

Nama: Rachmad Aprisandhy
kelas: TI-1B
noAbs : 18

JOBSHEET12

Percobaan1:

1. Buat file baru bernama percobaan1
2. Buat fungsi static dengan nama **faktorialrekursif()**

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

3. Buat fungsi static lagi dengan nama **faktorialIteratif()**

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

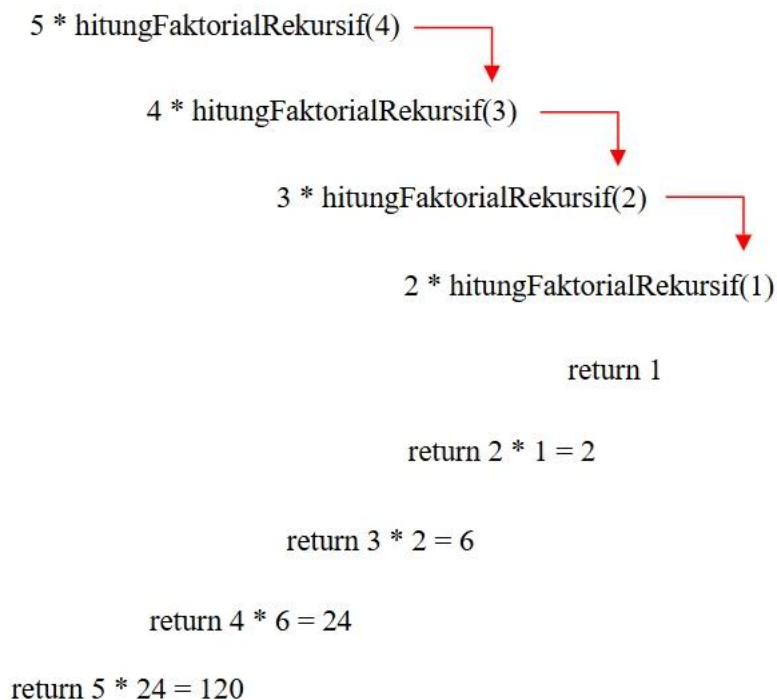
4. Buat fungsi main dan lakukan pemanggilan kembali ke 2 fungsi diatas

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

5. Amati hasil dari kode diatas

```
PS D:\Daspro praktik\Daspro-Jobsheet12\Daspro-Jobsheet12>  
ing\Code\User\workspaceStorage\d0f4ca05af216c7fd723793866c  
120  
120  
PS D:\Daspro praktik\Daspro-Jobsheet12\Daspro-Jobsheet12>  
odeDetailsInExceptionMessages' '-cp' 'C:\Users\ASUS\AppData  
Percobaan1 18'
```

6. Pada pemanggilan `FaktorialRekursif(5)`, maka proses yang terjadi dapat diilustrasikan menjadi seperti ini



Pertanyaan:

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri sebagai bagian dari proses eksekusi. Fungsi ini berguna untuk menyelesaikan masalah dengan cara memecah menjadi sub-masalah yang lebih kecil dengan pola yang serupa.
2. Menghitung angka dalam bentuk faktorial, maupun Mencari Angka Fibonacci
3. Iya sama
Fungsi FaktorialRekursif memanggil dirinya sendiri hingga mencapai *base case*. fungsi akan terus memanggil nilai N hingga nilai $N == 0$ dan jika sudah akan mengembalikan nilai 1 ke base case

sementara Fungsi Iteratif menggunakan perulangan `for` untuk menghitung faktorial. nilai N akan dihitung dalam perulangan lalu fungsi mengembalikan nilai akhir

Percobaan2:

1. Buat file baru bernama percobaan 2
2. Ketikkan library java

```
import java.util.Scanner;
```
3. Buat fungsi static dengan nama `hitung pangkat()`

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

4. Buat fungsi main dan deklarasikan Scanner

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

5. Buat 2 variabel dengan nama bilangan dan pangkat yang bertipe int

```
int bilangan, pangkat;
```

6. Lalu buat coding untuk menerima inputan dari keyboard

```
System.out.print(s:"Bilangan Yang Dihitung : ");  
bilangan = sc.nextInt();  
System.out.print(s:"pangkat : ");  
pangkat = sc.nextInt();
```

