



Modul Praktikum Pemrograman Berorientasi Objek

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Lampung

2022
Semester Ganjil

Modul 8. Static Method

Pada Java, kata kunci static dapat digunakan pada variabel, constant, dan method. Tujuan utama penggunaan kata kunci static adalah untuk mengatur memori agar dapat digunakan secara efisien.

Capaian Pembelajaran

1. Mahasiswa mampu membuat static method
2. Mahasiswa mampu menggunakan static method bawaan java
3. Mahasiswa mampu menggunakan static method buatan

Materi

Kata kunci static

Static digunakan untuk membuka akses suatu variabel/method kepada semua instance. Apabila suatu method/variabel diberi kata kunci static maka method/variabel tersebut dapat diakses dimanapun tanpa perlu membuat objek classnya terlebih dahulu.

Kata kunci final

kata kunci final digunakan dalam konteks yang berbeda-beda. Final adalah non-access modifier yang hanya berlaku untuk variabel, method, atau class. Berikut ini adalah konteks yang berbeda di mana final digunakan.

Final Variable	→	To Create constant variable
Final Methods	→	Prevent Method Overriding
Final Classes	→	Prevent Inheritance

Variabel final dapat diinisialisasi secara eksplisit hanya sekali. Variabel reference yang dideklarasikan final tidak akan pernah dapat dipindahkan untuk merujuk ke objek yang berbeda.

Namun, data di dalam objek dapat diubah. Jadi, state objek dapat diubah tetapi referensi ke objek tersebut tidak dapat diubah. Final modifier pada variabel sering digunakan dengan keyword static untuk membuat variable class constant.

Ketika sebuah method dideklarasikan dengan kata kunci final, method tersebut tidak dapat override. Sedangkan pada class, class tersebut tidak dapat di inherit.

Static Variable

Static variable merupakan variabel yang dimiliki oleh sebuah class, bukan objek dari class tersebut. Variabel ini akan diinisialisasi terlebih dahulu, sebelum inisialisasi variabel instance yang lain..

1. Satu salinan untuk dibagikan untuk semua instance class.
2. Static variable dapat diakses langsung dengan nama class dan tidak memerlukan objek apa pun.
3. Static variable bernilai sama untuk semua objek.

Pengertian Static Method

Static method di Java adalah method yang merupakan bagian dari class, bukan dari instance dari sebuah class. Setiap instance dari class memiliki akses ke static method. Static method memiliki akses ke variabel class (static variable) tanpa menggunakan objek class (instance). Artinya, tanpa membuat objek sekalipun, static method tetap dapat digunakan.

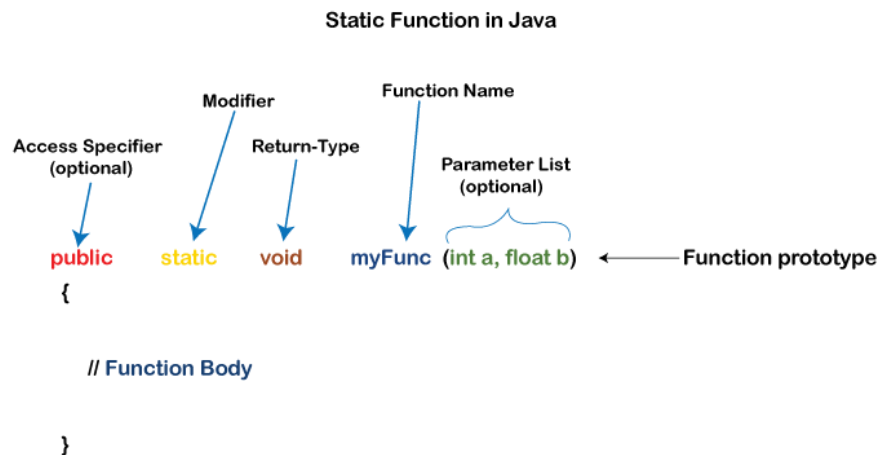
Salah satu contoh static method adalah fungsi yang terdapat dalam library math seperti

```
Math.sqrt(10);
```

Tanpa membuat objek Math, method sqrt() tetap dapat digunakan, sedangkan instance method seperti Scanner, anda harus membuat objek Scanner terlebih dahulu untuk menggunakan method yang ada didalamnya.

Cara penulisan static method:





Fitur Static Method

1. Static method di Java adalah method yang merupakan bagian dari class bukan turunan dari class itu.
2. Setiap instance dari class memiliki akses ke method.
3. Static method memiliki akses ke variabel class (static variable) tanpa menggunakan objek class (instance).
4. Hanya static variable yang dapat diakses dengan static method. Static method tidak dapat mengakses data yang tidak static (instance variable).
5. Dalam static method dan non-static, metode static dapat diakses secara langsung.

Batasan Static Method

Ada dua batasan utama dalam static method. yaitu:

1. Static method tidak dapat menggunakan anggota data non-static atau memanggil method non-static secara langsung.
2. kata kunci **this** dan **super** tidak dapat digunakan dalam context static method.

Perbedaan Static Method dengan Instance Method

Instance Methods	Static Methods
Membutuhkan objek class.	Tidak memerlukan instansiasi objek class.

Dapat mengakses semua atribut class.	Hanya dapat mengakses atribut static dari suatu class.
Method hanya dapat diakses dengan menggunakan referensi objek.	Method ini dapat diakses dengan nama class.
Syntax: <code>Objref.methodname()</code>	Syntax: <code>className.methodname()</code>
It's an example of pass-by-value programming.	It is an example of pass-by-reference programming.

