

Nama : Muhammad Zakaria Haniya

NIM : 244107020135

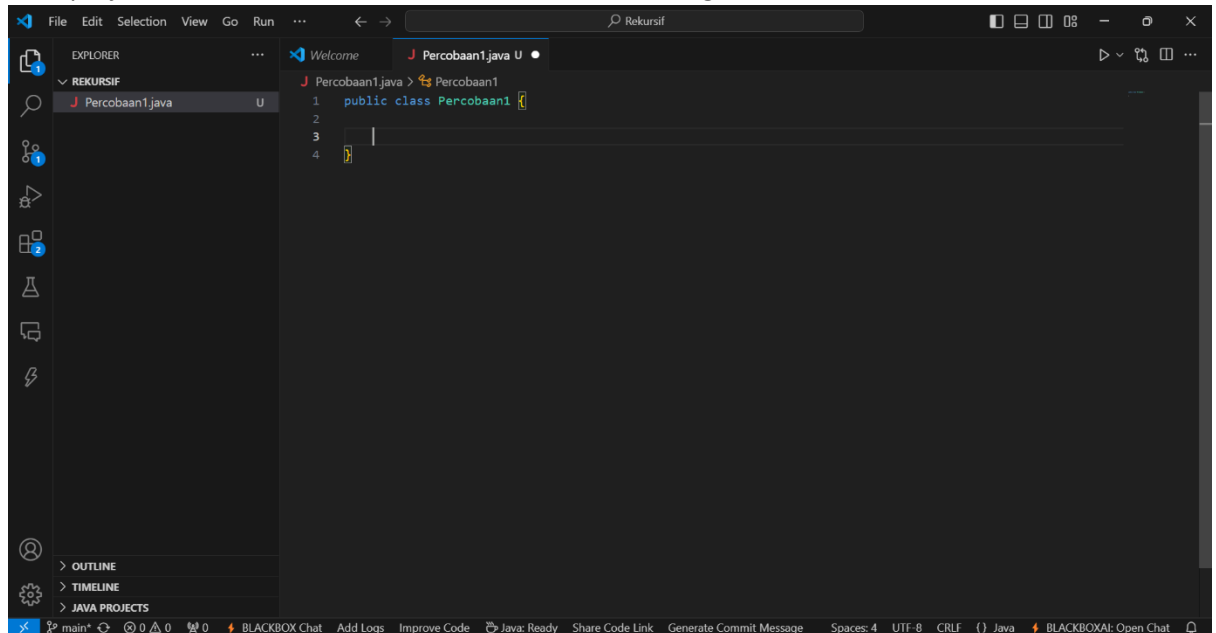
Prodi : D-IV Teknik Informatika

Kelas : 1B

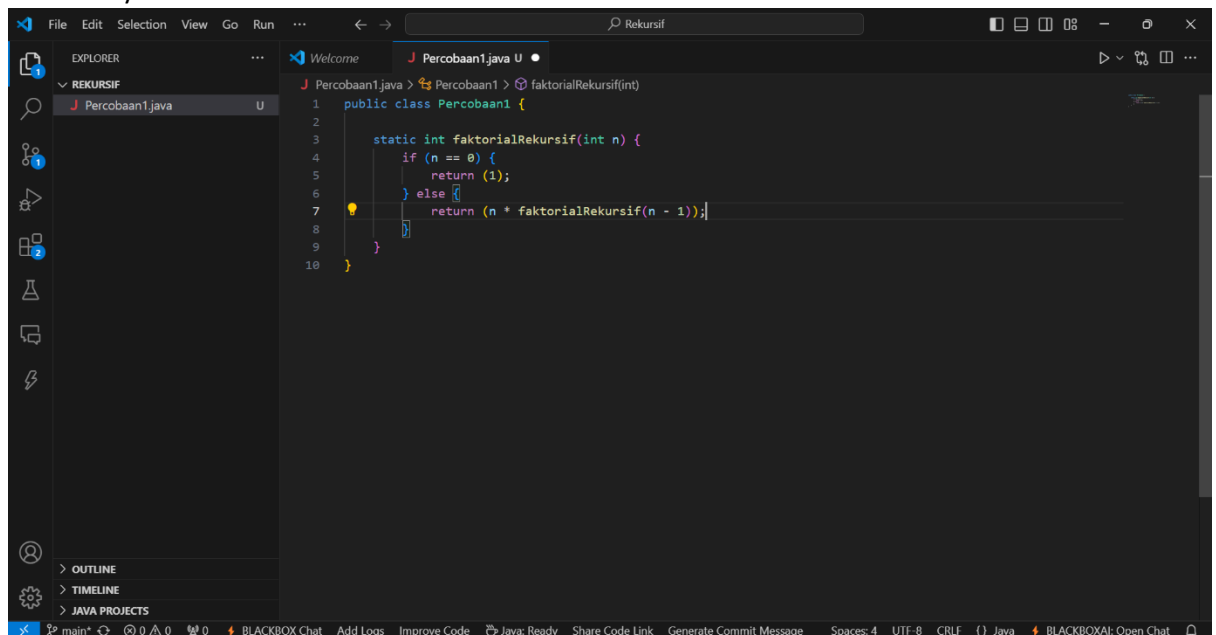
Absen : 16

Percobaan 1

1. Buat project baru bernama Rekursif, dan buat file Java dengan nama Percobaan1

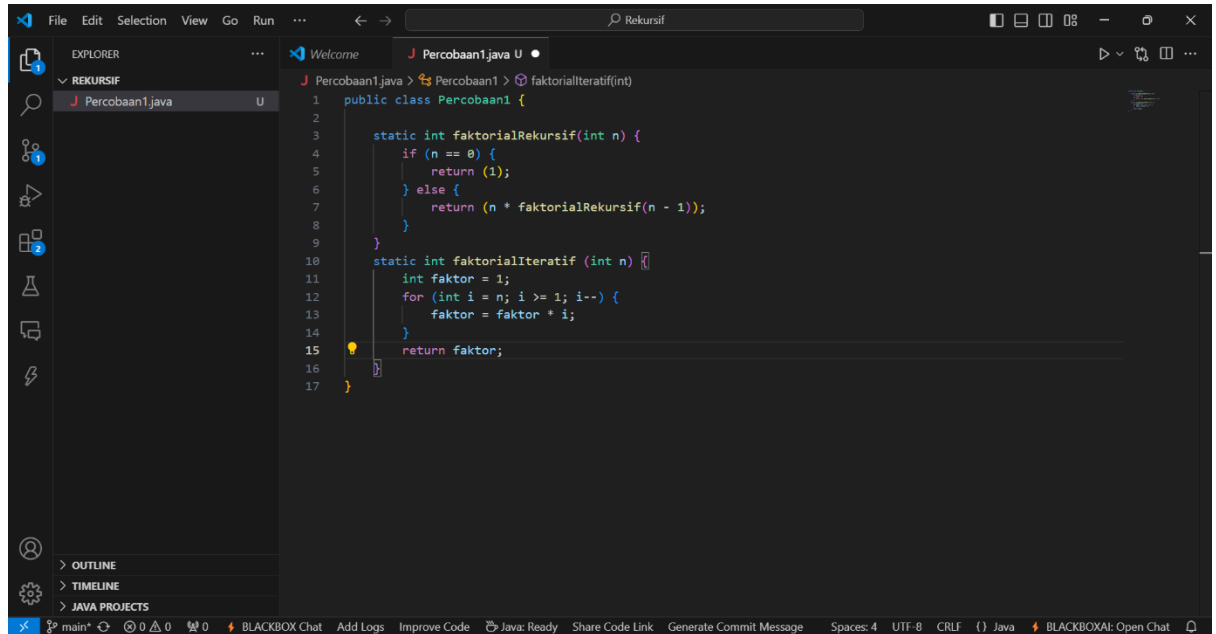


2. Buat fungsi static dengan nama faktorialRekursif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya.



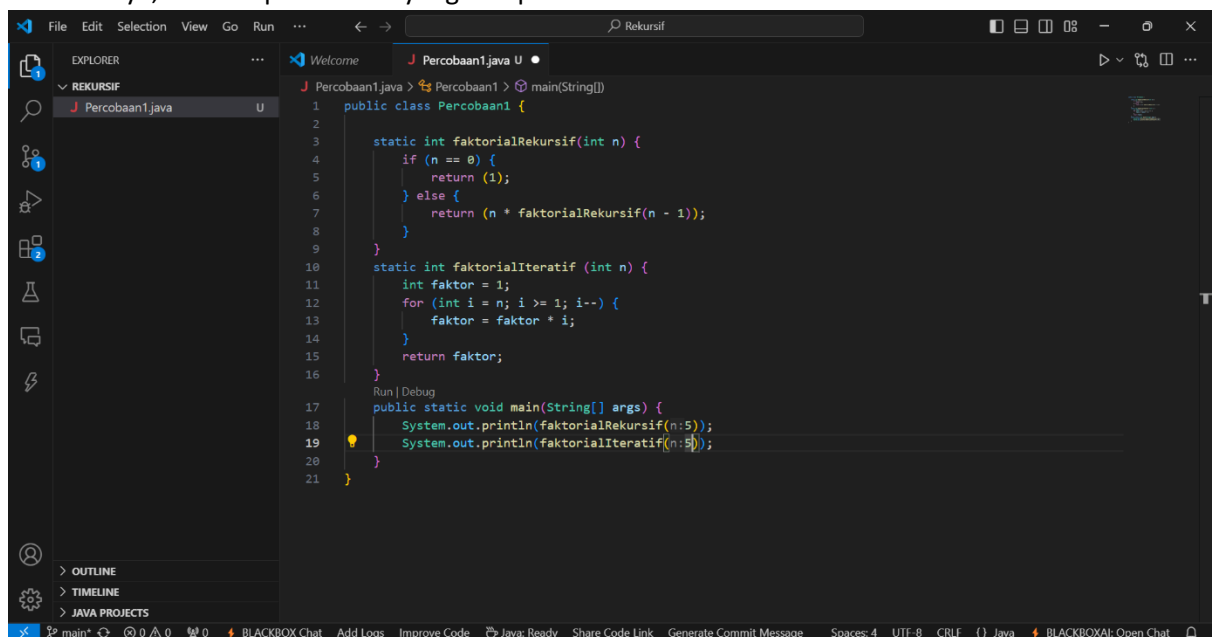
3. Buat lagi fungsi static dengan nama faktorialIteratif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai

faktorialnya.



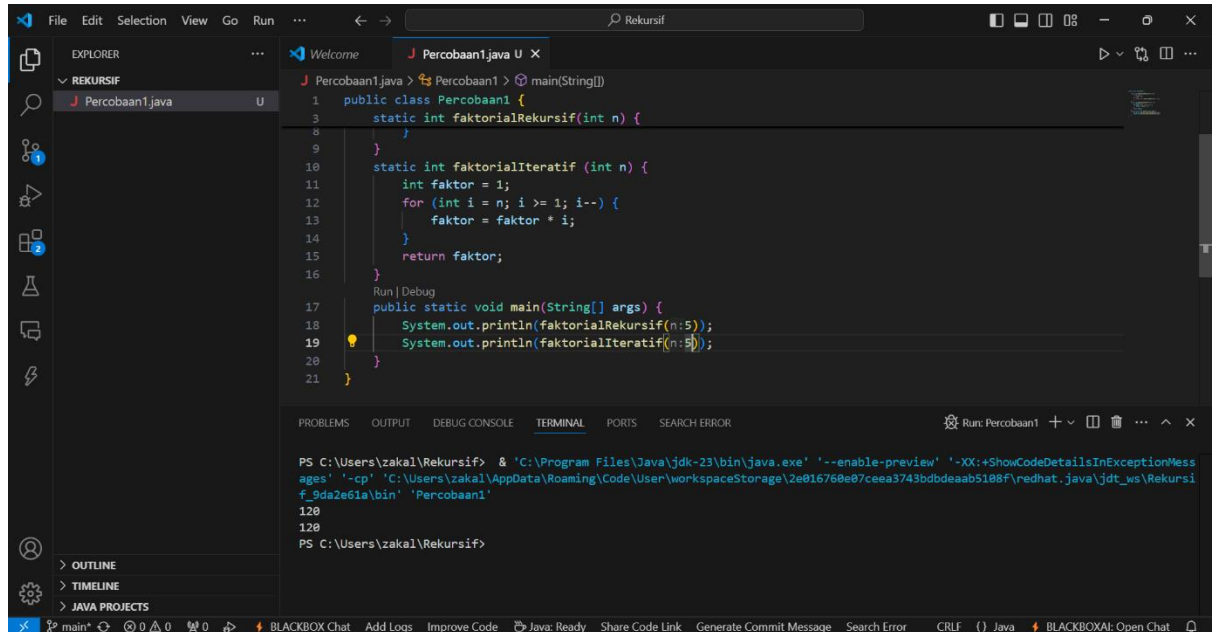
```
1 public class Percobaan1 {
2
3     static int faktorialRekursif(int n) {
4         if (n == 0) {
5             return 1;
6         } else {
7             return (n * faktorialRekursif(n - 1));
8         }
9     }
10
11     static int faktorialIteratif(int n) {
12         int faktor = 1;
13         for (int i = n; i >= 1; i--) {
14             faktor = faktor * i;
15         }
16         return faktor;
17     }
18 }
```

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



```
1 public class Percobaan1 {
2
3     static int faktorialRekursif(int n) {
4         if (n == 0) {
5             return 1;
6         } else {
7             return (n * faktorialRekursif(n - 1));
8         }
9     }
10
