

## MODUL PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

### INHERITANCE

#### Deskripsi Singkat

Praktikum pemrograman berorientasi objek adalah praktikum yang menggunakan bahasa Java sebagai bantuan dalam memahami konsep pemrograman berorientasi objek. Materi praktikum berisi teori, latihan dan soal pemrograman.

#### Tujuan

1. Memahami konsep inheritance.
2. Menggunakan konsep inheritance pada pemrograman berorientasi objek.
3. Memahami hak akses protected.
4. Memahami keyword super.

#### Prasyarat

Siswa telah melakukan praktikum 1-5.

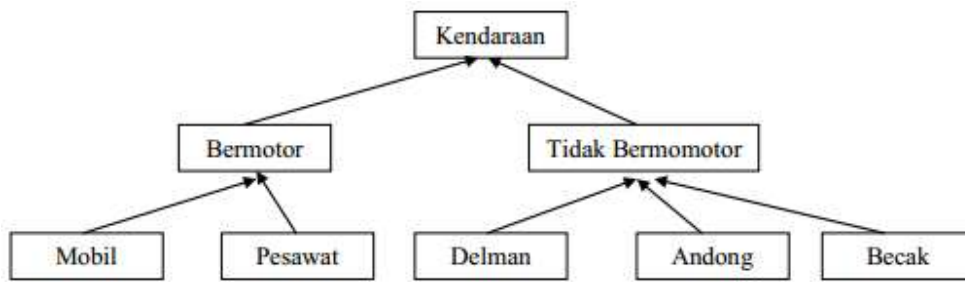
#### Materi 1 : Inheritance

Inheritance diterjemah menjadi pewarisan. Pada inheritance, satu kelas dapat lebih dispesifikan menjadi satu atau lebih sub-kelas. Sub-kelas akan mewarisi semua keadaan/atribut dan perilaku/method daripada super-kelasnya. Namun sub-kelas dapat memiliki keadaan/atribut dan perilaku/method lain yang spesifik kepada dirinya sendiri. Inheritance juga disebut sebagai relasi "**is-a**".

Agar sub-kelas mewarisi super-kelasnya digunakan kata kunci **extends**.

Contoh: **public class CD extends Item**

Java hanya memperkenankan adanya **single inheritance**. Konsep single inheritance hanya memperbolehkan suatu subclass mempunyai satu parent class. Dengan konsep single inheritance ini, masalah pewarisan akan dapat diamati dengan mudah. Konsep single inheritance bisa dilihat pada gambar di bawah. Pada gambar tersebut bisa dilihat bahwa tiap class pasti memiliki hanya satu parent class. Java mengizinkan suatu sub class untuk memiliki anak lagi. Hal ini disebut dengan konsep multi level inheritance. Contohnya class Kendaraan memiliki class anak yaitu Bermotor, dan class Bermotor memiliki class anak lagi yaitu Mobil dan Pesawat.



Dalam konsep dasar inheritance dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extends) parent class-nya. Contoh :

```

public class Pegawai {
    public String nama;
    public double gaji;
}
  