7. Lalu panggil fungsi hitung pangkat

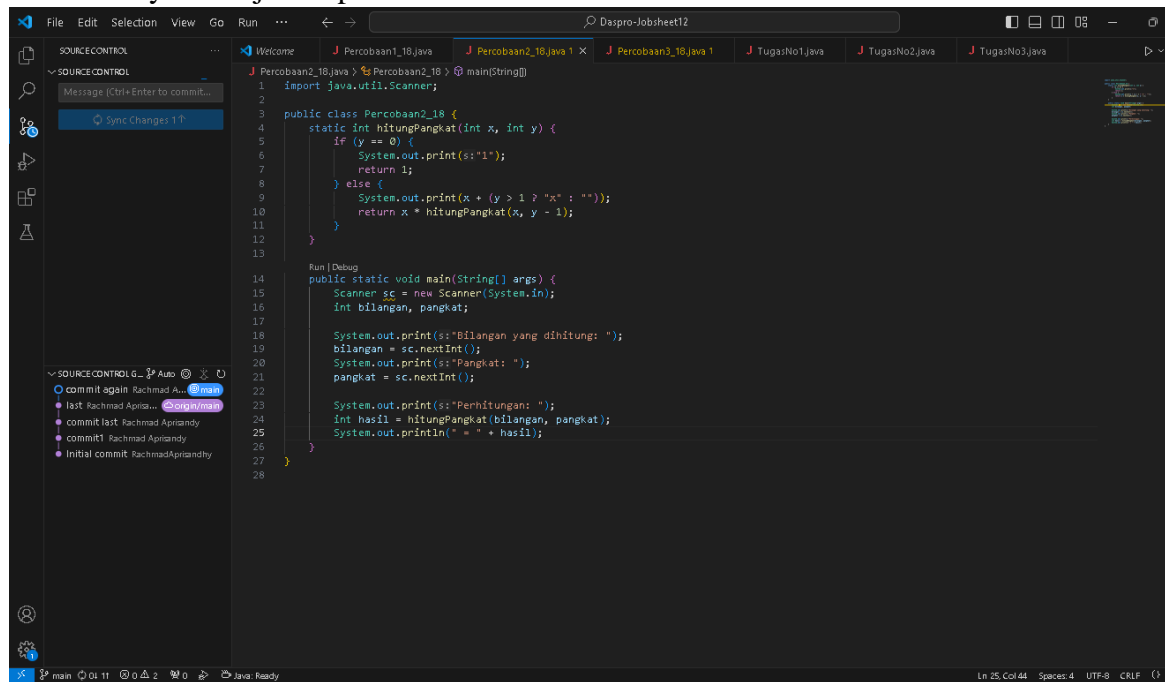
```
System.out.println(hitungPangkat(bilangan, pangkat));
```

8. Lalu amati hasilnya

```
public static void menu(String NamaPelanggan, boolean isMember){  
    System.out.println("selamat datang pelanggan : " + NamaPelanggan + "!");  
    if (isMember){  
        System.out.println(x:"Anda adalah member, anda mendapat diskon 10% di setiap pembelian");  
    }  
}
```

Pertanyaan:

1. Fungsi ini menggunakan konsep rekursif, yaitu memanggil dirinya sendiri dengan nilai parameter N yang berubah setiap kali diinputkan lalu akan mengembalikan nilai 1 ke base case jika parameter berpangkat == 0
2. Hasil editnya akan jadi seperti ini



Percobaan3:

1. Buat file baru dengan nama percobaan 3
2. Ketikkan library java

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

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

- ```
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));
```

- ```
PS D:\Daspro praktik\Daspro-Jobsheet12\Daspro-Jobsheet12> & 'C:\Program Files (Java)\
US\AppData\Roaming\Code\User\workspaceStorage\d0f4ca05af216c7fd723793866cc6c86\redhat
Jumlah Saldo Awal : 5000000
Lamanya Inevstasi (Tahun) : 4
Jumlah saldo setelah 4Tahun : 7590352.0500000004
PS D:\Daspro praktik\Daspro-Jobsheet12\Daspro-Jobsheet12>
```

1. base casenya adalah tahun = , refusioncalnya adalah return (1.11 * hitungLaba(saldo, tahun - 1));
2. Fase ekspansi adalah tahap di mana fungsi rekursif memanggil dirinya sendiri hingga mencapai *base case*. Berikut langkah-langkahnya:
HitungLaba(100000, 3):
return 1.11 * HitungLaba(100000, 2)
HitungLaba(100000, 2):
return 1.11 * HitungLaba(100000, 1)
HitungLaba(100000, 1):
return 1.11 * HitungLaba(100000, 0)
HitungLaba(100000, 0):
Base case tercapai, langsung mengembalikan saldo awal: 100000.