11     static int faktorialIteratif(int n) {
12         int faktor = 1;
13         for (int i = n; i >= 1; i--) {
14             faktor = faktor * i;
15         }
16         return faktor;
17     }
18
19     public static void main(String[] args) {
20         System.out.println(faktorialRekursif(5));
21         System.out.println(faktorialIteratif(5));
22     }
23 }
```

5. Jalankan program tersebut. Amati apa yang terjadi!



```
1 public class Percobaan1 {
2     static int faktorialRekursif(int n) {
3         if (n == 1) {
4             return 1;
5         }
6         return n * faktorialRekursif(n - 1);
7     }
8
9     static int faktorialIteratif(int n) {
10        int faktor = 1;
11        for (int i = n; i >= 1; i--) {
12            faktor = faktor * i;
13        }
14        return faktor;
15    }
16
17    public static void main(String[] args) {
18        System.out.println(faktorialRekursif(5));
19        System.out.println(faktorialIteratif(5));
20    }
21 }
```

```
PS C:\Users\zakal\Rekursif> & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceStorage\2e016760e07ceea3743bdbdeaab5108f\redhat_java\jdt_ws\Rekursif_9da2e61a\bin' 'Percobaan1'
120
120
PS C:\Users\zakal\Rekursif>
```

6. Jika ditelusuri, pada saat pemanggilan fungsi faktorialRekursif(5), maka proses yang terjadi dapat diilustrasikan sebagai berikut:

Pertanyaan :

