PERTEMUAN KEDELAPAN

Static Keyword, Static Method, Static Blocks

Prepared by Adi Wahyu Pribadi

Static Keyword

Sebagaimana yang telah diketahui bahwa untuk mengakses anggota Class, kita harus membuat instance/objek terlebih dahulu dari Class tersebut. Namun, terkadang terdapat situasi di mana, kita ingin mengakses anggota Class tanpa harus membuat objek terlebih dahulu.

Pada situasi tersebut, kita gunakan keyword static sebagai solusinya. Jadi kita harus mendeklarasikan anggota Class yang static (*declare the class members static*).

Sebagai contoh terdapat `Class Math` di Java yang hampis semua membernya static. Sehingga kita dapat mengaksesnya tapa harus membuat instances dari `Class Math`.

```
package app;

public class App {
    public static void main(String[] args) throws Exception {
        // accessing the methods of the Math class
        System.out.println("Absolute value of -12 =  " + Math.abs(-12));
        System.out.println("Value of PI = " + Math.PI);
        System.out.println("Value of E = " + Math.E);
        System.out.println("2^2 = " + Math.pow(2,2));
    }
}
```

Output:

```
Absolute value of -12 =  12
Value of PI = 3.141592653589793
Value of E = 2.718281828459045
2^2 = 4.0
```

Pada contoh di atas, kita tidak perlu membuat object apapun dari `Class Math`. Walau begitu kita mengakses method `abs()`, `pow()` dan variabel `PI` dan `E`.

Static Methods

Static Method adalah method yang dapat diakses langsung dari nama Classnya.

```
package app;
class StaticTest {

    // non-static method
    int multiply(int a, int b){
        return a * b;
    }

    // static method
    static int add(int a, int b){
        return a + b;
    }
}

public class App {
   public static void main( String[] args ) {

        // create an instance of the StaticTest class
        StaticTest st = new StaticTest();

        // call the nonstatic method
        System.out.println(" 2 * 2 = " + st.multiply(2,2));

        // call the static method
        System.out.println(" 2 + 3 = " + StaticTest.add(2,3));
   }
}
```

Output:

```
 2 * 2 = 4
 2 + 3 = 5
```

Pada contoh di atas terdapat non-static method bernama `multiply()` dan static method `add` di dalam class StaticTest. Di class App, kita liat bahwa non-static method diakses melalu objek `st` menggunakan `st.multiply(2, 2)`. Sedangkan kita dapat langsung memanggil static method menggunakan `StaticTest.add(2, 3)` tanpa harus membuat objek.

Static Variabels

Static Varibles adalah variable yang dapat diakses langsung dari nama Classnya.

```
package app;
class Test {
   // static variable
   static int max = 10;
   // non-static variable
   int min = 5;
}

public class App {
   public static void main(String[] args) {
       Test obj = new Test();
       // access the non-static variable
       System.out.println("min + 1 = " + (obj.min + 1));
       // access the static variable
       System.out.println("max + 1 = " + (Test.max + 1));
   }
}
```

Output:

```
min + 1 = 6
max + 1 = 11
```

Pada contoh di atas, terdapat non-static variable `min` dan static variable `max` di dalam class `Test`. Di dalam App class, kita dapat memanggil non-static variable melalui objek `obj` dengan cara `obj.min + 1`. Sedangakn untuk memanggil static variable cukup gunakan nama class `Test.max + 1`.

Catatan: Static variables jarang digunakan di Java. Sebagai gantinya, digunakan static constants. Static constants didefinisikan dengan keyword static final dan ditulis dengan huruf besar. Inilah mengapa, static variables juga sering ditulis dalam huruf besar.

Mengakses Static Variables dan Methods di dalam Class

Pada contoh-contoh sebelumnya kita mengakses static variables dan static methods dari Class yang berbeda. Namun, mengakses static variables dan static method dari dalam Class juga bisa dilakukan secara langsung.

```
package app;
public class App {
   // static variable
   static int age;
   // static method
   static void display() {
       System.out.println("Static Method");
   }
   public static void main(String[] args) {

       // access the static variable
       age = 30;
       System.out.println("Age is " + age);

       // access the static method
       display();
   }
}
```

Output:

```
Age is 30
Static Method
```

Pada contoh di atas, kita dapat mengakses static variable dan static method langsung tanpa menggunakan nama Class. Static variable dan static method defaultnya adalah public. Karena kita mengakses dari Class yang sama, kita tidak perlu menyebutkan nama Class-nya.

Static Blocks

Static blocks digunakan untuk inisiasi static variable.

```
package app;
class App {
   // static variables
   static int a = 23;
   static int b;
   static int max;
   // static blocks
   static {
       System.out.println("First Static block.");
       b = a * 4;
   }
   static {
       System.out.println("Second Static block.");
       max = 30;
   }
   // static method
   static void displaystatic() {
       System.out.println("a = " + a);
       System.out.println("b = " + b);
       System.out.println("max = " + max);
   }

   public static void main(String args[]) {
       // calling the static method
       displaystatic();
   }
}
```

Output:

```
First Static block.
Second Static block.
a = 23
b = 92
max = 30
```