Fase substitusi adalah tahap di mana nilai dari panggilan rekursif dikembalikan, dan perhitungan dilakukan secara berurutan dari *base case* ke atas.

Dari HitungLaba(100000, 0):

$\text{HitungLaba}(100000, 0) = 100000$

Substitusi ke $\text{HitungLaba}(100000, 1)$:

$\text{HitungLaba}(100000, 1) = 1.11 * 100000 = 111000$

Substitusi ke $\text{HitungLaba}(100000, 2)$:

$\text{HitungLaba}(100000, 2) = 1.11 * 111000 = 123210$

Substitusi ke $\text{HitungLaba}(100000, 3)$:

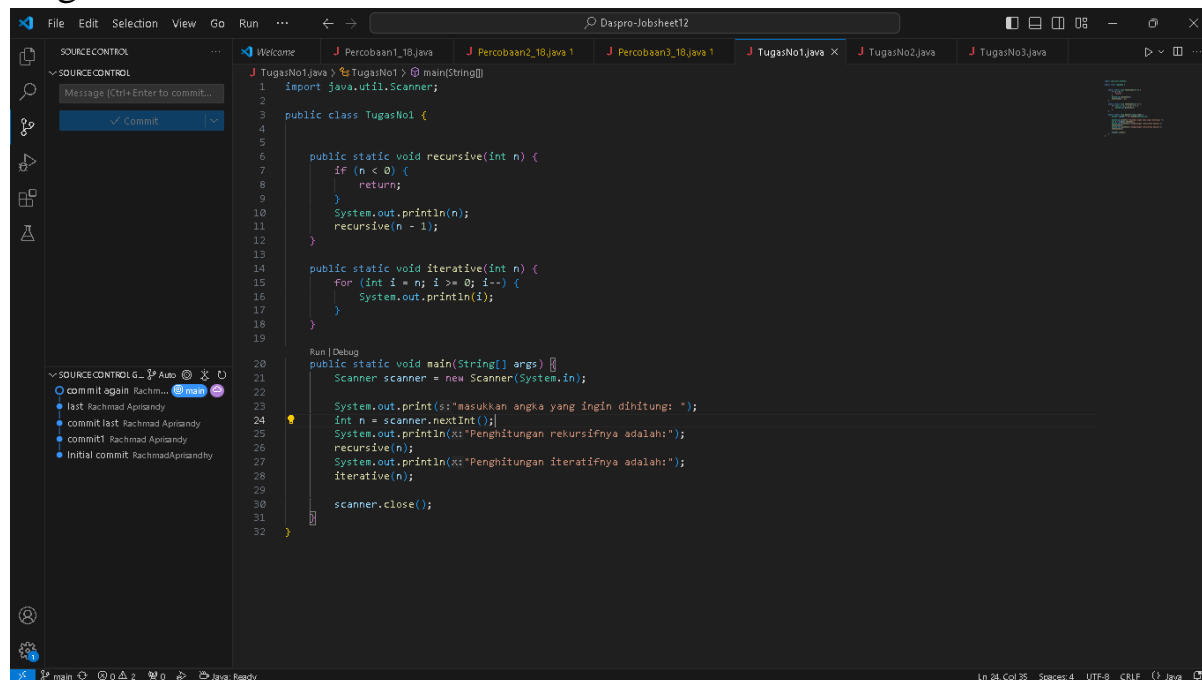
$\text{HitungLaba}(100000, 3) = 1.11 * 123210 = 136764.1$

Setelah fase substitusi selesai, hasil akhirnya adalah:

$\text{HitungLaba}(100000, 3) = 136764.1$

Tugas:

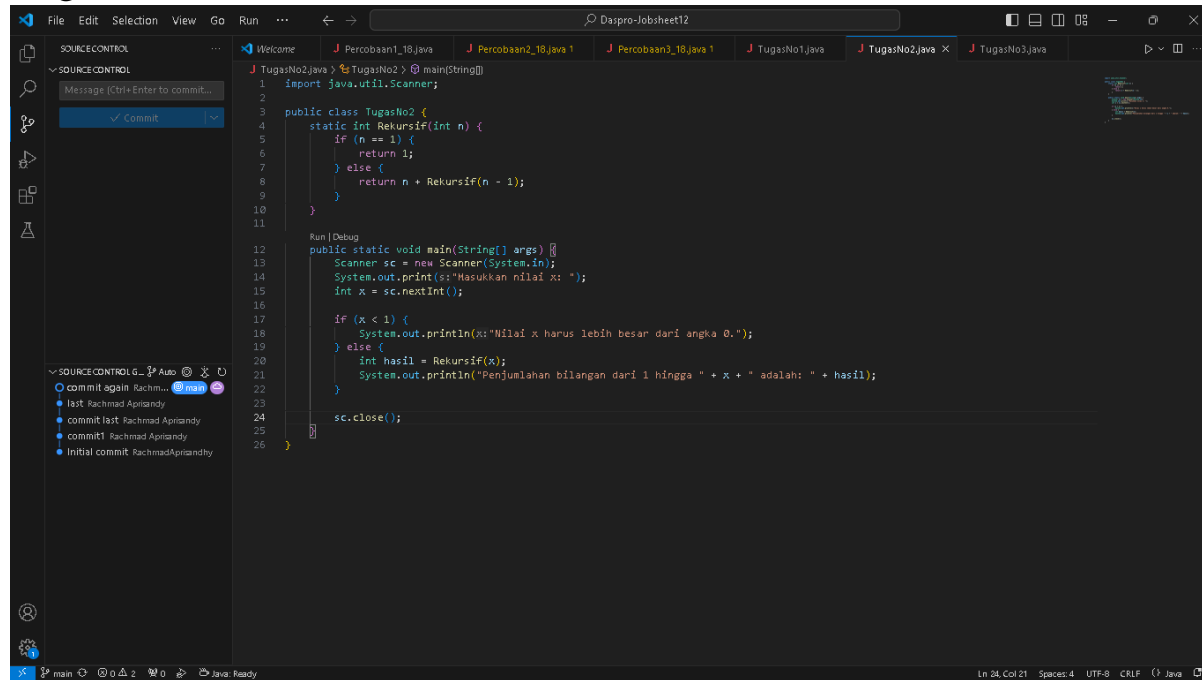
1. Tugas 1



The screenshot shows an IDE with a Java file named `TugasNo1.java`. The code implements two methods for calculating the value of HitungLaba at different interest rates. The `recursive` method uses a recursive approach, and the `iterative` method uses a loop. The `main` method prompts the user to input a number and then prints the results of both calculations.

```
1 import java.util.Scanner;
2
3 public class TugasNo1 {
4
5     public static void recursive(int n) {
6         if (n < 0) {
7             return;
8         }
9         System.out.println(n);
10        recursive(n - 1);
11    }
12
13    public static void iterative(int n) {
14        for (int i = n; i >= 0; i--) {
15            System.out.println(i);
16        }
17    }
18
19    public static void main(String[] args) {
20        Scanner scanner = new Scanner(System.in);
21
22        System.out.print("masukkan angka yang ingin dihitung: ");
23        int n = scanner.nextInt();
24        System.out.println("Penghitungan rekursifnya adalah:");
25        recursive(n);
26        System.out.println("Penghitungan iteratifnya adalah:");
27        iterative(n);
28
29        scanner.close();
30    }
31
32 }
```

