

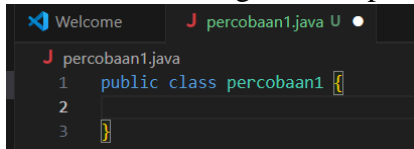
Nama : Siti Nikmatus Sholihah

NIM : 244107020014

Kelas / Prodi / jurusan : 1B / D4 Teknik Informatika / Teknologi Informasi

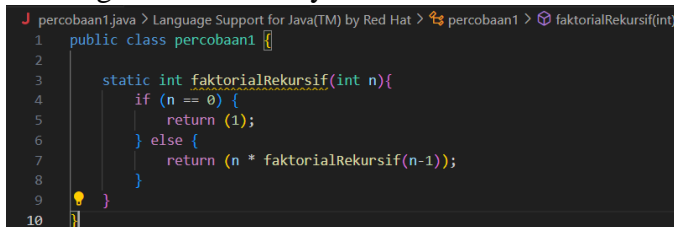
## Percobaan 1

1. Buka Vscode, open folder dengan nama rekursif
2. Buat file baru dengan nama percobaan1.java



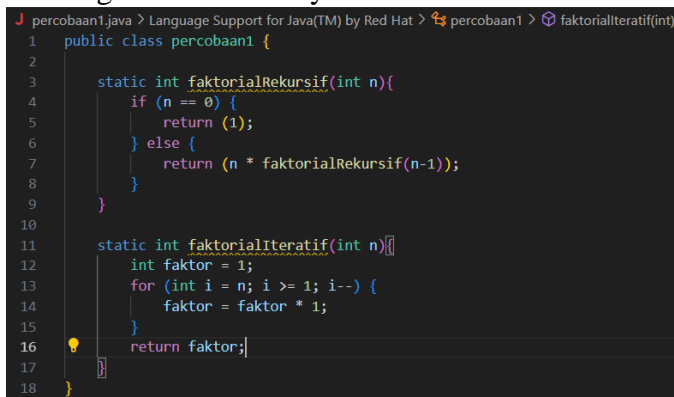
```
1 public class percobaan1 {  
2  
3 }
```

3. Buat fungsi static bernama faktorialRekursif() dengan tipe data pengembalian **int** dan memiliki satu parameter bertipe **int** yang merepresentasikan 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 }
```

4. Buat lagi fungsi static Bernama faktorialIteratif() dengan tipe data kembalian fungsi **int** dan mempunyai 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 }
```

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

```
J percobaan1.java > Language Support for Java(TM) by Red Hat > percobaan1 > main(String[])
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(n:5));
21        System.out.println(faktorialIteratif(n:5));
22    }
23 }
```

6. Jalankan program tersebut.

```
PS C:\Users\Elitebook\rekursif>
120
1
```

## Pertanyaan 1

1. Apa yang dimaksud dengan fungsi rekursif?

Jawab:

Fungsi rekursif adalah sebuah fungsi yang dimana di dalam sebuah fungsi tersebut terdapat perintah untuk memanggil fungsi itu sendiri (dirinya sendiri).

2. Bagaimana contoh kasus penggunaan fungsi rekrusif?

Jawab:

Salah satu contoh kasus penggunaan fungsi rekursif adalah perhitungan faktorial dari sebuah bilangan bulat positif.

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!

Jawab:

Pada pecobaan 1 hasil yang diberikan fungsi **faktorialRekursif()** dan fungsi **faktorialIteratif()** beda. Untuk perbedaan alur programnya dapat dilihat dari pengulangannya, pada **fungsi rekursif** pengulangan akan berhenti ketika base case terpenuhi sedangkan pengulangan **fungsi iteratif** akan berhenti ketika kondisi pengulangan bernilai False.

