

# **TUGAS MINGGU 7**

## **Menerapkan Inheritance, Polymorphysm, dan Abstract Class**



**GOLONGAN A**

**Disusun oleh :**

**Nama : Noga Muktiwati**

**NIM : E41200415**

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI POLITEKNIK  
NEGERI JEMBER 2020**

## Tugas Minggu 7

### A. Dasar teori + Praktikum BKPM minggu 7

#### a. Pewarisan

##### ❖ Konsep pewarisan :

Pewarisan : proses subclass diturunkan dari superclass. Sehingga mendapatkan ciri atau sifat dari class tersebut.

##### ❖ Sourcode konsep :

```
6 package pewarisan;
7 /**
8  *
9  * @author NOGA MUKTIWATI
10 */
11 public class A {
12     int x;
13     int y;
14     void TampilkanNilaiXY() {
15         System.out.println ("Nilai X:" + x + "      Y:" + y);
16     }
17 }
18
19 * @author NOGA MUKTIWATI
20 */
21 public class B extends A { //Mendeklarasikan class B yang diturunkan
22     int z; //dari Class A
23     void TampilkanJumlah () {
24         System.out.println ("Jumlah : " + (x+y+z));
25         //subclass dapat mengakses member dari superclass
26     }
27 }
28
29 * @author NOGA MUKTIWATI
30 */
31 public class Demo {
32     public static void main (String[] args) {
33         A superClass = new A ();
34         B subClass = new B ();
35
36         System.out.println ("Super Class: ");
37         superClass.x = 1;
38         superClass.y = 2;
39         superClass.TampilkanNilaiXY();
40
41         System.out.println ("Sub Class: ");
42         subClass.x = 2;
43         subClass.y = 4;
44         subClass.TampilkanNilaiXY();
45
46         //member tambahan yang hanya ada pada subClass
47         subClass.z = 8;
48         subClass.TampilkanJumlah();
49     }
50 }
```

##### ❖ Output :

```
Output - Workshop_sistem_informasi_minggu7 (run) #2
run:
Super Class:
Nilai X:1      Y:2
Sub Class:
Nilai X:2      Y:4
Jumlah : 14
BUILD SUCCESSFUL (total time: 0 seconds)
```

### Manfaat Pewarisan :

Class B punya semua yg dimiliki class A. Oleh karena itu pada class B cukup ditambahkan bagian yang memang merupakan spesifik milik dari class B itu sendiri. Dengan demikian prinsip “reuse of code” telah diterapkan walaupun dalam bentuk yang paling sederhana.

### ❖ Source code praktikum :

```
package pewarisan2;

/**
 *
 * @author Noga Muktiwati
 */
public class orang {
    private String nama;
    private double tinggi;
    private double berat;
    public orang (String nama, double tinggi, double berat){
        this.nama = nama ;
        this.tinggi = tinggi;
        this.berat = berat;
    }
    public String toString() {
        return ("Nama: " + nama + "Tinggi: " + tinggi + "Berat: " + berat);
    }
}
```

```
package pewarisan2;

/**
 *
 * @author NOGA MUKTIWATI
 */
public class Pelajar extends orang {
    private String nim;
    private String asalSekolah;
    private double nilai; //range 0-30
    public Pelajar (String nama, double tinggi, double berat, String nim, String asalSekolah, double nilai){
        super (nama,tinggi, berat);
        this.nim = nim;
        this.asalSekolah = asalSekolah;
        this.nilai = nilai;
    }
    public String toString() {
        return (super.toString()+ "NIM :"+ nim + "Asal Sekolah: " + asalSekolah+ "Nilai: "+ nilai);
    }
}
```

```
package pewarisan2;

/**
 *
 * @author NOGA MUKTIWATI
 */
public class Test {
    public static void main (String []args) {
        Pelajar mahasiswa = new Pelajar ("NOGA", 152, 45, "E4100415", "Polije",99);
        System.out.println (mahasiswa.toString());
    }
}
```

### ❖ Output :

```
Output - Workshop_sistem_informasi_minggu7 (run)

run:
Nama: NOGATinggi: 152.0Berat: 45.0NIM :E4100415Asal Sekolah: PolijeNilai: 99.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

## b. Eksekusi Konstruktor

Karena subclass memiliki semua yang dimiliki oleh superclass, maka subclass juga memiliki konstruktor yang dimiliki oleh superclass-nya.

