

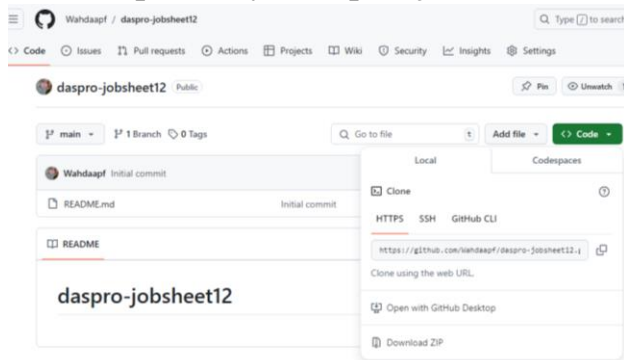
Nama: Wahda Adella Putri Febriana

NIM : 244107020156

Kelas: 1B

## Percobaan 1

### 1. Buat Repository “daspro-jobsheet12”



### 2. Clone dan Open di VSC, buat file Percobaan1.java, dan fungsi faktorialRekursif(), faktorialIteratif()

```
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;  
    }  
}
```

### 3. Panggil function di main

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

### 4. Hasil run

```
ceptionMessages -cp C:\Users\Administrator\AppData\Local\Temp\ea2c\redhat.java\jdt_ws\daspro-jobsheet12_f8121e2  
120  
120  
PS D:\Project\Belajar\Java\daspro-jobsheet12>
```

## Pertanyaan 1

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri selama eksekusi untuk menyelesaikan suatu masalah dengan cara membaginya menjadi submasalah yang lebih kecil dan serupa.
2. a. menghitung nilai faktorial  
b. Fibonacci
3. Sama, fungsi faktorialRekursif memanggil dirinya sendiri secara terus-menerus hingga mencapai  $n == 0$ , fungsi berhenti memanggil dirinya sendiri dan mulai "kembali" (return) hasil perhitungan secara berurutan ke pemanggil sebelumnya, fungsi faktorialIteratif menggunakan perulangan for untuk menghitung faktorial. Variabel faktor digunakan untuk menyimpan hasil perhitungan sementara, dan loop mengalikan nilai faktor dengan angka yang menurun dari  $n$  hingga 1.

## Percobaan 2

1. Buat file Percobaan2.java, dan fungsi hitungPangkat

```
public class Percobaan2 {  
    static int hitungPangkat(int x, int y) {  
        if(y == 0) {  
            return (1);  
        } else {  
            return (x * hitungPangkat(x, y - 1));  
        }  
    }  
}
```

2. Deklarasi scanner

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

3. Tambahkan kodingan di main

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

## Pertanyaan 2

1. Pada program Percobaan2, fungsi hitungPangkat(x, y) dipanggil secara rekursif hingga mencapai kondisi base case, yaitu saat  $y == 0$ . Pada titik ini, fungsi berhenti memanggil dirinya sendiri dan mulai mengembalikan nilai hasil perhitungan.
2. Kode

```
static String deret = "";

static int hitungPangkat(int x, int y) {
    if (y == 0) {
        deret += "1";
        return 1;
    } else {
        deret += x + "x";
        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.print(s:"Bilangan yang dihitung: ");
    bilangan = sc.nextInt();
    System.out.print(s:"Pangkat: ");
    pangkat = sc.nextInt();

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

    sc.close();
}
```

## Percobaan 3

1. Buat file Percobaan3.java dan fungsi hitungLaba

```
Percobaan3.java > ...
public class Percobaan3 {
    static double hitungLaba(double saldo, int tahun) {
        if (tahun == 0) {
            return (saldo);
        } else {
            return (1.11 * hitungLaba(saldo, tahun - 1));
        }
    }
}
```

2. Tambah kodingan ini di main

```

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.nextDouble();
    System.out.print(s:"Lamanya investasi (tahun) : ");
    tahun = sc.nextInt();

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

    sc.close();
}

```

## Pertanyaan

1. Base case nya
 

```

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

 Recursion
 

```

      return (1.11 * hitungLaba(saldo, tahun - 1));
      
```
2. Fase Ekspansi (Memecah Masalah)
  - a.  $1.11 \times \text{hitungLaba}(100000, 3) = 136761$
  - b.  $1.11 \times \text{hitungLaba}(100000, 2) = 123210$
  - c.  $1.11 \times \text{hitungLaba}(100000, 1) = 111000$
  - d.  $\text{hitungLaba}(100000, 0) = 100000$

## Percobaan 4

1. Buat file pengunjungKafeNo24.java, dan tambahkan kodingan berikut