## Percobaan 2

1. Buat file baru dengan nama percobaan2.java
2. Buat fungsi static dengan nama hitungPangkat(), dengan tipe data kembalian fungsi int dan mempunyai 2 parameter dengan tipe data int berupa bilangan yang akan dihitung pangkatnya dan bilangan pangkatnya.

```
J percobaan2.java > Language Support for Java(TM) by Red Hat > percobaan2
1 public class percobaan2{
2
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 }
```

3. Buat fungsi main dan deklarasikan scanner

```
J percobaan2.java > Language Support for Java(TM) by Red Hat > percobaan2 > main(String[])
1 import java.util.Scanner;
2 public class percobaan2{
3
4     static int hitungPangkat(int x, int y){
5         if ( y == 0 ) {
6             return(1);
7         } else {
8             return (x * hitungPangkat(x, y - 1));
9         }
10    }
11
12    public static void main(String[] args) {
13        Scanner sc = new Scanner(System.in);
14    }
15 }
```

4. Buat 2 variabel bertipe int dengan nama bilangan dan pangkat

```
J percobaan2.java > Language Support for Java(TM) by Red Hat > percobaan2 > main(String[])
1 import java.util.Scanner;
2 public class percobaan2{
3
4     static int hitungPangkat(int x, int y){
5         if ( y == 0 ) {
6             return(1);
7         } else {
8             return (x * hitungPangkat(x, y - 1));
9         }
10    }
11
12    public static void main(String[] args) {
13        Scanner sc = new Scanner(System.in);
14        int bilangan, pangkat;
15    }
16 }
```

5. Tambahkan program untuk menerima input

```
J percobaan2.java > Language Support for Java(TM) by Red Hat > percobaan2 > main(String[])
1 import java.util.Scanner;
2 public class percobaan2{
3
4     static int hitungPangkat(int x, int y){
5         if ( y == 0 ) {
6             return(1);
7         } else {
8             return (x * hitungPangkat(x, y - 1));
9         }
10    }
11
12    public static void main(String[] args) {
13        Scanner sc = new Scanner(System.in);
14        int bilangan, pangkat;
15
16        System.out.print(s:"Bilangan yang dihitung: ");
17        bilangan = sc.nextInt();
18        System.out.print(s:"Pangkat: ");
19        pangkat = sc.nextInt();
20    }
21 }
```

6. Lakukan pemanggilan fungsi `hitungPangkat` yang telah dibuat sebelumnya dengan mengirimkan 2 parameter

```
percobaan2.java > Language Support for Java(TM) by Red Hat > percobaan2
1  import java.util.Scanner;
2  public class percobaan2{
3
4      static int hitungPangkat(int x, int y){
5          if ( y == 0 ) {
6              return(1);
7          } else {
8              return (x * hitungPangkat(x, y - 1));
9          }
10     }
11
12     Run | Debug | Run main | Debug main
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15         int bilangan, pangkat;
16
17         System.out.print(s:"Bilangan yang dihitung: ");
18         bilangan = sc.nextInt();
19         System.out.print(s:"Pangkat: ");
20         pangkat = sc.nextInt();
21
22         System.out.println(hitungPangkat(bilangan, pangkat));
23     }
24 }
```

7. Jalankan program tersebut

```
PS C:\Users\Elitebook\rekursif>
Bilangan yang dihitung: 5
Pangkat: 2
25
```

## 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 rekursif `hitungPangkat(x, y)` dilakukan berulang kali dengan mengurangi nilai `y` hingga mencapai 0. Setelah itu, fungsi mulai mengembalikan hasil perkalian satu per satu, dan akhirnya menghasilkan nilai akhir yang dikembalikan ke fungsi `main()`.

2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : **hitungPangkat(2,5)** dicetak  $2 \times 2 \times 2 \times 2 \times 1 = 32$

Jawab:

```
percobaan2.java > Language Support for Java(TM) by Red Hat > percobaan2 > main(String[])
1  import java.util.Scanner;
2
3  public class percobaan2 {
4
5      static int hitungPangkat(int x, int y, String deret) {
6          if (y == 0) {
7              System.out.print(deret + "1");
8              return 1;
9          } else {
10             return x * hitungPangkat(x, y - 1, deret + x + "x");
11          }
12     }
13
14     Run | Debug | Run main | Debug main
15     public static void main(String[] args) {
16         Scanner sc = new Scanner(System.in);
17         int bilangan, pangkat;
18
19         System.out.print(s:"Bilangan yang dihitung: ");
20         bilangan = sc.nextInt();
21         System.out.print(s:"Pangkat: ");
22         pangkat = sc.nextInt();
23
24         System.out.print(s:"Deret perhitungan: ");
25         int hasil = hitungPangkat(bilangan, pangkat, deret:"");
26         System.out.println(" = " + hasil);
27     }
28 }
```