```
6 package konstruktor;
7 /**
8  *
9  * @author NOGA MUKTIWATI
10 */
11 public class A {
12     A(){
13         System.out.println("Konstruktor class A dieksekusi...");
14     }
15 }
```

```
6 package konstruktor;
7 /**
8  *
9  * @author NOGA MUKTIWATI
10 */
11 public class B extends A {
12     B(){
13         System.out.println ("Konstruktor class B dieksekusi...");
14     }
15 }
```

```
6 package konstruktor;
7 /**
8  *
9  * @author NOGA MUKTIWATI
10 */
11 public class C extends B {
12     C(){
13         System.out.println ("Konstruktor class C dieksekusi...");
14     }
15 }
```

```
6 package konstruktor;
7 /**
8  *
9  * @author noga muktiwati
10 */
11 public class Demo {
12     public static void main (String [] args){
13         C subClass = new C();
14     }
15 }
```

Konstruktor dieksekusi dari konstruktor milik superclass milik superclass tertinggi, baru kemudian yang terakhir adalah konstruktor dari class itu sendiri.

### ❖ Output :

```
Output - Workshop_sistem_informasi_minggu7 (run)
run:
Konstruktor class A dieksekusi...
Konstruktor class B dieksekusi...
Konstruktor class C dieksekusi...
BUILD SUCCESSFUL (total time: 0 seconds)
```

### c. Method Overriding

Jika dalam suatu subclass Anda mendefinisikan sebuah method yang sama dengan yang dimiliki oleh superclass, maka method yang Anda buat dalam subclass tersebut dikatakan meng-override superclass-nya. Sehingga jika Anda mencoba memanggil method tersebut dari instance subclass yang Anda buat, maka method milik subclass-lah yang akan dipanggil, bukan lagi method milik super-class.

#### ❖ Source Code:

```
6 package Overriding;
7 /**
8  *
9  * @author NOGA MUKTIWATI
10 */
11 public class A {
12     public void tampilkanKeLayar () {
13         System.out.println ("MOTHOD CLASS A dipanggil");
14     }
15 }
16
17 package Overriding;
18 /**
19  *
20  * @author NOGA MUKTIWATI
21 */
22 public class B extends A {
23     public void tampilkanKeLayar () {
24         super.tampilkanKeLayar();
25
26         System.out.println (" MOTHOD CLASS B dipanggil");
27     }
28 }
29
30 package Overriding;
31 /**
32  *
33  * @author NOGA MUKTIWATI
34 */
35 public class Demo {
36     public static void main (String[] args) {
37         B subOb = new B ();
38         subOb.tampilkanKeLayar();
39     }
40 }
```

#### ❖ Output :

```
Output - Workshop_sistem_informasi_minggu7 (run)
run:
MOTHOD CLASS A dipanggil
MOTHOD CLASS B dipanggil
BUILD SUCCESSFUL (total time: 0 seconds)
```

Class B merupakan turunan dari class A, meng-override method `tampilkanKeLayar()`. Sehingga pada waktu kita memanggil method tersebut dari variabel instance class B (variabel `subOb`), yang terpanggil adalah method yang sama yang ada di class B. Sebenarnya Anda tetap dapat memanggil method milik superclass yang telah di-override tersebut dari dalam subclass-nya dengan menggunakan keyword `super`, dimana penggunaannya mirip jika Anda hendak mengakses property milik dari superclass yang diakses dari subclass (lihat penjelasan penggunaan keyword `super` yang telah dijelaskan sebelumnya)

#### d. Abstract Class

Class dimana tidak semua method yang dideklarasikan di dalamnya memiliki implementasi. Class ini (**superclass**) sebagai dasar digunakan subclass, di mana implementasi dari method tidak dibuat di dalam superclassnya tersebut, harus diimplementasikan oleh masing – masing. Dengan demikian, superclass dapat pastikan memiliki method yg telah ditentukan sebelumnya. Class demikian ini disebut class abstrak.

Aturan class abstrak:

- Method yg tidak memiliki implementasi disuatu class, dideklarasikan sebagai abstrak dengan menggunakan keyword **abstract**.
- Class abstrak harus diturunkan terlebih dahulu, agar dapat membuat instance dari class abstrak di mana class hasil turunannya tersebut harus meng-override semua method abstrak dari superclass-nya dan membuat implementasinya (sehingga method tersebut bukan lagi merupakan method abstrak) baru kemudian subclass ini dapat dibuat instance-nya.
- Jika ada satu saja method (baik dari superclass-nya sebagai hasil dari turunan maupun yang dideklarasikan di dalam class itu sendiri) yg tidak di -implementasikan, maka class tersebut juga harus dideklarasikan sebagai abstrak menggunakan keyword **abstract**.

#### ❖ Sourcode