Static Nested Class

Kata kunci `static` digunakan untuk membuat sebuah nested class, atau class yang berada di dalam class lainnya. Tanpa kata kunci `static` anda tidak bisa membuat class di dalam class.

Contoh :

```
public class Main {
    public static class NestedClass {

    }
}
```

Nested class hanya bisa digunakan pada class utamanya sehingga tidak memungkinkan apabila dipanggil di class lainnya. Dengan menggunakan kata kunci `static`, untuk menggunakan nested class tidak perlu membuat objek classnya terlebih dahulu.

Langkah Praktikum

Membuat Project

1. Buka Netbeans
2. Buat project baru dengan nama **Praktikum9_nama**

Membuat Atribut Static

1. Pada class utama buatlah sebuah atribut **String atributBukanStatic** dengan access modifier public, kemudian isi nilainya dengan **"Atribut Bukan Static"**
2. Kemudian buat juga sebuah atribut **String atributStatic** kali ini dengan access modifier **static**, kemudian isi nilainya dengan **"Atribut Static"**

```
public String atributBukanStatic = "Atribut Bukan Static";  
static String atributStatic = "Atribut Static";
```

3. Kemudian cobalah tampilkan kedua atribut tersebut pada method main.

```
public static void main(String[] args) {  
    System.out.println(atributBukanStatic);  
    System.out.println(atributStatic);  
}
```

Output yang dihasilkan dari program tersebut adalah

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
 Cannot make a static reference to the non-static field atributBukanStatic

Program tersebut error dikarenakan kita mencoba memanggil atribut non-static secara langsung tanpa membuat object terlebih dahulu.

Membuat method static

1. Pada class utama, buatlah sebuah method dengan nama **bukanStatic()** yang akan menampilkan pesan **"Bukan Method Static"**
2. Kemudian buat juga sebuah method static dengan nama **methodStatic()** yang akan menampilkan pesan **"Method Static"**

```
public void bukanStatic(){
    System.out.println("Bukan Static");
}

static void methodStatic(){
    System.out.println("Method Static");
}
```

3. Kemudian cobalah untuk memanggil kedua method tersebut pada method main, tanpa membuat objek.

```
public static void main(String[] args) {
    bukanStatic();
    methodStatic();
}
```

Hasil running dari program tersebut akan error dan menampilkan pesan seperti ini :

```
Exception in thread "main" java.lang.Error: Unresolved compilation
problem:
    Cannot make a static reference to the non-static method
    bukanStatic()
```

Program tersebut error dikarenakan kita mencoba memanggil method non-static secara langsung tanpa membuat object terlebih dahulu.

Membuat Objek untuk atribut dan method non-static

1. Sekarang buatlah sebuah objek dari class utama anda untuk memanggil atribut dan method non-static.
2. Kemudian panggil kembali method dan atribut non-static yang sebelumnya telah dibuat

```

public static void main(String[] args) {
    Praktikum praktikum = new Praktikum();
    // Atribut
    System.out.println(praktikum.atributBukanStatic);
    System.out.println(atributStatic);

    // Method
    praktikum.bukanStatic();
    methodStatic();
}

```

Hasil running dari program tersebut adalah :

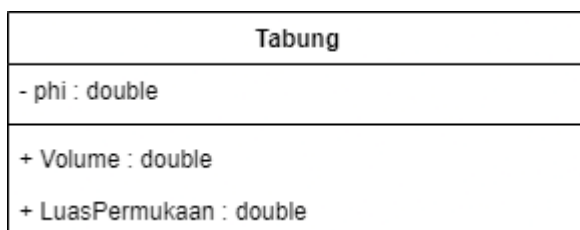
```

Atribut Bukan Static
Atribut Static
Bukan Static
Method Static

```

Membuat Class dengan method static

1. Buatlah sebuah class **Tabung** dan buatlah atribut serta methodnya sesuai dengan class diagram berikut



2. Atribut **phi** memiliki acces modifier **private static final** yang memiliki nilai **3.14**
3. Method Volume dan LuasPermukaan memiliki access modifier **public static**
4. Method Volume dan LuasPermukaan memiliki 2 parameter **double** yaitu **jari-jari** dan **tinggi**. Kedua method tersebut akan mengembalikan nilai **double**.
5. Gunakan atribut **phi** yang telah dibuat sebelumnya untuk menghitung volume dan luas permukaan tabung.

Memanggil method static

1. Gunakan method static yang telah dibuat pada class **Tabung** untuk menghitung volume dan Luas Permukaan tabung pada class main.

2. Hitunglah berapa volume dan luas permukaan sebuah tabung yang memiliki jari-jari 13 cm dan tinggi 30 cm.

```
public static void main(String[] args) {
    double r = 13;
    double t = 30;
    double vol = Tabung.Volume(r, t);
    double lp = Tabung.LuasPermukaan(r, t);

    System.out.println(vol);
    System.out.println(lp);
}
```

Output dari program tersebut antara lain :

```
15927.874753700251
3512.300586713389
```

Membuat Static Class

1. Pindahkan class **Tabung** yang telah dibuat sebelumnya ke dalam class utama.
2. Tambahkan access modifier **static** pada class tersebut agar menjadi nested static class.

```
public class Praktikum {
    ...

    public static class Tabung {
        private static final double phi = 3.14;

        public static double Volume(double jari, double tinggi) {
            double volume = phi * jari * jari * tinggi;
            return volume;
        }

        public static double LuasPermukaan(double jari, double tinggi) {
            double luasPermukaan = 2 * phi * jari * (jari + tinggi);
            return luasPermukaan;
        }
    }
}
```

