



Java Fundamental | Bootcamp

OOP

Subjects : OOP1

1. Method
2. Instance Class
3. Encapsulation

Method

```
public class KelasDenganSatuMethod {  
    public void metodeSatu() {  
    }  
}
```

Method

```
package day11;

public class KelasDenganTigaMethod {

    public void metodeSatu() {
        // method1
    }

    public void metodeDua() {
        //method2
    }

    Run | Debug
    public static void main(String[] args) {
        //method3
    }

    public void metodeEmpat() {
        //mothod4
    }
}
```

Method

```
public class Ikan {  
  
    public void berenang() {  
  
    }  
  
    public void bernafas() {  
  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
  
    }  
}
```

Method Modifiers

Method modifiers merupakan tipe dari method yang kita buat.

Tipe method ini dibagi menjadi 3 yakni :

- **Public**

Tipe method yang bisa diakses di class manapun, asalkan masih dalam satu project yang sama.

- **Protected**

Tipe method yang hanya bisa diakses oleh class lain yang terletak di package yang sama dengan method yang berada di class induk.

- **Private**

Tipe method yang hanya bisa diakses didalam class itu sendiri,

Instance

```
public class Parent {  
  
    String nama;  
    int usia;  
  
    public void printParent() {  
        System.out.println("Nama      : "+nama+"\nUsia      : "+usia);  
    }  
}
```

Run | Debug

```
public static void main(String[] args) {
```

Return Constructor

```
    Parent objectParent = new Parent();
```

```
    objectParent.nama = "Supriyanto";
```

```
    objectParent.usia = 55;
```

```
    objectParent.printParent();
```

```
}
```

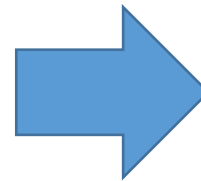
Instance Variable & Method

```
public class Parent {  
  
    String nama; —————> Instance Fields / Variable  
    int usia;  
  
    public Parent(String nama, int usia) {  
        this.nama = nama;  
        this.usia = usia;  
    }  
  
    public void printParent() {  
        System.out.println("Nama      : "+nama+"\nUsia      : "+usia); —————> Instance Method  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
  
        Parent objectParent = new Parent("Supriyanto", 55);  
        objectParent.printParent(); ←—————  
    }  
}
```

Instance

- Memanggil suatu class

```
public class InstanceClassDasar {  
  
    public void metodeSatu() {  
  
    }  
  
    public void metodeDua() {  
  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
  
    }  
}
```



```
public class InstanceClassDasar {  
  
    public void metodeSatu() {  
  
    }  
  
    public void metodeDua() {  
  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        ins  
    }  
}
```

InstanceClassDasar - day11 day11.InstanceClassDasar
InstantiationError - java.lang
InstantiationException - java.lang
Insert128Op - org.graalvm.compiler.lir.amd64.vect...
InsertBreakAction - javax.swing.text.DefaultEdito...
InsertContentAction - javax.swing.text.DefaultEdi...
InsertHTMLTextAction - javax.swing.text.html.HTML...
InsertTabAction - javax.swing.text.DefaultEditorK...
InsetsUIResource - javax.swing.plaf
InsList - jdk.internal.org.objectweb.asm.tree
InsNode - jdk.internal.org.objectweb.asm.tree

Instance

- Icd itu nama variable singkatan dari InstanceClassDasar (Variable Object)
- Nama variable boleh apa saja, cuma programmer senior biasa singkat

```
public class InstanceClassDasar {  
  
    public void metodeSatu() {  
  
    }  
  
    public void metodeDua() {  
  
    }  
  
}
```

Run | Debug

```
public static void main(String[] args) {
```

```
    InstanceClassDasar icd = new ins
```

Constructor

```
InstanceClassDasa... day11.InstanceClassDasar.Instance...  
InsetsUIResource(int top, int left, int bottom, i...  
InsnList()  
InsnNode(int opcode)  
InstalledCode(String name)  
InstanceAlreadyExistsException()  
InstanceAlreadyExistsException(String message)
```

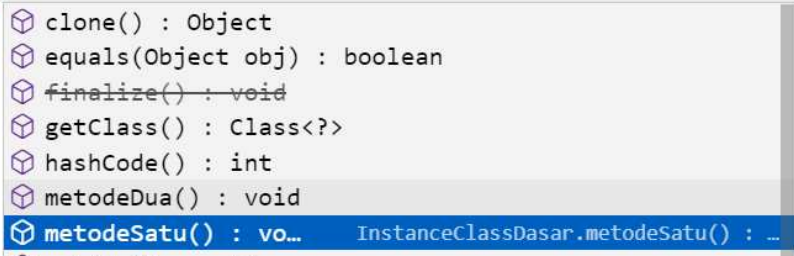
Instance

```
public class InstanceClassDasar {  
  
    public void metodeSatu() {  
  
    }  
  
    public void metodeDua() {  
  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        InstanceClassDasar icd = new InstanceClassDasar();  
    }  
}
```