```
6 package Abstract;
7 /**
8  *
9  * @author NOGA MUKTIWATI
10 */
11 public abstract class A {
12     abstract public void method2();
13
14     //method konkrit yang memiliki implementasi
15     public void method1(){
16         System.out.println ("Method konkrit dari kelas A");
17     }
18 }
19
20 public class B extends A {
21     public void method2(){ /// method abstract di override dan dibuat
22         System.out.println ("Method konkrit dalam kelas B");
23     }
24 }
25
26 public class demo {
27     public static void main (String [] args){
28         B ob = new B ();
29         ob.method1();
30         ob.method2();
31     }
32 }
```

#### ❖ Output

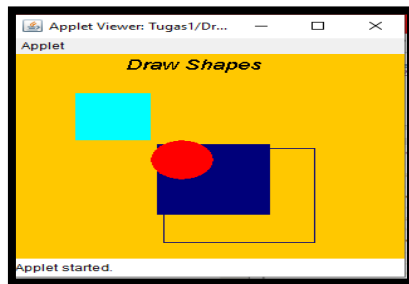
```
Output - Workshop_sistem_informasi_minggu7 (run)
run:
Method konkrit dari kelas A
Method konkrit dalam kelas B
BUILD SUCCESSFUL (total time: 0 seconds)
```

## B. Tugas BKPM Minggu 7

1. Ketikkan code program berikut dan jelaskan outputnya.

```
7 import java.awt.*;
8 import java.applet.*;
9 /**
10  *
11  * @author NOGA MUKTIWATI
12  */
13 public class DrawShapes extends Applet {
14     Font font ;
15     Color redColor;
16     Color blueColor;
17     Color backgroundColor;
18     public void init () {
19         //the font is arial, size 18 and is italicized
20         font = new Font ("Arial", Font.ITALIC, 18);
21
22         //some color are predefined in the Color class
23         redColor = Color.red;
24         backgroundColor = Color.orange;
25
26         //Color can be created using Red, Green, Blue values
27         blueColor = new Color (0,0,122);
28
29         //set the background Color of the applet
30         setBackground (backgroundColor);
31
32     public void stop() {
33     }
34     /**
35     * this method paints the shapes to the screen
36     */
37     public void paint (Graphics graph) {
38         //Set font
39         graph.setFont(font);
40         //create a title
41         graph.drawString("Draw Shapes", 90, 20);
42         //Set the color for the first shape
43         graph.setColor (blueColor);
44         //Draw a rectangle using drawRect (int x, int y, int width, int height)
45         graph.drawRect (120, 120, 120, 120);
46         //This will fill a rectangle
47         graph.fillRect (115,115,90,90);
48         //set the color for the next shape
49         graph.setColor (redColor);
50         //Draw a circle using drawArc (int x, int y, int width, int height, int startAngle, int arcAngle)
51         graph.fillArc (110, 110, 50, 50, 0, 360);
52         //set color for next shape
53         graph.setColor (Color.CYAN);
54         //draw another rectangle
55         graph.drawRect (50, 50, 50, 50);
56         //This will fill a rectangle
57         graph.fillRect(50,50,60,60);
58     }
59 }
```

**Output :**



## Penjelasan Output:

Java applet adalah sebuah program kecil yang ditulis dengan menggunakan bahasa pemrograman Java, yang diakses melalui halaman Web dan dapat didownload ke dalam mesin klien yang kemudian menjalankannya di dalam jendela penjelajah web. Java applet dapat secara dinamis menambahkan beberapa fungsi kepada halaman-halaman Web yang bersifat statis.

Tentang import java.awt. Grafical User Interface (GUI) adalah salah satu kemampuan Java dalam mendukung dan manajemen antarmuka berbasis grafis. Tampilan grafis yang akan ditampilkan terhubung dengan program serta tempat penyimpanan data. Elemen dasar di Java untuk penciptaan tampilan berbasis grafis adalah dua paket yaitu AWT dan Swing

### 1. setBackground (backgroundColor):

Digunakan untuk pemberian warna background pada Applet.

### 2. graph.setFont (font) :

Digunakan untuk memanggil jenis font, ukuran font, Italic (miring) yang sudah di set sebelumnya.

### 3. graph.drawString:

Digunakan untuk membuat sebuah judul pada Applet.

### 4. graph.setColor :

Pemberian warna terhadap bentuk yang digunakan.

### 5. graph.drawRect :

Menggambar persegi Panjang dan menentukan x, y, lebar, dan tinggi.

### 6. graph.fillRect :

Mengisi Warna pada rectangle.

### 7. graph.fillArc :

Digunakan untuk membuat lingkaran dan menentukan x,y,lebar dan tinggi.

## 2. Ketikkan code program berikut dan jelaskan outputnya.

### a. Class A