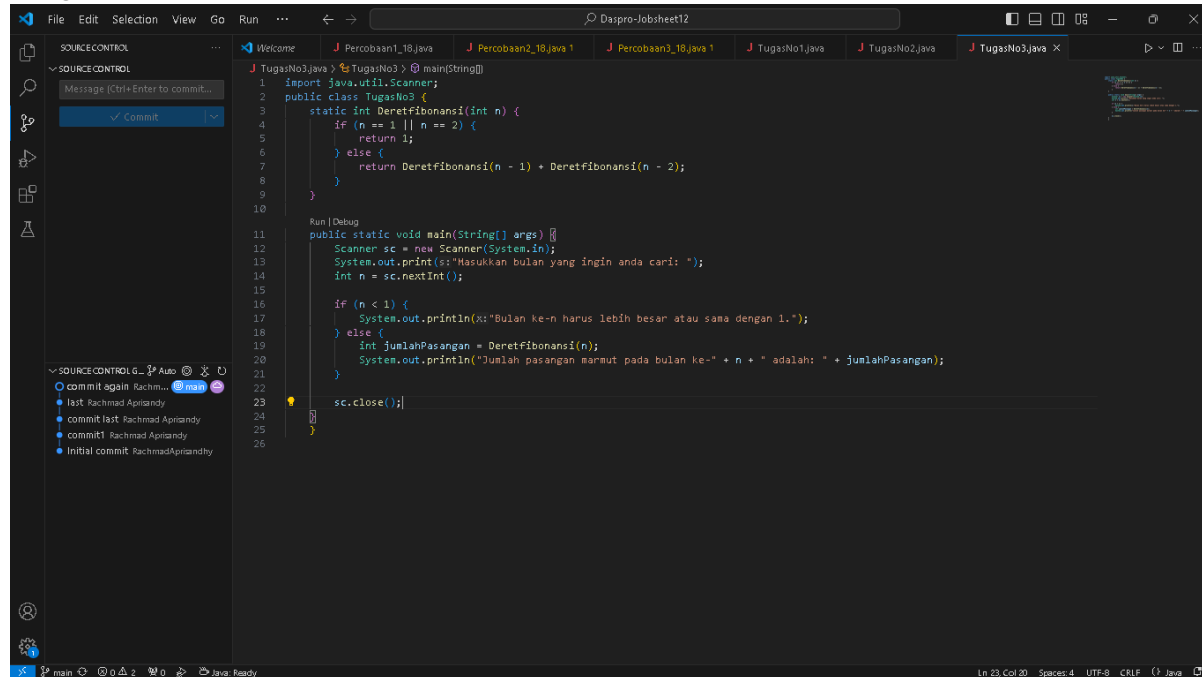
2. Tugas 2



The screenshot shows an IDE with a Java file named `TugasNo2.java`. The code implements a recursive function `rekursif` to calculate the sum of numbers from 1 to `n`. The `main` method uses a `Scanner` to take input from the user and prints the result.

```
1 import java.util.Scanner;
2
3 public class TugasNo2 {
4     static int rekursif(int n) {
5         if (n == 1) {
6             return 1;
7         } else {
8             return n + rekursif(n - 1);
9         }
10    }
11
12    public static void main(String[] args) {
13        Scanner sc = new Scanner(System.in);
14        System.out.print("Masukkan nilai x: ");
15        int x = sc.nextInt();
16
17        if (x < 1) {
18            System.out.println("Nilai x harus lebih besar dari angka 0.");
19        } else {
20            int hasil = rekursif(x);
21            System.out.println("Penjumlahan bilangan dari 1 hingga " + x + " adalah: " + hasil);
22        }
23
24        sc.close();
25    }
26 }
```

3. Tugas3



The screenshot shows an IDE with a Java file named `TugasNo3.java`. The code implements a recursive function `DeretFibonansi` to calculate the `n`th Fibonacci number. The `main` method uses a `Scanner` to take input from the user and prints the result.

```
1 import java.util.Scanner;
2
3 public class TugasNo3 {
4     static int DeretFibonansi(int n) {
5         if (n == 1 || n == 2) {
6             return 1;
7         } else {
8             return DeretFibonansi(n - 1) + DeretFibonansi(n - 2);
9         }
10    }
11
12    public static void main(String[] args) {
13        Scanner sc = new Scanner(System.in);
14        System.out.print("Masukkan bulan yang ingin anda cari: ");
15        int n = sc.nextInt();
16
17        if (n < 1) {
18            System.out.println("Bulan ke-n harus lebih besar atau sama dengan 1.");
19        } else {
20            int jumlahPasangan = DeretFibonansi(n);
21            System.out.println("Jumlah pasangan marmut pada bulan ke-" + n + " adalah: " + jumlahPasangan);
22        }
23
24        sc.close();
25    }
26 }
```