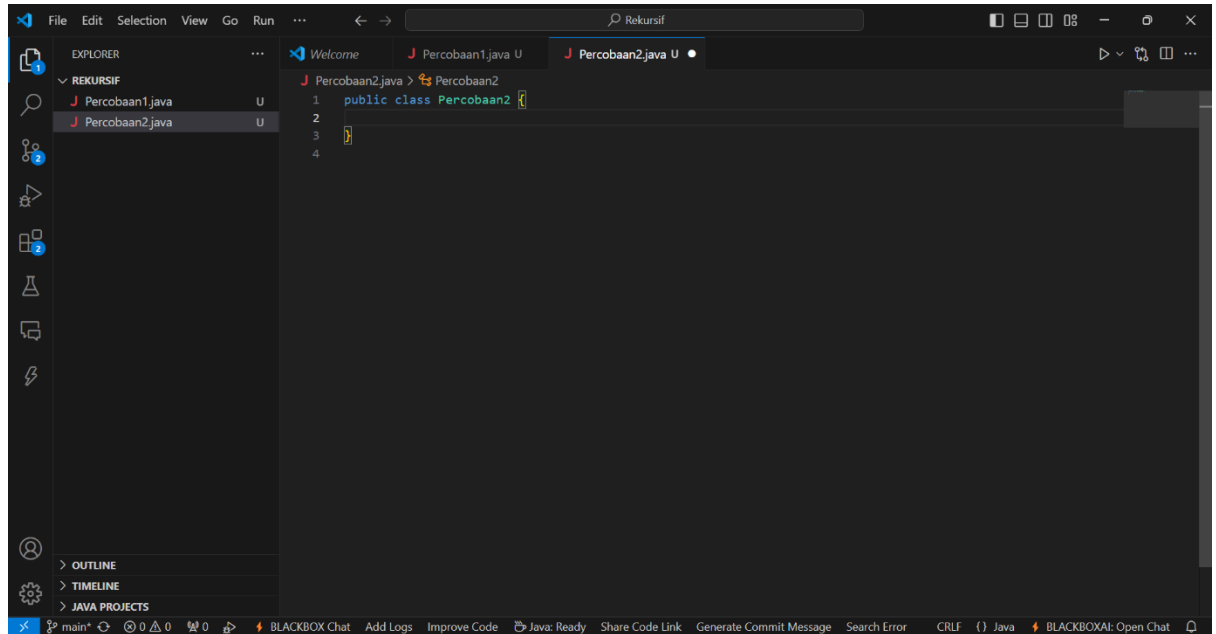
1. Apa yang dimaksud dengan fungsi rekursif?
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?
3. Pada Percobaan1, apakah hasil yang diberikan fungsi faktorialRekursif() dan fungsi faktorialIteratif() sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iteratif!

Jawaban :

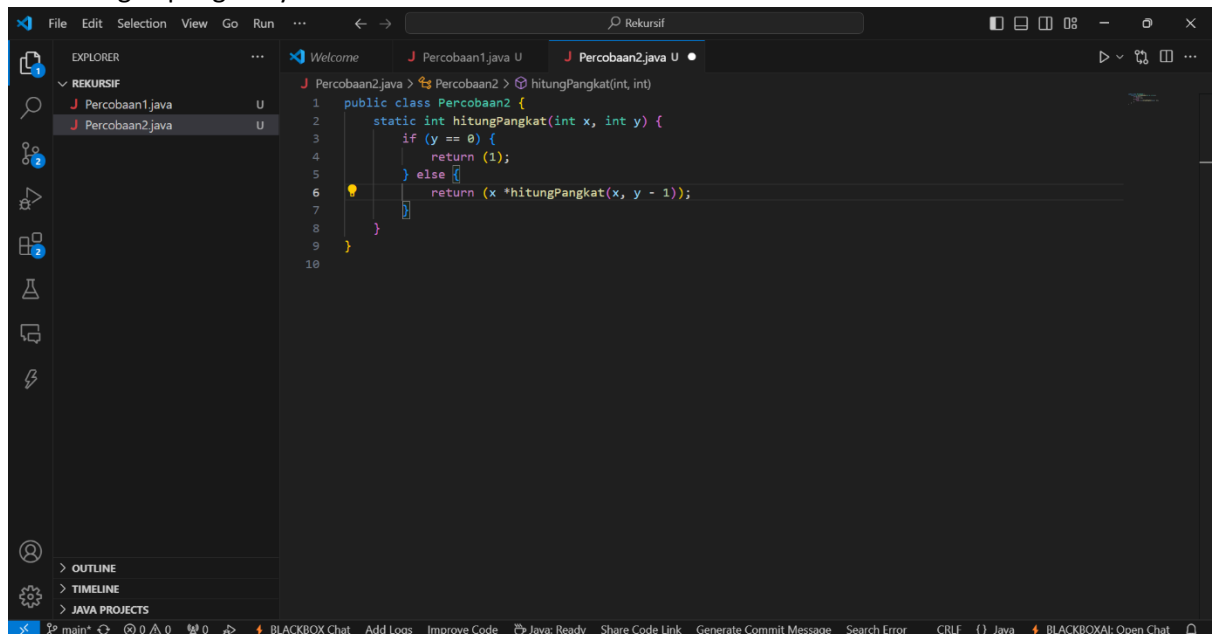
1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri selama proses eksekusi, Fungsi rekursif biasanya digunakan untuk menyelesaikan masalah yang dapat dibagi menjadi sub-masalah yang lebih kecil dan memiliki pola yang berulang
2. Faktorial, deret fibonacci, dan penjumlahan angka dari 1 hingga ke n
3. Fungsi faktorialRekursif() dan faktorialIteratif() memiliki hasil yang sama, perbedaan dari fungsi rekursif dengan iteratif adalah rekursif bergantung pada pemanggilan fungsi yang berulang sedangkan iteratif bergantung pada perulangan

Percobaan 2

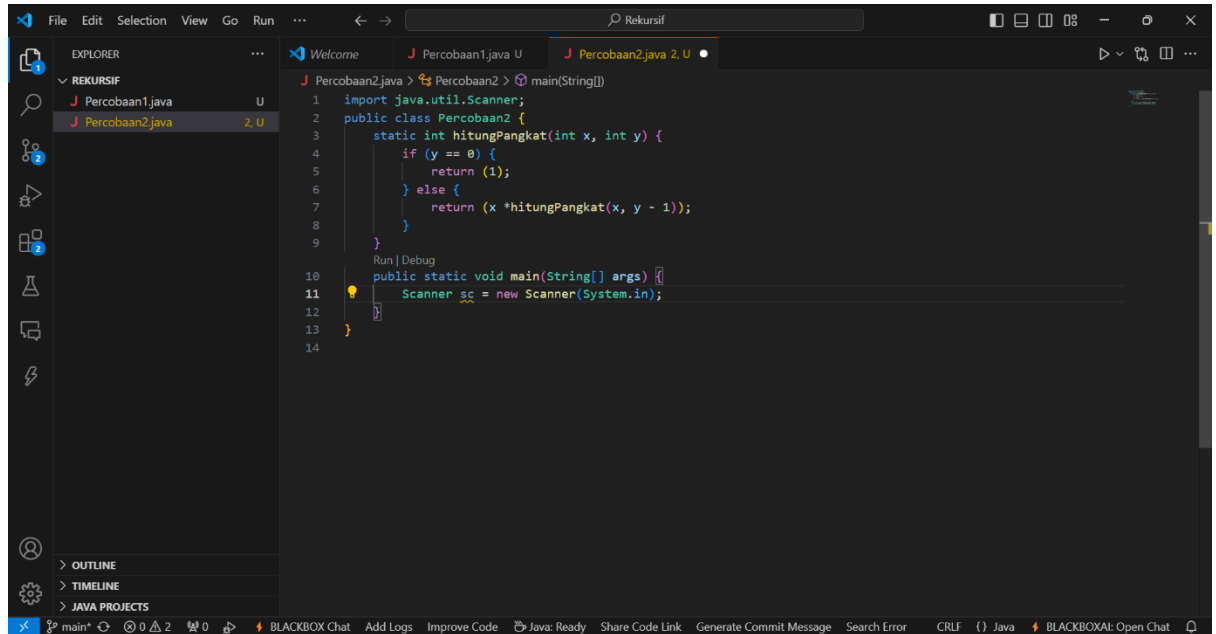
1. Pada project Rekursif, dan buat file Java dengan nama Percobaan2



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.