```
9 | * @author NOGA MUKTIWATI
10 | */
11 | public class A {
12 |     void callthis () {
13 |         System.out.println ("Inside Class A's Method !");
14 |     }
15 | }
```

### b. Class B

```
12 | public class B extends A {
13 |     void callthis () {
14 |         System.out.println ("Inside Class B's Method!");
15 |     }
16 | }
```

### c. Class C

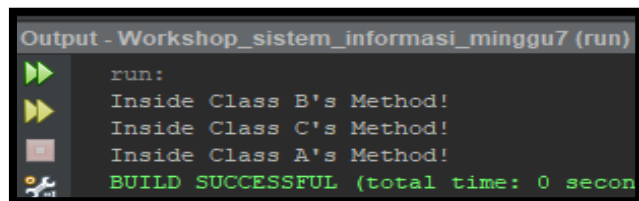
```
12 | public class C extends A {
13 |     void callthis () {
14 |         System.out.println ("Inside Class C's Method!");
15 |     }
16 | }
```



#### d. Class Dynamic Dispatch

```
12 public class DynamicDistpach {
13     public static void main (String args []) {
14         A a = new A ();
15         B b = new B ();
16         C c = new C ();
17         A ref;
18
19         ref = b ;
20         ref.callthis () ;
21
22         ref = c ;
23         ref.callthis () ;
24
25         ref = a ;
26         ref.callthis () ;
27     }
28 }
```

#### Output :



```
Output - Workshop_sistem_informasi_minggu7 (run)
run:
Inside Class B's Method!
Inside Class C's Method!
Inside Class A's Method!
BUILD SUCCESSFUL (total time: 0 secon
```

#### Penjelasan :

Class A merupakan induk ( Superclass) dari class B dan class C. Yang mana hal tersebut membuat kelas B dan C mempunyai property dan method yang juga dimiliki oleh kelas A. Disini, terdapat method **callThis()** yang merupakan method yang berasal dari kelas A. Namun method tersebut diwariskan dan diberi perintah yang berbeda pada tiap kelas turunannya. Lalu pada kelas DyanamicDispatch terjadi instansiasi kelas A, B, dan C. Setelah itu kelas A mewakilkan kelasnya pada variable ref, yang mana nantinya variable tersebut bisa memanggil method yang ada.

#### ❖ Extends

Mempunyai fungsi untuk memanggil fungsi dari class lain( class induk) Sehingga kita tidak perlu lagi membuat script yang sama pada class yang akan kita buat dengan class yang kita buat sebelumnya.

#### ❖ Void

Mempunyai fungsi untuk tidak mengembalikan nilai apapun ketika main method dijalankan.

#### ❖ DynamicDispatch

Adalah Proses di mana Java dapat menentukan metode mana yang akan dipanggil ketika metode telah diganti.

3. Ketikkan code program berikut dan jelaskan outputnya

a. **CLASS ANIMAL**