```

public class PengunjungCafe24 {

    public static void daftarPengunjung (String... namaPengunjung) {
        System.out.println(x:"Daftar Nama Pengunjung: ");
        for (int i = 0; i < namaPengunjung.length; i++) {
            System.out.println("-"+namaPengunjung[i]);
        }
    }

    Run | Debug
    public static void main(String[] args) {
        daftarPengunjung (...namaPengunjung:"Ali", "Budi", "Citra");
    }
}

```

## Pertanyaan

1. bertujuan untuk memungkinkan metode menerima **jumlah argumen yang tidak ditentukan sebelumnya**.
2. Ubah menjadi for each

```
public static void daftarPengunjung (String... namaPengunjung) {  
    System.out.println(x:"Daftar Nama Pengunjung: ");  
    for (String nama : namaPengunjung) { // Menggunakan for-each loop  
        System.out.println("-" + nama);  
    }  
}
```

3. Tidak, tidak dapat menggunakan dua parameter varargs (...) dalam satu fungsi. Karena parameter varargs akan menangkap semua argumen yang sesuai, sehingga akan membingungkan compiler mengenai argumen mana yang harus masuk ke parameter varargs tertentu.

## Tugas

1. Output

```
Deret menggunakan Rekursif:  
5 4 3 2 1 0  
  
Deret menggunakan Iteratif:  
5 4 3 2 1 0  
PS D:\Project\Belajar\Java\daspro-jobsheet12>
```

## Code

```
Tugas1.java > Tugas1 > deretDescendingIteratif(int)  
import java.util.Scanner;  
  
public class Tugas1 {  
    // Fungsi Rekursif  
    static void deretDescendingRekursif(int n) {  
        if (n < 0) {  
            return; // Base case: jika n < 0, hentikan rekursi  
        }  
        System.out.print(n + " ");  
        deretDescendingRekursif(n - 1); // Rekursif dengan n - 1  
    }  
  
    // Fungsi Iteratif  
    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 nilai n: ");  
    int n = sc.nextInt();  
  
    System.out.println(x:"\nDeret menggunakan Rekursif:");  
    deretDescendingRekursif(n);  
  
    System.out.println(x:"\n\nDeret menggunakan Iteratif:");  
    deretDescendingIteratif(n);  
  
    sc.close();  
}
```

## 2. code

```
public class Tugas2 {
    static int hitungPenjumlahan(int f) {
        if (f == 0) {
            return 0;
        } else {
            return f + hitungPenjumlahan(f - 1);
        }
    }

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

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

        System.out.println("Penjumlahan dari 1 hingga " + f + " = " + hitungPenjumlahan(f));

        sc.close();
    }
}
```

## Ouput

```
edhat.java\jdt_ws\daspro-jobsheet12_f8121e27\bin
Masukkan nilai f: 5
Penjumlahan dari 1 hingga 5 = 15
PS D:\Project\Belajar\Java\daspro-jobsheet12> |
```

## 3. Output

```
edhat.java\jdt_ws\daspro-jobsheet12_f8121e27\bin - Tugas2
Jumlah pasangan marmut pada akhir bulan ke-12 adalah: 144
Bulan    Jumlah Pasangan    Produktif    Belum Produktif
1         1                0            1
2         1                0            1
3         2                1            1
4         3                1            2
5         5                2            3
6         8                3            5
7        13                5            8
12       144            55           89
PS D:\Project\Belajar\Java\daspro-jobsheet12> |
```

## Kode

```
public static int hitungPasanganMarmut(int bulan) {
    if (bulan == 1 || bulan == 2) {
        return 1;
    }
    return hitungPasanganMarmut(bulan - 1) + hitungPasanganMarmut(bulan - 2);
}

Run | Debug
public static void main(String[] args) {
    int bulan = 12;
    System.out.println("Jumlah pasangan marmut pada akhir bulan ke-" + bulan + " adalah: " +
        hitungPasanganMarmut(bulan));
    System.out.printf(format:"%-10s%-20s%-15s%-15s\n", ...args:"Bulan", "Jumlah Pasangan", "Produktif",
        "Belum Produktif");
    int produktif = 0, belumProduktif = 1;

    for (int i = 1; i <= bulan; i++) {
        int totalPasangan = hitungPasanganMarmut(i);

        if (i > 2) {
            produktif = hitungPasanganMarmut(i - 2);
        }
        belumProduktif = totalPasangan - produktif;

        System.out.printf(format:"%-10d%-20d%-15d%-15d\n", i, totalPasangan, produktif, belumProduktif);
    }
}
```