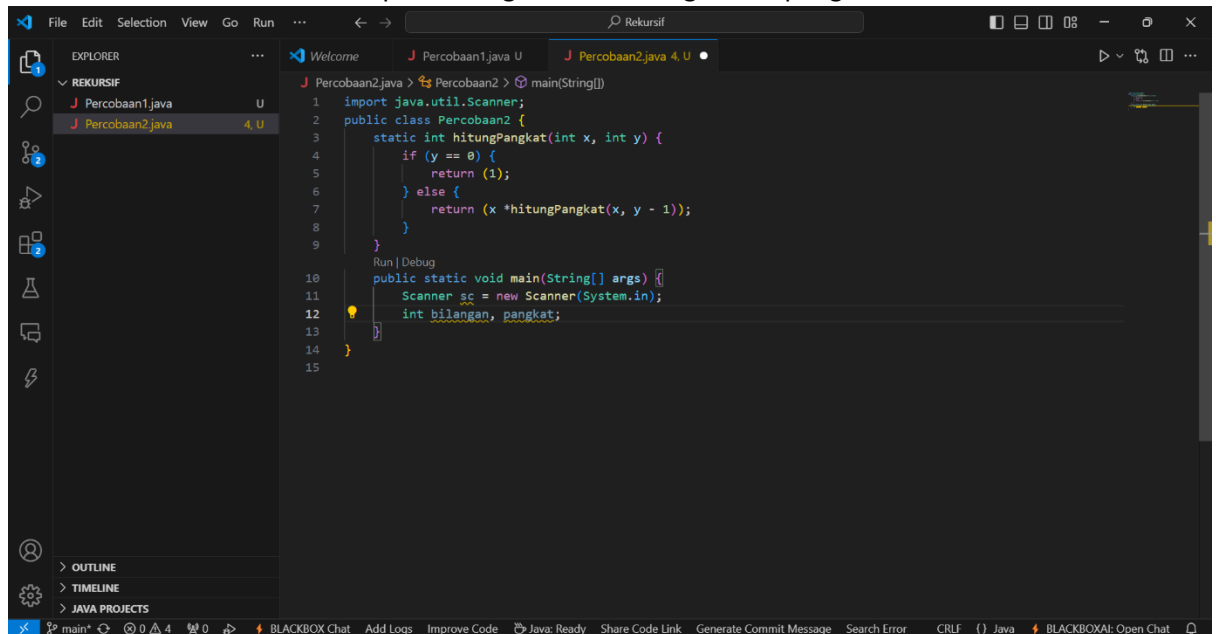
3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc



The screenshot shows an IDE with a project named 'REKURSIF'. The file explorer on the left shows 'Percobaan1.java' and 'Percobaan2.java'. The main editor displays the code for 'Percobaan2.java'. The code includes a recursive function 'hitungPangkat' and a 'main' method that initializes a 'Scanner' object named 'sc'.

```
1 import java.util.Scanner;
2 public class Percobaan2 {
3     static int hitungPangkat(int x, int y) {
4         if (y == 0) {
5             return (1);
6         } else {
7             return (x * hitungPangkat(x, y - 1));
8         }
9     }
10
11     public static void main(String[] args) {
12         Scanner sc = new Scanner(System.in);
13     }
14 }
```

4. Buatlah dua buah variabel bertipe int dengan nama bilangan dan pangkat



The screenshot shows the same IDE as before, but with two additional integer variables, 'bilangan' and 'pangkat', declared in the 'main' method of 'Percobaan2.java'.

```
1 import java.util.Scanner;
2 public class Percobaan2 {
3     static int hitungPangkat(int x, int y) {
4         if (y == 0) {
5             return (1);
6         } else {
7             return (x * hitungPangkat(x, y - 1));
8         }
9     }
10
11     public static void main(String[] args) {
12         Scanner sc = new Scanner(System.in);
13         int bilangan, pangkat;
14     }
15 }
```

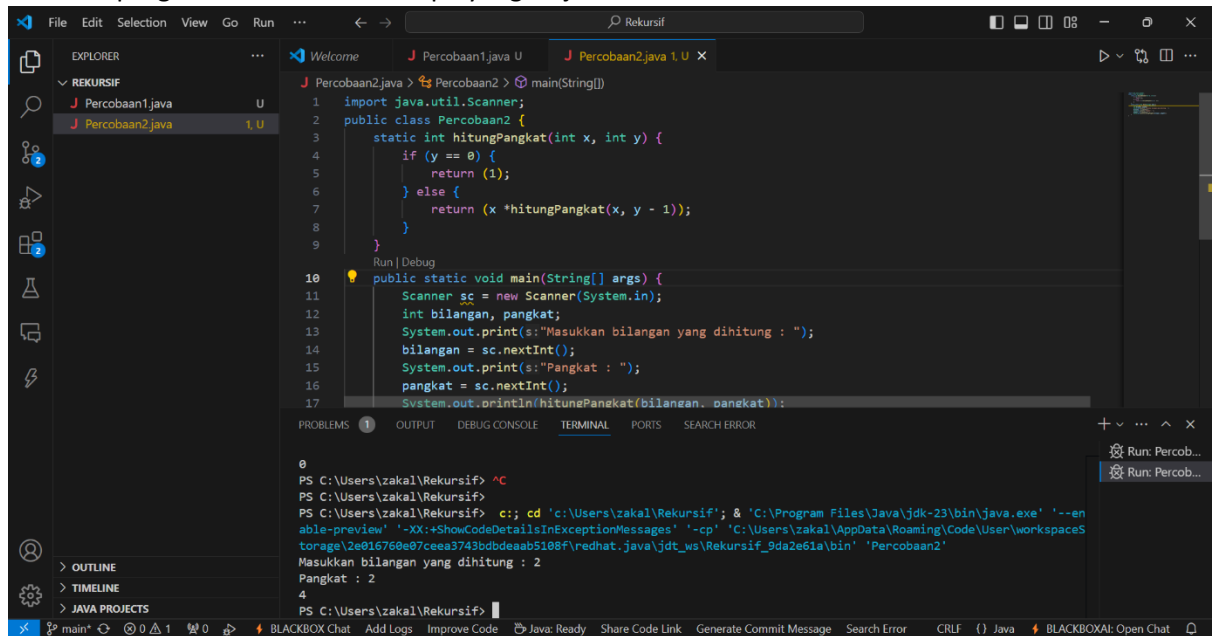
5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
1 import java.util.Scanner;
2 public class Percobaan2 {
3     static int hitungPangkat(int x, int y) {
4         if (y == 0) {
5             return (1);
6         } else {
7             return (x * hitungPangkat(x, y - 1));
8         }
9     }
10     public static void main(String[] args) {
11         Scanner sc = new Scanner(System.in);
12         int bilangan, pangkat;
13         System.out.print("Masukkan bilangan yang dihitung : ");
14         bilangan = sc.nextInt();
15         System.out.print("Pangkat : ");
16         pangkat = sc.nextInt();
17     }
18 }
19
```

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

```
1 import java.util.Scanner;
2 public class Percobaan2 {
3     static int hitungPangkat(int x, int y) {
4         if (y == 0) {
5             return (1);
6         } else {
7             return (x * hitungPangkat(x, y - 1));
8         }
9     }
10     public static void main(String[] args) {
11         Scanner sc = new Scanner(System.in);
12         int bilangan, pangkat;
13         System.out.print("Masukkan bilangan yang dihitung : ");
14         bilangan = sc.nextInt();
15         System.out.print("Pangkat : ");
16         pangkat = sc.nextInt();
17         System.out.println(hitungPangkat(bilangan, pangkat));
18     }
19 }
20
```

7. Jalankan program tersebut. Amati apa yang terjadi!