```
6 package TugasC;
7 /**
8  *
9  * @author NOGA MUKTIWATI
10 */
11 public class Animal {
12     public void makeNoise () {
13         System.out.println("talk");
14     }
15 }
```

b. **CLASS DOG**

```
12 public class Dog extends Animal {
13     public void makeNoise () {
14         System.out.println ("BARK !");
15     }
16 }
```

c. **DEMO**

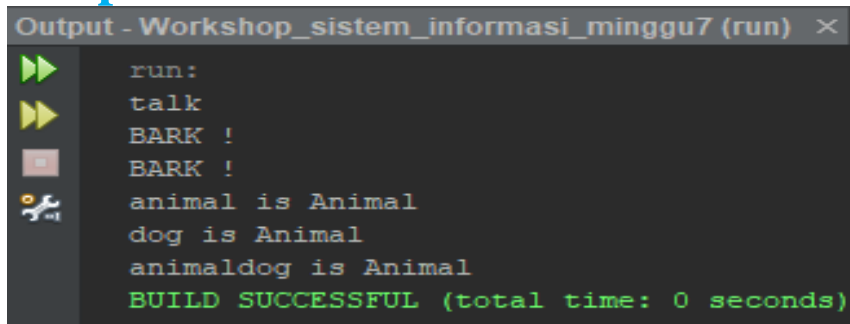
```
package TugasC;
/**
 *
 * @author NOGA MUKTIWATI
 */
public class Demo {
    public static void main (String args[]) {
        Animal animal = new Animal ();
        animal.makeNoise();

        Dog dog = new Dog ();
        dog.makeNoise();

        Animal animaldog = new Dog ();
        animaldog.makeNoise ();

        if (animal instanceof Animal)
            System.out.println ("animal is Animal");
        if (dog instanceof Animal)
            System.out.println ("dog is Animal");
        if (animaldog instanceof Animal)
            System.out.println ("animaldog is Animal");
        if (animal instanceof Dog)
            System.out.println ("animal is Dog");
    }
}
```

## Output :



```
Output - Workshop_sistem_informasi_minggu7 (run) ×
run:
talk
BARK !
BARK !
animal is Animal
dog is Animal
animaldog is Animal
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Penjelasan :

1. Kelas Animal sebagai superclass yang punya anak (subclass) Class Dog.
2. Kelas Animal punya method makeNoise() mewariskan method tersebut pada kelas Dog. Sehingga method makeNoise() pada kelas Dog, menunjukkan output “Bark”.
3. Kemudian di dalam kelas Animal, dicoba instansiasi kelas Animal serta Dog, dimana variable animal ialah instance dari kelas Animal, serta variable dog ialah instance dari kelas Dog, pula animalDog ialah instance dari kelas Animal serta Dog.

# TUGAS TAMBAHAN via WhatsApp

## 1. Source Code

### a. SuperClass (Televisi)

```
package Tugas_Tambahan;
/**
 *
 * @author NOGA MUKTIWATI
 */
public class Televisi {
    public String deks;
    public int jumlahChannel;
    private String [] channels = null;
    private int channelAktif;
    private int volume;

    public Televisi (String deks, int jumlahChannel){
        this.deks = deks ;
        this.jumlahChannel = jumlahChannel;
    }
    public String getDesk () {
        return deks;
    }
    public int getJumlahChannel () {
        return jumlahChannel;
    }
    public String [] getChannels () {
        if (channels == null){
            System.out.println ("Belum ada Channel yang diatur!");
        }
        return channels;
    }

    public void setChannels (String [] channels) {
        if (channels.length > jumlahChannel){
            System.out.println ("Maaf TV ini hanya dapat menyimpan"+
                jumlahChannel + "Channel");
        }
        else {
            this.channels = channels;
            System.out.println ("Informasi Channel berhasil di update !");
        }
    }
    public int getChannelAktif () {
        return channelAktif;
    }
    public void setChannelAktif (int channelAktif) {
        if (channelAktif > channels.length){
            System.out.println ("Channel yang dicari belum diatur");
            return;
        }
        this.channelAktif = channelAktif;
        System.out.println ("Pindah Channel ke : " + channels[channelAktif]);
    }
    public int getVolume () {
        return volume;
    }
    public void setVolume (int volume) {
        this.volume = volume;
        System.out.println ("intensitas volume sekarang:" + volume);
    }
}
```

