 Percobaan 3

Waktu Percobaan : 60

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung jumlah uang investor yang digunakan sebagai investasi setelah mendapatkan laba selama beberapa tahun dengan menggunakan fungsi rekursif.

1. Pada project **Rekursif**, dan buat file Java 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.

Pada kasus ini dianggap laba yang ditentukan adalah 11% per tahun. Karena perhitungan laba adalah $\text{laba} * \text{saldo}$, sehingga untuk menghitung besarnya uang setelah ditambah laba adalah $\text{saldo} + \text{laba} * \text{saldo}$. Dalam hal ini, besarnya laba adalah $0.11 * \text{saldo}$, dan saldo dianggap $1 * \text{saldo}$, sehingga $1 * \text{saldo} + 0.11 * \text{saldo}$ dapat diringkas menjadi **1.11 * saldo** untuk perhitungan saldo setelah ditambah laba (dalam setahun).

```
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**
4. Buatlah sebuah variabel bertipe double dengan nama **saldoAwal** dan sebuah variabel bertipe int bernama **tahun**
5. Tambahkan kode berikut ini untuk menerima input dari keyboard

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

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

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

7. Jalankan program tersebut. Amati apa yang terjadi!



Pertanyaan

1. Pada **Percobaan3**, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!
2. Jabarkan trace fase ekspansi dan fase substitusi algoritma perhitungan laba di atas jika diberikan nilai **hitungLaba(100000,3)**

3. Tugas

Waktu Pengerjaan 120 menit

Buatlah program yang mengimplementasikan **fungsi rekursif** dan **fungsi iteratif** untuk menghitung dan mencetak total nilai, dengan input $N \rightarrow 1 + 2 + 3 + 4 + 5 + \dots + \dots + N$. Contoh:

```
Masukkan jumlah angka yang ingin dihitung (N): 4
Masukkan angka ke-4: 100
Masukkan angka ke-3: 80
Masukkan angka ke-2: 75
Masukkan angka ke-1: 65
Total dari 4 angka yang dimasukkan adalah: 320
```

~ Selamat Mengerjakan ~