```
1 import java.util.Scanner;
2 public class Percobaan2 {
3     static int hitungPangkat(int x, int y) {
4         if (y == 0) {
5             return (1);
6         } else {
7             return (x * hitungPangkat(x, y - 1));
8         }
9     }
10    public static void main(String[] args) {
11        Scanner sc = new Scanner(System.in);
12        int bilangan, pangkat;
13        System.out.print("Masukkan bilangan yang dihitung : ");
14        bilangan = sc.nextInt();
15        System.out.print("Pangkat : ");
16        pangkat = sc.nextInt();
17        System.out.println(hitungPangkat(bilangan, pangkat));
18    }
19 }
```

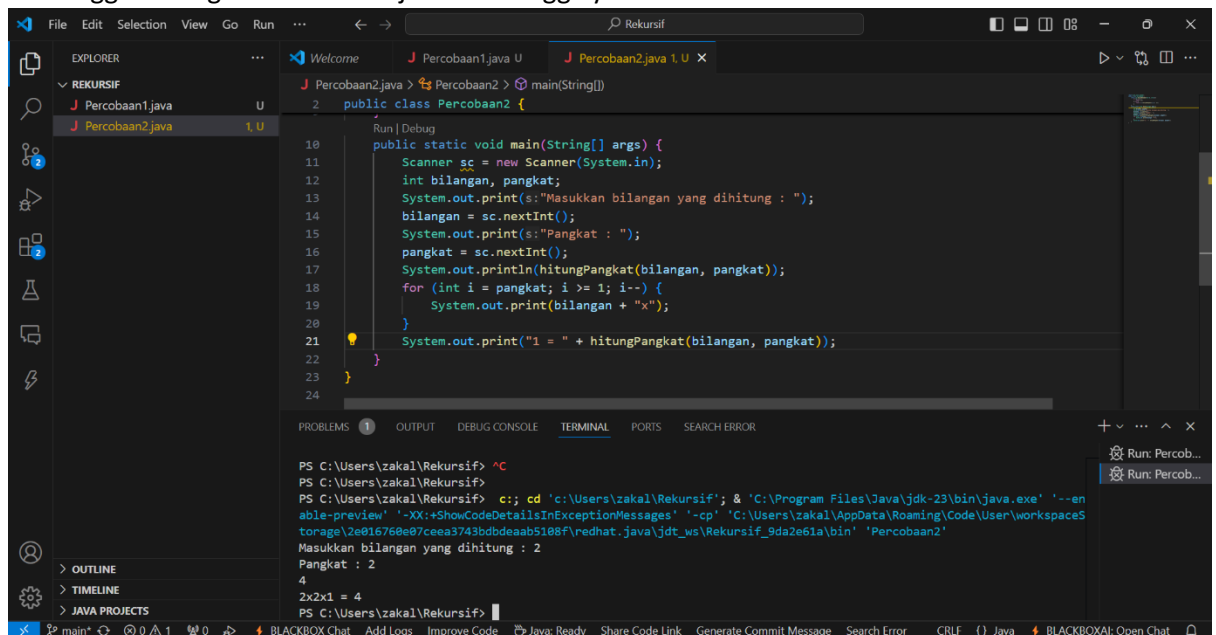
```
PS C:\Users\zakal\Rekursif> ^C
PS C:\Users\zakal\Rekursif>
PS C:\Users\zakal\Rekursif> c:: cd 'c:\Users\zakal\Rekursif'; & 'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceStorage\2e016760e07ceea3743b0bdeab5108f\redhat.java\jdt_ws\Rekursif_9da2e61a\bin' 'Percobaan2'
Masukkan bilangan yang dihitung : 2
Pangkat : 2
4
PS C:\Users\zakal\Rekursif>
```

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 $2 \times 2 \times 2 \times 2 \times 2 = 32$

Jawaban :

1. Pemanggilan fungsi akan terus dijalankan hingga $y = 1$



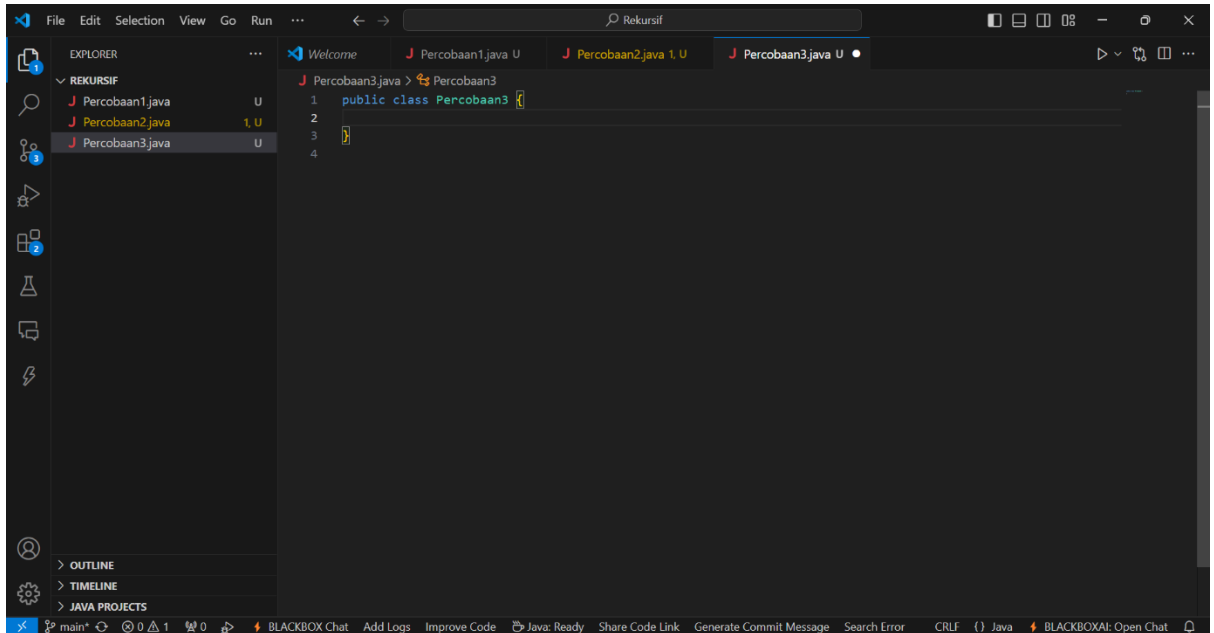
```
2 public class Percobaan2 {
3     static int hitungPangkat(int x, int y) {
4         if (y == 0) {
5             return (1);
6         } else {
7             return (x * hitungPangkat(x, y - 1));
8         }
9     }
10    public static void main(String[] args) {
11        Scanner sc = new Scanner(System.in);
12        int bilangan, pangkat;
13        System.out.print("Masukkan bilangan yang dihitung : ");
14        bilangan = sc.nextInt();
15        System.out.print("Pangkat : ");
16        pangkat = sc.nextInt();
17        System.out.println(hitungPangkat(bilangan, pangkat));
18        for (int i = pangkat; i >= 1; i--) {
19            System.out.print(bilangan + "x");
20        }
21        System.out.print("1 = " + hitungPangkat(bilangan, pangkat));
22    }
23 }
```