## b. Class User

```
package Tugas_Tambahan;

/**
 *
 * @author NOGA MUKTIWATI
 */
public class UserTelevisi {

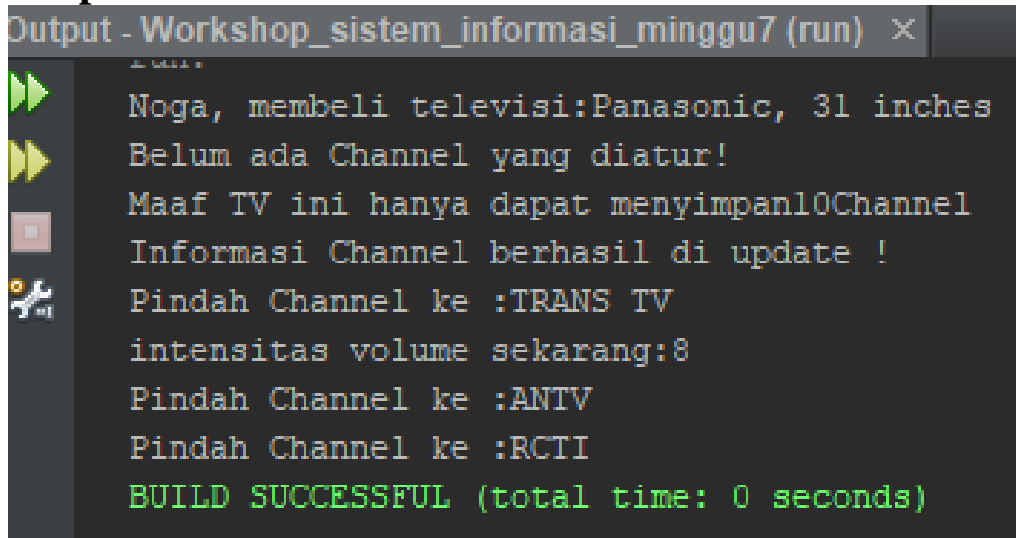
    public static void main (String []args){
        Televisi tv = new Televisi ("Panasonic, 31 inches", 10);

        String [] semuaChannel = {"METRO TV","RCTI","NET TV","TRANS TV",
            "TRANS 7","SCTV", "INDOSIAR", "ANTV", "TVRI", "MNC TV", "JTV"};
        String[] channelFavorit = {"MNC", "TRANS 7", "TRANS TV",
            "SCTV", "RCTI","ANTV"};
        System.out.println ("Noga, membeli televisi:" + tv.getDesk());

        tv.getChannels();
        tv.setChannels(semuaChannel);
        tv.setChannels(channelFavorit);

        tv.setChannelAktif(2);
        tv.setVolume(8);
        tv.setChannelAktif(5);
        tv.setChannelAktif(4);
    }
}
```

## c. Output



```
Output - Workshop_sistem_informasi_minggu7 (run) x
Noga, membeli televisi:Panasonic, 31 inches
Belum ada Channel yang diatur!
Maaf TV ini hanya dapat menyimpan10Channel
Informasi Channel berhasil di update !
Pindah Channel ke :TRANS TV
intensitas volume sekarang:8
Pindah Channel ke :ANTV
Pindah Channel ke :RCTI
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan :

- 1) Deskripsi digunakan untuk menyimpan nama/ deskripsi dari televisi.  
Property ini hanya dapat diset saat pembuatan saja, dapat dibaca namun tidak dapat diubah setelahnya.

- 2) JumlahChannel digunakan untuk menyimpan banyaknya channel yang dapat disimpan oleh televisi. Property ini hanya dapat diset saat pembuatan saja, dapat dibaca namun tidak dapat diubah setelahnya.
- 3) Channels digunakan untuk menyimpan data-data channel stasiun televisi yang sudah diset. Property ini dapat dibaca dan diubah kapan saja.
- 4) ChannelAktif digunakan untuk menyimpan nomor channel yang sedang aktif saat ini. Property ini dapat dibaca dan diubah kapan saja.
- 5) Volume digunakan untuk menyimpan informasi besarnya volume televisi yang sedang aktif saat ini. Property ini dapat dibaca dan diubah kapan saja.
- 6) tv.setChannelAktif(4) digunakan untuk berpindah ke channel no 10 dan melihat apakah channel tersebut aktif atau tidak, jika tidak maka no channel akan berpindah
- 7) tv.setChannelAktif(5) digunakan untuk berpindah ke channel no 10 dan melihat apakah channel tersebut aktif atau tidak, jika tidak maka no channel akan berpindah.
- 8) tv.getDesk() digunakan untuk memanggil output deskripsi dari televisi
- 9) tv.getChannels() digunakan untuk memanggil output Channel TV
- 10) tv.setChannels(semuaChannels) digunakan untuk memanggil output semua Channel yang telah dituliskan diatasnya
- 11) tv.setChannel(channelfavorite) digunakan untuk memanggil output semua Channel Favorite yang sudah dituliskan diatas
- 12) tv.setChannelAktif(2) digunakan untuk memanggil output apakah channel no 2 ini aktif
- 13) tv.setVolume(8) digunakan untuk mengatur volume televisi.

## 2. Source Code

### a. Class TV