```

```

public class Manajer extends Pegawai {
    public String departemen;
}
  
```

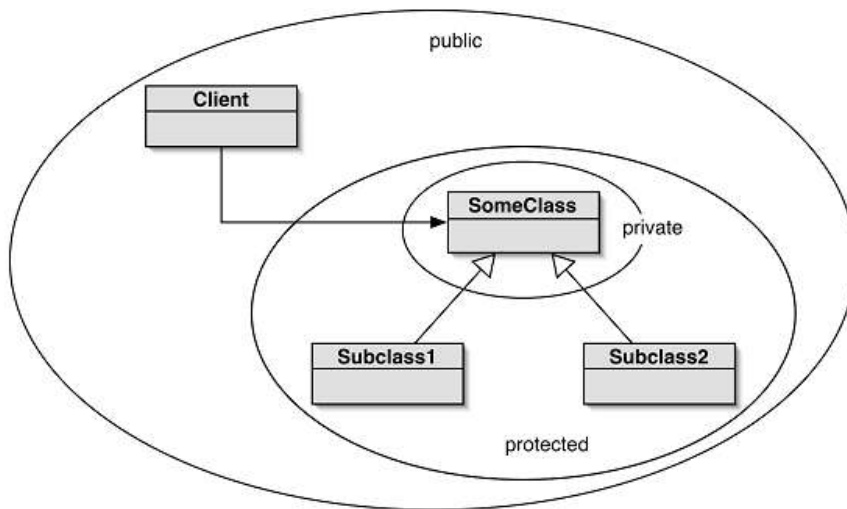
Pada saat class Manajer menurunkan atau memperluas (extends) class Pegawai, maka ia mewarisi semua data yang dimiliki oleh class Pegawai kecuali data tersebut memiliki hak akses private.

Kata kunci **super** digunakan untuk memanggil method yang ada pada super-kelas, baik itu method constructor maupun method yang lainnya.

## Materi 2 : Hak Akses Protected

Suatu parent class dapat tidak mewariskan datanya kepada sub-classnya jika hak akses yang digunakan adalah private. Modifier hak akses pada Java dapat dilihat pada tabel di bawah.

Modifier	class yang sama	package yang sama	subclass	class manapun
private	√			
default	√	√		
protected	√	√	√	
public	√	√	√	√



Modifier hak akses default akan digunakan saat suatu atribut/ instance variable tidak mendefinisikan hak akses.

```
public class Pegawai {
    String nama;
```

Pada konsep inheritance, kita mengenal hak akses baru yaitu protected. Atribut/ instance variable yang didefinisikan sebagai protected, maka atribut tersebut dapat diakses dan diubah nilainya pada sub-class.

```
public class Pegawai {
    private String nama;
    protected double gaji;
```

### Materi 3 : Keyword super

Dalam inheritance, data class yang diwariskan hanyalah variabel dan method saja. Sedangkan constructor parent class tidak diwariskan ke sub class. Namun constructor pada parent class dapat dipanggil dengan menggunakan kata kunci super(). Namun pemanggilan super constructor ini haruslah pada baris pertama. Jika tidak pada baris pertama maka akan muncul error.

Namun method selain dari method constructor dari induk class dapat dengan bebas digunakan. Dengan memanfaatkan super yang diikuti nama methodnya.

Jika ada class kosong seperti di bawah.

```
public class Parent {

}
```

Kemudian buat class Child.

Praktikum 6  
Inheritance

```
public class Child extends Parent {  
    int x;  
    public Child() {  
        x = 5;  
        super();  
    }  
}
```

Kompilasi class Child. Pasti akan muncul error. Berikutnya ubah menjadi:

```
public class Child extends Parent {  
    int x;  
    public Child() {  
        super();  
        x = 5;  
    }  
}
```

#### **Materi 4 : Method Constructor pada Subclass**

Method constructor tidak diturunkan ke sub class. Untuk membuktikan hal tersebut, buat class berikut.

```
public class Pegawai {  
    private String nama;  
    private double gaji;  
  
    public Pegawai(String nama, double gaji)  
    {  
        this.nama = nama;  
        this.gaji = gaji;  
    }  
  
    public void setNama(String nama)  
    {  
        this.nama = nama;  
    }  
  
    public String getNama()  
    {  
        return this.nama;  
    }  
  
    public void setGaji(double gaji)  
    {  
        this.gaji = gaji;  
    }  
  
    public double getGaji()  
    {  
        return this.gaji;  
    }  
}
```

```
}  
  
public class Manajer extends Pegawai {  
    private String departemen;  
  
    public Manajer(String nama, double gaji, String departemen)  
    {  
        Pegawai p = new Pegawai(nama, gaji);  
        this.departemen = departemen;  
    }  
  
    public void setDepartemen(String departemen)  
    {  
        this.departemen = departemen;  
    }  
  
    public String getDepartemen()  
    {  
        return this.departemen;  
    }  
}
```

Kompilasi class Manajer. Apakah ada error?