Instance - Method

- Variable icd adalah sebuah instance
- Var icd dapat memanggil 2 method lainnya selain method dirinya sendiri yaitu method main

```
public class InstanceClassDasar {  
    public void metodeSatu() {  
    }  
    public void metodeDua() {  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        InstanceClassDasar icd = new InstanceClassDasar();  
        icd.  
    }  
}
```



Instance - Method

```
public class InstanceClassMethod {  
  
    public void metodeSatu() {  
        InstanceClassMethod icm = new InstanceClassMethod(); // instance class  
        icm.metodeDua(); // var icm memanggil metodeDua di metodeSatu  
    }  
  
    public void metodeDua() {  
  
    }  
  
}
```

Instance - Method

- Jelaskan mengapa outputnya seperti itu?

```
public class IkanDenganInstanceMethod {  
  
    Run | Debug  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        IkanDenganInstanceMethod idim = new IkanDenganInstanceMethod();  
        // instance class telah jadi  
        idim.berenang();  
        idim.bernafas();  
    }  
  
    public void berenang() {  
        System.out.println("berenang di air");  
    }  
  
    public void bernafas() {  
        System.out.println("bernafas dengan insang");  
    }  
}
```

berenang di air
bernafas dengan insang

Instance - Variable

```
public class IkanDenganInstanceVariable {  
  
    int jarak = 100;  
  
    Run | Debug  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        IkanDenganInstanceVariable idiv = new IkanDenganInstanceVariable();  
        // instance class telah jadi  
        int x = idiv.jarak; // nilai var x diambil dari idiv.jarak  
  
        System.out.println(x);  
  
        idiv.berenang(); // panggil method berenang  
    }  
  
    public void berenang() {  
        int tempuh = jarak;  
        // nilai var tempuh diambil dari varGlobal jarak  
        System.out.println("berenang di air "+tempuh);  
    }  
  
    // itulah mengapa instance class utk variable hanya berlaku utk var global  
    // Instance class utk var jarang dilakukan karena var global dari lgsg dipanggil  
}
```

Instance – Other Class

- Buat satu kelas bernama KelasSepupu, dengan satu method motor isinya cetak Motor Bebek

```
public class KelasSepupu {  
    public void motor() {  
        System.out.println("Motor Bebek");  
    }  
}
```

Instance – Other Class

- KelasSepupu di-*instance* (dipanggil) di KelasAku
- Maka KelasAku bisa cetak Motor Bebek karena dapat menggunakan method motor di kelas KelasSepupu

```
public class KelasAku {  
    Run | Debug  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        KelasSepupu ks = new KelasSepupu(); // instance class KelasSepupu  
        ks.motor();  
        ks.mobil();  
        System.out.println();  
        Adik ad = new Adik();  
        ad.nangis();  
    }  
}
```

Encapsulation – Same Package – Same Class

```
public class Saya {  
    Run | Debug  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Saya s = new Saya();  
        s.nama();  
        s.kamera();  
        s.handphone();  
    }  
  
    public void nama() {  
        System.out.println("Nama Saya");  
    }  
  
    // methodnya bisa dipanggil dalam class yang sama,  
    // bisa juga dipanggil dalam class lain yang berada di package yang sama  
    protected void kamera() {  
        System.out.println("Kamera Saya");  
    }  
  
    private void handphone() { // method yg bisa digunakan hanya dalam class itu sendiri  
        System.out.println("Handphone Saya");  
    }  
}
```

Encapsulation – Same Package – Other Class

- Jelaskan kenapa tidak ada method handphone?

```
public class Adik {  
    Run | Debug  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Saya s = new Saya();  
        s.nama();  
        s.kamera();  
    }  
    public void nangis() {  
        System.out.println("Teriak tidak jelas");  
    }  
}
```

Nama Saya
Kamera Saya

Encapsulation – Other Package – Other Class

- Jelaskan kenapa hanya ada method nama?

```
package day11.kerabat;
```

```
import day11.keluarga.Saya;       Nama Saya
```

```
public class Paman {  
    |  
    | Run | Debug  
    | public static void main(String[] args) {  
    |     | // TODO Auto-generated method stub  
    |     | Saya s = new Saya();  
    |     | s.nama();  
    |     |  
    |     | }  
    | }  
}
```