```
6 package TV;
7 /**
8  *
9  * @author NOGA MUKTIWATI
10 */
11 public class TV {
12     String deskripsi = "";
13     int Tchannel = 0;
14     String[] channel = new String[Tchannel];
15     int CAktif = 0;
16     int vlm = 0;
17     String pembeli = "";
18
19     // constructor (Method Khusus)
20     TV(final String n, final String deskripsi, final int Tchannel) {
21         this.pembeli = n;
22         this.deskripsi = deskripsi;
23         this.Tchannel = Tchannel;
24     }
25
26     // set deskripsi
27     public void getdeskripsi() {
28         System.out.println(pembeli + " membeli " + deskripsi);
29     }
30
31     // set channel
32     public void setchannel(String[] channel) {
33         if (channel.length > this.Tchannel) {
34             System.out.println("Tv ini hanya dapat menyimpan " + this.Tchannel + " channel");
35         } else {
36             this.channel = channel;
37             System.out.println("Informasi channel anda berhasil diupdate!");
38         }
39     }
40
41     // channel aktif
42     public void getchannel() {
43         if (this.channel.length == 0) {
44             System.out.println("Belum ada channel yang diseting!");
45         } else {
46             String data = "";
47             for (String chnl : this.channel) {
48                 data += chnl + " ";
49             }
50             System.out.println("Channel yang ada : [ " + data + " ]");
51         }
52     }
53
54     // set channel aktif
55     public void setCAktif(int x) {
56         this.CAktif = x;
57         int data = x;
58         System.out.println("Pindah ke-channel : " + this.channel[x-1]);
59     }
60
61     // set volume
62     public void setvlm(int v) {
63         System.out.println("Volume sekarang : " + v);
64     }
65
66 }
67
68 class Use {
69     public static void main(String[] args) {
70
71         TV Noga= new TV("NOGA MYUKTIWATI", "TV jumbo THOSIBA, Warna Hitam", 11);
72         String[] channel = {"SCTV", "TransTV", "TVONE", "RCTI", "GlobalTV",
73                             "KompasTV", "MOPChanel", "JTV", "Net",
74                             "Indosiar", "ANTV"};
75
76         Noga.getdeskripsi();
77         Noga.getchannel();
78     }
79 }
```

```

        Noga.setchannel(channel);

        Noga.setCAktif(8);
        Noga.setvlm(9);
        Noga.setCAktif(11);
    }
}

```

## b. Class TVmodern

```

    * @author NOGA MUKTIWATI
    */
    public class TVmodern extends TV {
        public TVmodern (String n, String deskripsi, int TChannel) {
            super(n, deskripsi, TChannel);
        }

        static String TELETxt = "Teletxt";
        static String TV = "Televisi";

        public void setModusTampilan(String t) {
            this.TELETxt = t;
        }

        public void getModusTampilan() {
            System.out.println("Modus tampilan : " + TV);
        }

        public void setHlmnTeletxt(int t) {
            System.out.println("Berpindah ke-teletext " + t);
        }

