

JOOBSHEET 11/REKURSIF

Nama : Siti Mutmainah

Kelas : TI-1B

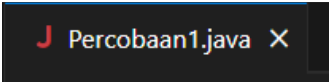
Absen : 21

NIM : 244107020143

Percobaan 1

Langkah-langkah Percobaan :

1. Buat Repository di GitHub, penamaan disesuaikan.
2. Lakukan cloning di terminal.
Dengan ketik **git.clone** (code repository), lalu **cd** (penaruhan file).
3. Buat file baru dengan nama file disesuaikan.
Dengan klik +



Percobaan1.java

4. Buat struktur class.

```
public class Percobaan1 {
```

5. 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);  
    }  
}
```

6. 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;  
}
```

7. 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(n:5));
        System.out.println(faktorialIteratif(n:5));
    }
}

```

- Run program tersebut.

```

PS D:\DASPRO\Rekursif> cd "d:\DASPRO\Rekursif"
PS D:\DASPRO\Rekursif> java Percobaan1
120
120

```

- Push dan commit ke GitHub.

Pertanyaan 1

- Apa yang dimaksud dengan fungsi rekursif?
Jawab: Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri untuk menyelesaikan masalah secara bertahap dengan membagi menjadi bagian yang lebih kecil.
- Bagaimana contoh kasus penggunaan fungsi rekursif ?
Jawab: Menghitung faktorial, dan menghitung pangkat.
- 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 fungsiRekursif

Fungsi memanggil dirinya sendiri hingga mencapai base case (nilai $n=0$). Setiap langkah memecah masalah menjadi lebih kecil dan lebih sederhana. Setelah mencapai base case, hasil dikembalikan secara bertahap ke panggilan fungsi sebelumnya.

$f(5)=5 \times f(4)$
 $f(4)=4 \times f(3)$
 $f(3)=3 \times f(2)$
 $f(2)=2 \times f(1)$
 $f(1)=1 \times f(0)$
 $f(0)=1$ (base case)

Alur fungsiIteratif

Menggunakan struktur pengulangan. Nilai dihitung secara langsung dengan cara mengalikan angka dari n hingga 1 dalam satu langkah berulang.

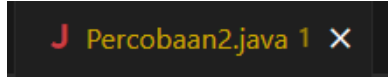
Iterasi factor = $1 \times 5 = 5$
 Iterasi factor = $5 \times 4 = 20$
 Iterasi factor = $20 \times 3 = 60$
 Iterasi factor = $60 \times 2 = 120$
 Iterasi factor = $120 \times 1 = 120$

Percobaan 2

Langkah-langkah Percobaan :

1. Buat file baru dengan nama file disesuaikan.

Dengan klik +



2. Buat struktur class.

```
public class Percobaan2 {
```

3. 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);  
    }  
}
```

4. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc.

```
import java.util.Scanner;  
  
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);
```

5. Buatlah dua buah variabel bertipe int dengan nama bilangan dan pangkat. Tambahkan kode berikut ini untuk menerima input dari keyboard.

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

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

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

7. Run program.

```
PS D:\DASPRO\Rekursif> cd "d:\DASPRO\R  
Bilangan yang dihitung:  
2  
Pangkat:  
2  
4
```

8. Push dan commit ke GitHub.

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!
Jawab: Pemanggilan fungsi akan terus dijalankan hingga nilai pangkat mencapai 0, merupakan base case.
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : hitungPangkat(2,5) dicetak 2x2x2x2x2x1 = 32

Jawab:

```

    System.out.print(s:"Hitung pangkat: ");
    for (int i = 0; i < pangkat; i++) {
        System.out.print(bilangan + "x");
    }
    System.out.print(s:"1");

    int hasil = hitungPangkat(bilangan, pangkat);
    System.out.println(" = " + hasil);
}

```

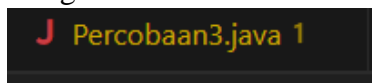
PS D:\DASPRO\Rekursif> cd "d:\DASPRO\Rek
Bilangan yang dihitung:
2
Pangkat:
5
Hitung pangkat: 2x2x2x2x2x1 = 32

Percobaan 3

Langkah-langkah Percobaan :

1. Buat file baru dengan nama file disesuaikan.

Dengan klik +



2. Buat struktur class.

```
public class Percobaan3 {
```

3. 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 laba * saldo, sehingga untuk menghitung besarnya uang setelah ditambah laba adalah saldo + laba * saldo. Dalam hal ini, besarnya laba adalah 0.11 * saldo, dan saldo dianggap 1 * saldo, sehingga 1 * saldo + 0.11 * 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);
    }
}
```

4. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc.

```
import java.util.Scanner;
```

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

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

```
System.out.print(s:"Jumlah saldo awal : ");
double saldoAwal = sc.nextDouble();
System.out.print(s:"Lamanya investasi (tahun) : ");
int 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(hitungLaba(saldoAwal, tahun));
}
```

7. Run program.

```
PS D:\DASPRO\Rekursif> cd "d:\DASPRO\Rekursif\" ; if ($?) {
Jumlah saldo awal : 1000
Lamanya investasi (tahun) : 10
Jumlah saldo setelah 10 tahun : 2839.420986069018
}
```

8. Push dan commit ke GitHub.

Pertanyaan 3

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

```
} else {
    return 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:

Fase ekspansi

fungsi `hitungLaba` memanggil dirinya sendiri berulang kali dengan parameter yang semakin kecil, hingga mencapai base case `tahun = 0`.

- $\text{hitungLaba}(100000,3) = 1.11 \times \text{hitungLaba}(100000,2)$
- $\text{hitungLaba}(100000,2) = 1.11 \times \text{hitungLaba}(100000,1)$
- $\text{hitungLaba}(100000,1) = 1.11 \times \text{hitungLaba}(100000,0)$
- $\text{hitungLaba}(100000,0) = 100000$

Fase substitusi

Setelah mencapai base case, fungsi mulai menghitung hasilnya secara terbalik, menggantikan setiap pemanggilan rekursif dengan hasil yang dihitung.

- $\text{hitungLaba}(100000,0) = 1.11 \times 100000 = 111000$
- $\text{hitungLaba}(100000,1) = 1.11 \times 111000 = 123210$
- $\text{hitungLaba}(100000,2) = 1.11 \times 123210 = 136763.1$
- $\text{hitungLaba}(100000,3) = 1.11 \times 136763.1 = 151319.921$