```
PS C:\Users\zakal\Rekursif> ^C
PS C:\Users\zakal\Rekursif>
PS C:\Users\zakal\Rekursif> c:: cd 'c:\Users\zakal\Rekursif'; & 'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceStorage\2e016760e07ceea3743b0bdeab5108f\redhat.java\jdt_ws\Rekursif_9da2e61a\bin' 'Percobaan2'
Masukkan bilangan yang dihitung : 2
Pangkat : 2
4
2x2x1 = 4
PS C:\Users\zakal\Rekursif>
```

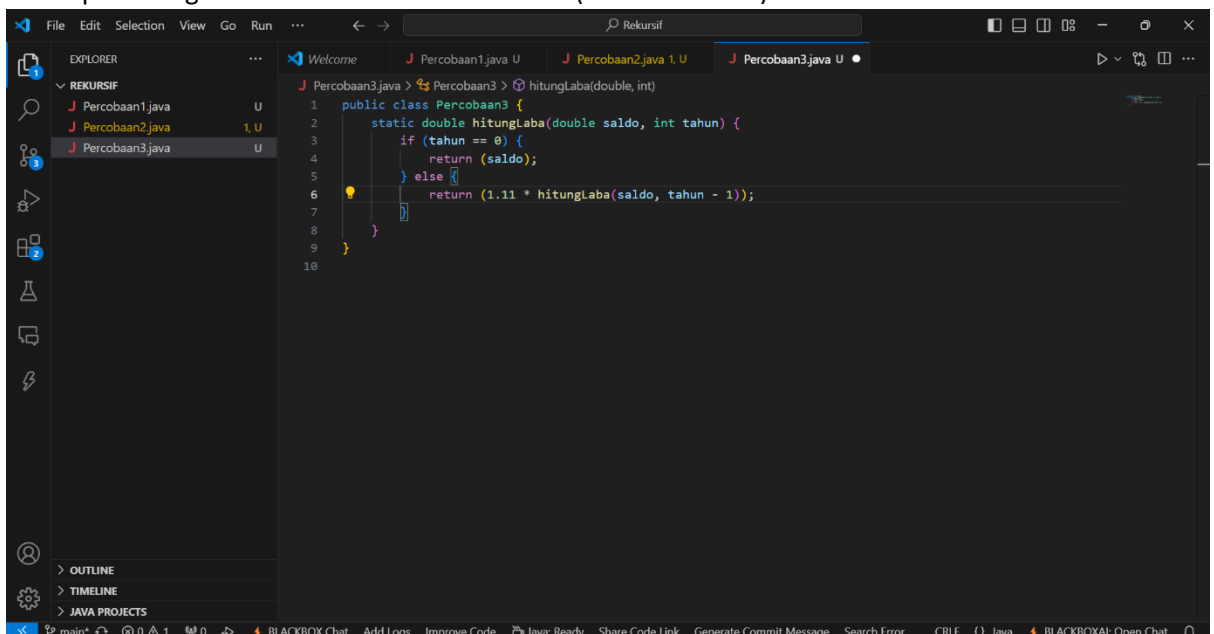
- 2.

Percobaan 3

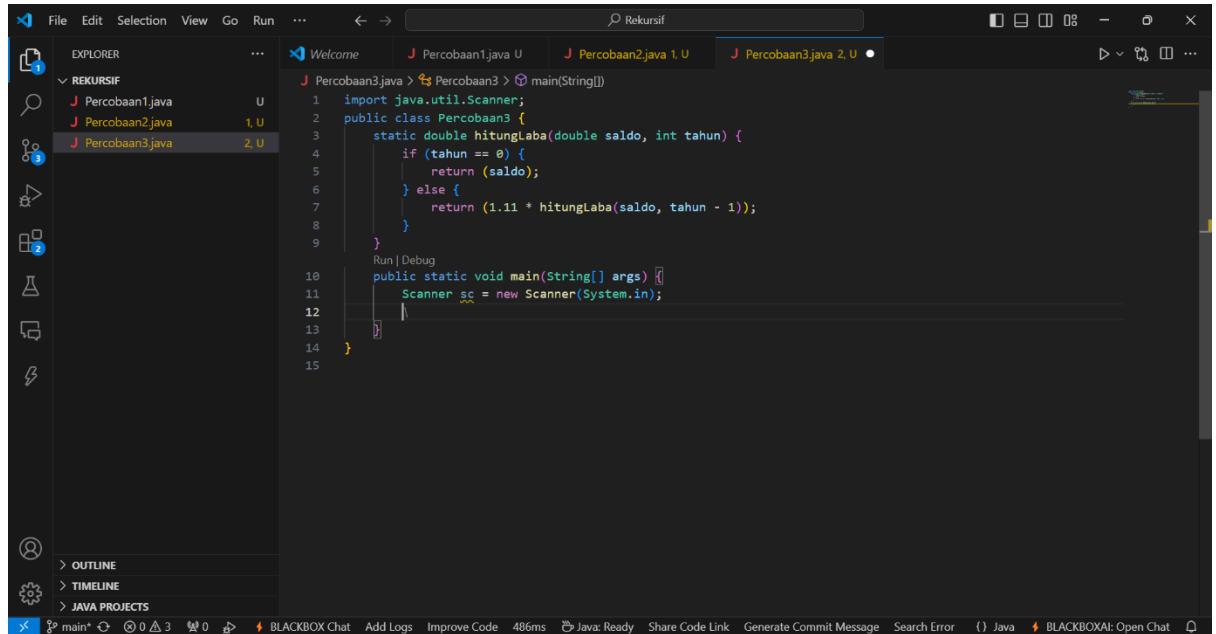
1. Pada project Rekursif, dan buat file Java dengan nama Percobaan3



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 laba * saldo, sehingga untuk menghitung besarnya uang setelah ditambah laba adalah saldo + laba * 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 $1.11 * \text{saldo}$ untuk perhitungan saldo setelah ditambah laba (dalam setahun).



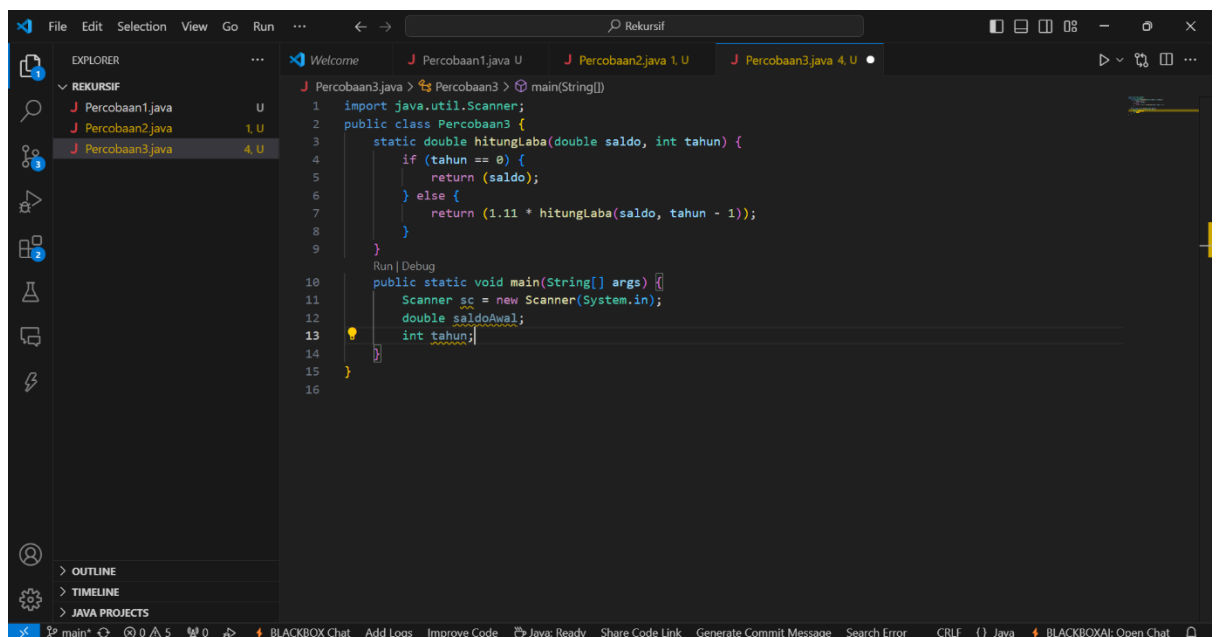
3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc



The screenshot shows an IDE with a project named 'REKURSIF'. The Explorer panel on the left lists three Java files: 'Percobaan1.java', 'Percobaan2.java', and 'Percobaan3.java'. The 'Percobaan3.java' file is selected and open in the editor. The code in the editor is as follows:

```
1 import java.util.Scanner;
2 public class Percobaan3 {
3     static double hitungLaba(double saldo, int tahun) {
4         if (tahun == 0) {
5             return (saldo);
6         } else {
7             return (1.11 * hitungLaba(saldo, tahun - 1));
8         }
9     }
10    Run | Debug
11    public static void main(String[] args) {
12        Scanner sc = new Scanner(System.in);
13    }
14 }
15
```

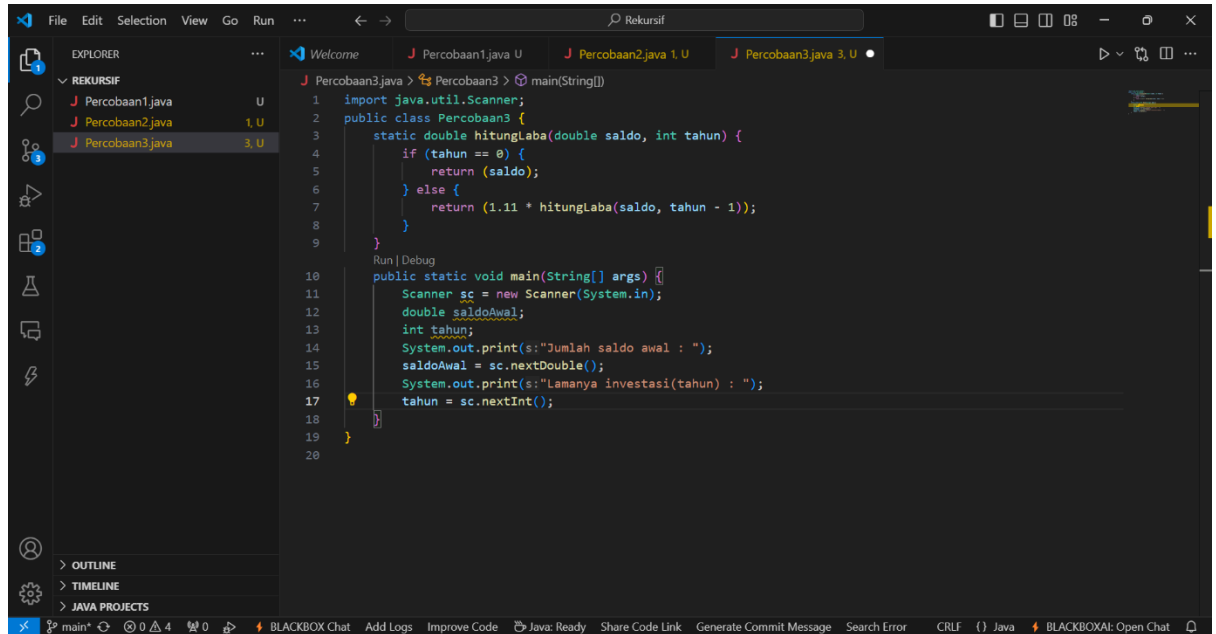
4. Buatlah sebuah variabel bertipe double dengan nama saldoAwal dan sebuah variabel bertipe int bernama tahun



The screenshot shows the same IDE as before, but now the 'Percobaan3.java' file is open at a different point in the code. The code in the editor is as follows:

```
1 import java.util.Scanner;
2 public class Percobaan3 {
3     static double hitungLaba(double saldo, int tahun) {
4         if (tahun == 0) {
5             return (saldo);
6         } else {
7             return (1.11 * hitungLaba(saldo, tahun - 1));
8         }
9     }
10    Run | Debug
11    public static void main(String[] args) {
12        Scanner sc = new Scanner(System.in);
13        double saldoAwal;
14        int tahun;
15    }
16 }
```

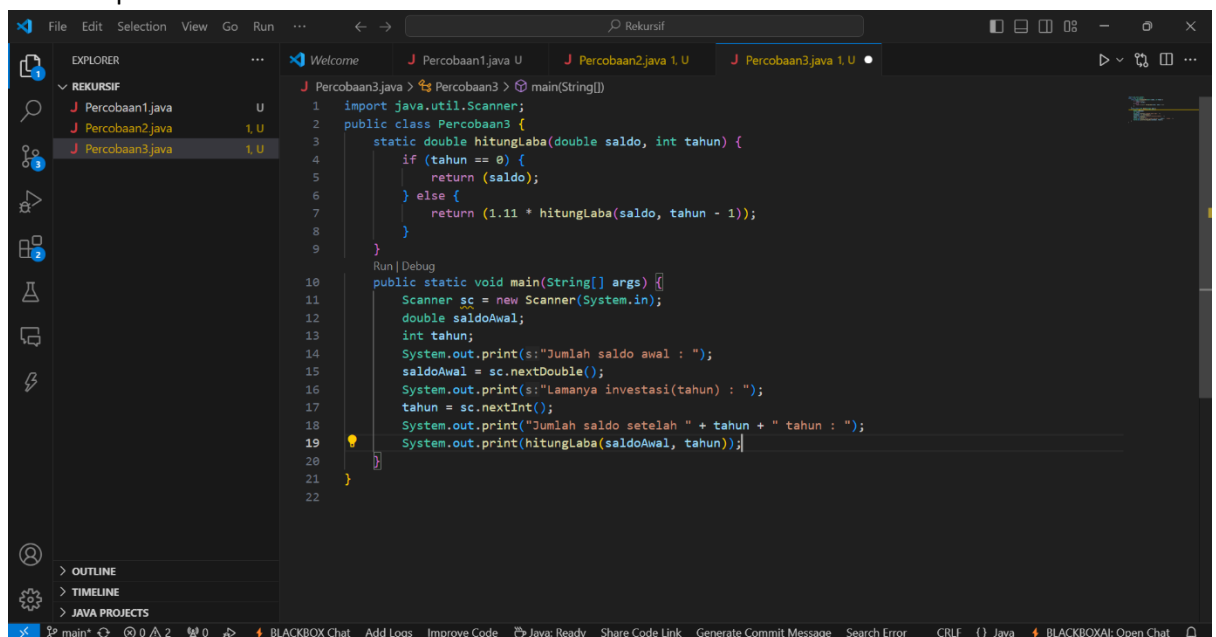
5. Tambahkan kode berikut ini untuk menerima input dari keyboard



The screenshot shows an IDE with a project named 'REKURSIF'. The file explorer on the left shows three Java files: 'Percobaan1.java', 'Percobaan2.java', and 'Percobaan3.java'. The main editor displays the code for 'Percobaan3.java'. The code defines a recursive method 'hitungLaba' that calculates compound interest based on 'saldo' (balance) and 'tahun' (years). The 'main' method uses a 'Scanner' to take user input for the initial balance and the number of years, then calls the 'hitungLaba' method.

```
1 import java.util.Scanner;
2 public class Percobaan3 {
3     static double hitungLaba(double saldo, int tahun) {
4         if (tahun == 0) {
5             return (saldo);
6         } else {
7             return (1.11 * hitungLaba(saldo, tahun - 1));
8         }
9     }
10    Run | Debug
11    public static void main(String[] args) {
12        Scanner sc = new Scanner(System.in);
13        double saldoAwal;
14        int tahun;
15        System.out.print(s:"Jumlah saldo awal : ");
16        saldoAwal = sc.nextDouble();
17        System.out.print(s:"Lamanya investasi(tahun) : ");
18        tahun = sc.nextInt();
19    }
20 }
```

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



The screenshot shows the same IDE as before, but the code in 'Percobaan3.java' has been updated. The 'main' method now includes an additional print statement to display the calculated compound interest after the specified number of years.

```
1 import java.util.Scanner;
2 public class Percobaan3 {
3     static double hitungLaba(double saldo, int tahun) {
4         if (tahun == 0) {
5             return (saldo);
6         } else {
7             return (1.11 * hitungLaba(saldo, tahun - 1));
8         }
9     }
10    Run | Debug
11    public static void main(String[] args) {
12        Scanner sc = new Scanner(System.in);
13        double saldoAwal;
14        int tahun;
15        System.out.print(s:"Jumlah saldo awal : ");
16        saldoAwal = sc.nextDouble();
17        System.out.print(s:"Lamanya investasi(tahun) : ");
18        tahun = sc.nextInt();
19        System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
20        System.out.print(hitungLaba(saldoAwal, tahun));
21    }
22 }
```

7. Jalankan program tersebut. Amati apa yang terjadi!