        // CD

        String cd = "";
        public void setDiscTray(String d) {
            this.cd = d;
        }

        public void played() {
            if (cd.equals("")) {
                System.out.println("CD tidak ditemukan");
            } else {
                System.out.println("Memutar film " + cd);
            }
        }

        public static void main(String[] args) {

            TVmodern Noga = new TVmodern("NOGA MUKTIWATI", "TV Polytron, Warna Hitam", 11);

            String[] channel = {"SCTV", "TransTV", "TVONE", "RCTI", "GlobalTV",
                                "KompasTV", "MOPChanel", "JTV", "Net",
                                "Indosiar", "ANTV"};

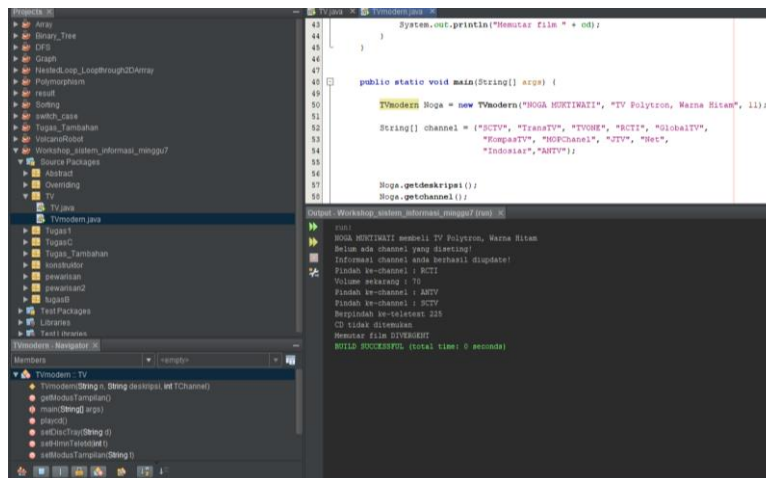
            Noga.getdeskripsi();
            Noga.getchannel();
            Noga.setchannel(channel);
            Noga.setCAktif(4);
            Noga.setvlm(70);
            Noga.setCAktif(11);
            Noga.setCAktif(1);
            Noga.setModusTampilan(TVmodern.TELETxt);
            Noga.setHlmnTeletxt(225);
            Noga.setModusTampilan(TVmodern.TV);
            Noga.playcd();
            Noga.setDiscTray("DIVERGENT");
            Noga.playcd();
        }
    }
}

```



## Output :

```
Output - Workshop_sistem_informasi_minggu7 (run) x
run:
NOGA MUKTIWATI membeli TV Polytron, Warna Hitam
Belum ada channel yang diseting!
Informasi channel anda berhasil diupdate!
Pindah ke-channel : RCTI
Volume sekarang : 70
Pindah ke-channel : ANTV
Pindah ke-channel : SCTV
Berpindah ke-teletext 225
CD tidak ditemukan
Memutar film DIVERGENT
BUILD SUCCESSFUL (total time: 0 seconds)
```



## Penjelasan :

- 1) Pada TelevisiM (superclass) kita melakukan inisialisasi semua method yang ingin digunakan beserta type datanya.
- 2) TelevisiModern adalah kelas turunan dari kelas induk (superClass) Televisi. Terdapat beberapa penambahan method yang tidak dimiliki oleh kelas induk tersebut untuk menjalankan televisi modern. Terdapat method: TelevisiModern (String desk, int jumlahChannels), setModusTampilan(String a), setHalamanTeletex(int teletext), setDiscTray(String discTray), getDiscTray(), playCD().
- 3) Static String TELETEXT = "Teletext" digunakan untuk menampung nilai statis dari variable teletext.
- 4) Public void setDiscTray(String discTray) digunakan untuk melakukan seting dimana nilai variable discTray = parameterdiscTray
- 5) Public void setHalamanTeleText(int teletext) digunakan agar program menampilkan berpindah ke halaman sesuai inputan dari parameter teletext.
- 6) Public String getHalamanTeleText() digunakan untuk mengembalikan nilai yang ada pada variable teletext
- 7) Public void setModusTampilan(String a) digunakan untuk mengatur modus tampilan yang nilainya didapat dari parameter

**THANK'S A LOT 😊**