### Percobaan 3

1. Buat file Java baru dengan nama Percobaan3.java.
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.

```
J percobaan3.java > percobaan3 > main(String[] args)
1 public class percobaan3{
2
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 }
```

3. Buat fungsi main dan deklarasikan scanner

```
J percobaan3.java > Language Support for Java(TM) by Red Hat > percobaan3 > main(String[])
1 import java.util.Scanner;
2
3 public class percobaan3{
4
5     static double hitungLaba(double saldo, int tahun){
6         if (tahun == 0){
7             return (saldo);
8         } else {
9             return (1.11 * hitungLaba(saldo, tahun - 1));
10        }
11    }
12
13    public static void main(String[] args) {
14        Scanner sc = new Scanner(System.in);
15    }
16 }
```

4. Buat sebuah variabel dengan tipe data double dengan nama saldoAwal dan sebuah tipe data int dengan nama tahun

```
J percobaan3.java > Language Support for Java(TM) by Red Hat > percobaan3 > main(String[])
1 import java.util.Scanner;
2
3 public class percobaan3{
4
5     static double hitungLaba(double saldo, int tahun){
6         if (tahun == 0){
7             return (saldo);
8         } else {
9             return (1.11 * hitungLaba(saldo, tahun - 1));
10        }
11    }
12
13    public static void main(String[] args) {
14        Scanner sc = new Scanner(System.in);
15
16        double saldoAwal;
17        int tahun;
18    }
19 }
```

5. Tambahkan kode program untuk menerima input

```
J percobaan3.java > Language Support for Java(TM) by Red Hat > percobaan3 > main(String[])
1 import java.util.Scanner;
2
3 public class percobaan3{
4
5     static double hitungLaba(double saldo, int tahun){
6         if (tahun == 0){
7             return (saldo);
8         } else {
9             return (1.11 * hitungLaba(saldo, tahun - 1));
10        }
11    }
12
13    public static void main(String[] args) {
14        Scanner sc = new Scanner(System.in);
15
16        double saldoAwal;
17        int tahun;
18
19        System.out.print(s:"Jumlah saldo awal: ");
20        saldoAwal = sc.nextDouble();
21        System.out.print(s:"Lamanya investasi (tahun): ");
22        tahun = sc.nextInt();
23    }
24 }
```

6. Lakukan pemanggilan fungsi hitungLaba yang telah dibuat sebelumnya

```
percobaan3.java > Language Support for Java(TM) by Red Hat > percobaan3
1  import java.util.Scanner;
2
3  public class percobaan3{
4
5      static double hitungLaba(double saldo, int tahun){
6          if (tahun == 0){
7              return (saldo);
8          } else {
9              return (1.11 * hitungLaba(saldo, tahun - 1));
10         }
11     }
12
13     Run | Debug | Run main | Debug main
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16
17         double saldoAwal;
18         int tahun;
19
20         System.out.print(s:"Jumlah saldo awal: ");
21         saldoAwal = sc.nextDouble();
22         System.out.print(s:"Lamanya investasi (tahun): ");
23         tahun = sc.nextInt();
24
25         System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
26         System.out.print(hitungLaba(saldoAwal, tahun));
27     }
28 }
```

7. Jalankan kode program

```
PS C:\Users\Elitebook\rekursif> cd "c:\Users\Elitebook\rekursif"; if ($?) { javac percobaan3.java } ;
($?) { java percobaan3 }
Jumlah saldo awal: 20000
Lamanya investasi (tahun): 2
Jumlah saldo setelah 2 tahun : 24642.000000000007
```

Pertanyaan 3

1. Pada **Percobaan3**, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!

Jawab:

Base case = `if (tahun == 0){`

Recursion call = `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:

**Ekspansi**

1. `hitungLaba(100000, 3) → hitungLaba(100000, 2)`
2. `hitungLaba(100000, 2) → hitungLaba(100000, 1)`
3. `hitungLaba(100000, 1) → hitungLaba(100000, 0)`

**Substitusi**

1. `hitungLaba(100000, 0) → 100000`
2. `hitungLaba(100000, 1) → 1.11 × 100000 = 111000`  
`1.11 × 111000 = 123210`
3. `hitungLaba(100000, 2) → 1.11 × 123210 = 136763.1`  
`1.11 × 136763.1 = 151807.041`
4. `hitungLaba(100000, 3) → 1.11 × 151807.041 = 168505.81551`