```

1 public class Percobaan3 {
2     static double hitungLaba(double saldo, int tahun) {
3         return (1.11 * hitungLaba(saldo, tahun - 1));
4     }
5 }
6
7 public static void main(String[] args) {
8     Scanner sc = new Scanner(System.in);
9     double saldoAwal;
10    int tahun;
11    System.out.print("Jumlah saldo awal : ");
12    saldoAwal = sc.nextDouble();
13    System.out.print("Lamanya investasi(tahun) : ");
14    tahun = sc.nextInt();
15    System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
16    System.out.print(hitungLaba(saldoAwal, tahun));
17 }

```

```

Jumlah saldo setelah 2 tahun : 6160.500000000002
PS C:\Users\zakal\Rekursif> ^C
PS C:\Users\zakal\Rekursif>
PS C:\Users\zakal\Rekursif> c:: cd 'c:\Users\zakal\Rekursif'; & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-en
able-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceS
torage\2e016760e07ceea3743b0bdeab5108f\redhat.java\jdt_ws\Rekursif_9da2e61a\bin' 'Percobaan3'
Jumlah saldo awal : 100000
Lamanya investasi(tahun) : 10
Jumlah saldo setelah 10 tahun : 2839420.9860690176
PS C:\Users\zakal\Rekursif>

```

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)

Jawaban :

1. Base case dari percobaan3 adalah tahun = 0, recursion call dari percobaan3 adalah pada bagian return (1.11 * hitungLaba(saldo, tahun – 1));

Fase ekspansi :

Langkah 1:

hitungLaba(100000,3)=1.11×hitungLaba(100000,2)hitungLaba(100000, 3) = 1.11 \times hitungLaba(100000, 2)hitungLaba(100000,3)=1.11×hitungLaba(100000,2)

Langkah 2:

hitungLaba(100000,2)=1.11×hitungLaba(100000,1)hitungLaba(100000, 2) = 1.11 \times hitungLaba(100000, 1)hitungLaba(100000,2)=1.11×hitungLaba(100000,1)

Langkah 3:

hitungLaba(100000,1)=1.11×hitungLaba(100000,0)hitungLaba(100000, 1) = 1.11 \times hitungLaba(100000, 0)hitungLaba(100000,1)=1.11×hitungLaba(100000,0)

Langkah 4 (Base Case):

hitungLaba(100000,0)=100000hitungLaba(100000, 0) = 100000hitungLaba(100000,0)=100000

Fase Substitusi :

Langkah 1:

hitungLaba(100000,0)=100000hitungLaba(100000, 0) = 100000hitungLaba(100000,0)=100000

Langkah 2:

hitungLaba(100000,1)=1.11×100000=111000hitungLaba(100000, 1) = 1.11 \times 100000 = 111000hitungLaba(100000,1)=1.11×100000=111000

Langkah 3:

$\text{hitungLaba}(100000, 2) = 1.11 \times 111000 = 123210$
 $\text{hitungLaba}(100000, 2) = 1.11 \times 111000 = 123210$

Langkah 4:

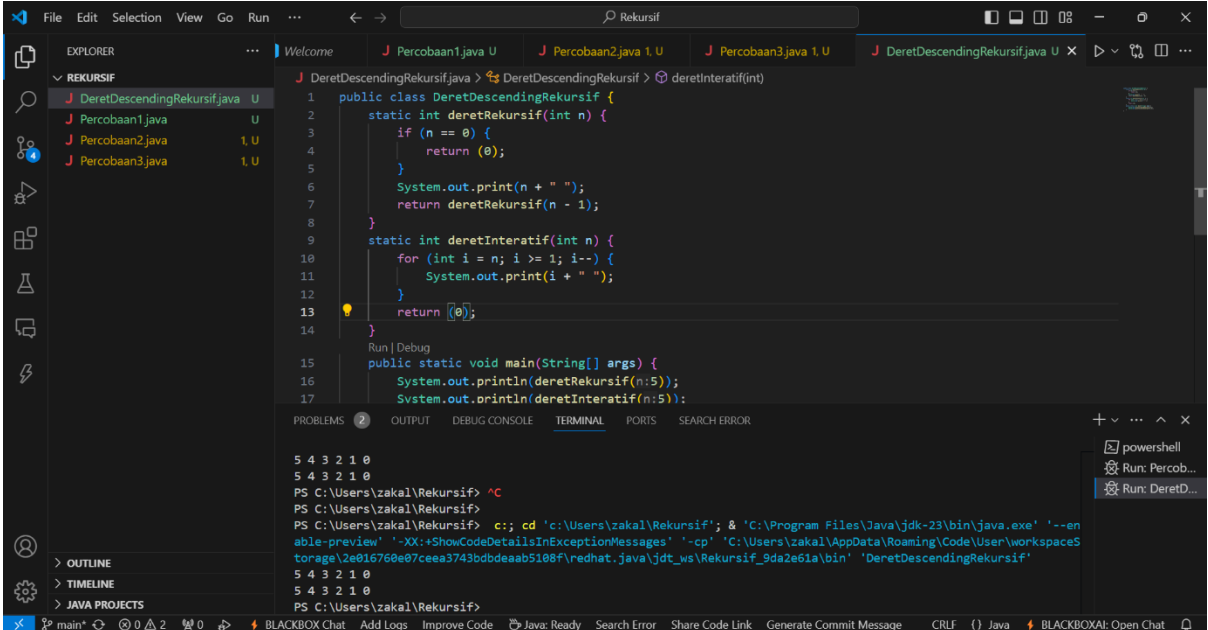
$\text{hitungLaba}(100000, 3) = 1.11 \times 123210 = 136763.1$
 $\text{hitungLaba}(100000, 3) = 1.11 \times 123210 = 136763.1$

Tugas

1. Buatlah program untuk menampilkan bilangan n sampai 0 dengan menggunakan fungsi rekursif dan fungsi iteratif. (DeretDescendingRekursif).
2. Buatlah program yang di dalamnya terdapat fungsi rekursif untuk menghitung penjumlahan bilangan. Misalnya $f = 8$, maka akan dihasilkan $1+2+3+4+5+6+7+8 = 36$ (PenjumlahanRekursif).
3. Sepasang marmut yang baru lahir (jantan dan betina) ditempatkan pada suatu pembiakan. Setelah dua bulan pasangan marmut tersebut melahirkan sepasang marmut kembar (jantan dan betina). Setiap pasangan marmut yang lahir juga akan melahirkan sepasang marmut juga setiap 2 bulan. Berapa pasangan marmut yang ada pada akhir bulan ke-12? Buatlah programnya menggunakan fungsi rekursif! (Fibonacci). Berikut ini adalah ilustrasinya dalam bentuk tabel.

Jawaban :

1.



```
public class DeretDescendingRekursif {
    static int deretRekursif(int n) {
        if (n == 0) {
            return (0);
        }
        System.out.print(n + " ");
        return deretRekursif(n - 1);
    }
    static int deretIteratif(int n) {
        for (int i = n; i >= 1; i--) {
            System.out.print(i + " ");
        }
        return (0);
    }
    public static void main(String[] args) {
        System.out.println(deretRekursif(5));
        System.out.println(deretIteratif(5));
    }
}
```

5 4 3 2 1 0
5 4 3 2 1 0
PS C:\Users\zakal\Rekursif> ^C
PS C:\Users\zakal\Rekursif>
PS C:\Users\zakal\Rekursif> c:; cd 'c:\Users\zakal\Rekursif'; & 'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceStorage\2e01676e07ceea3743bdbdeaab5108f\redhat.java\jdt_ws\Rekursif_9da2e61a\bin' 'DeretDescendingRekursif'
5 4 3 2 1 0
5 4 3 2 1 0
PS C:\Users\zakal\Rekursif>

The bottom status bar shows 'main*' and 'Java: Ready'."/>

2.

3.