## LATIHAN 1

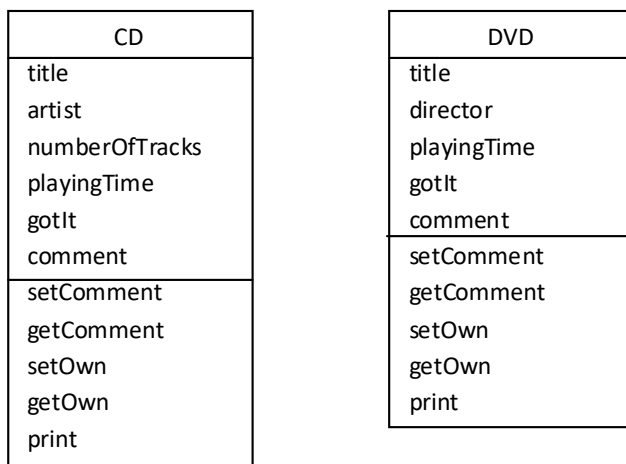
Buka software BlueJ, dan buka projek dome-v1 yang terletak di folder chapter08. Compile. Klik kanan dan pilih Open Editor untuk melihat source codenya.

Projek dome-v1 ini merupakan aplikasi sederhana untuk mendata koleksi CD dan DVD yang anda punya. Coba projek dome-v1 dan pahami cara kerjanya. Langkah-langkah:

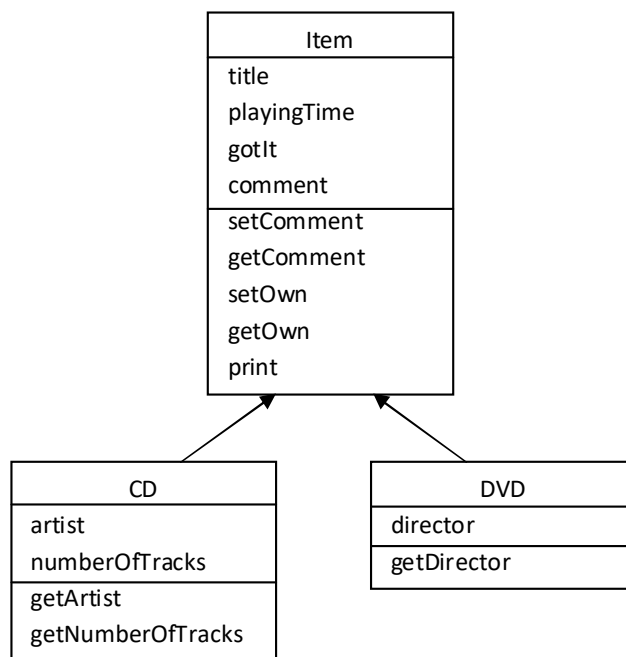
- Buatlah 2 atau 3 objek CD dan 2 atau 3 objek DVD.
- Buat objek Database.
- Masukkan semua objek CD ke dalam Database dengan memilih method addCD.
- Masukkan semua objek DVD ke dalam Database dengan memilih method addDVD.
- Kemudian list isi dari database.

Perhatikan dan pahami code pada kelas CD, DVD dan Database. Disini dapat dilihat kelas CD dan DVD hampir sama. Pada kelas Database, juga banyak terdapat hal yang berulang seperti pada method print.

Praktikum 6  
Inheritance



Jadi untuk mengatasi masalah duplikasi kode ini maka kita dapat menggunakan konsep inheritance. Kita kenalkan satu objek baru yang mengakomodir kesamaan antara objek CD dan DVD. Hal ini dapat dilihat pada diagram di bawah.



Buka software BlueJ, dan buka projek dome-v2 yang terletak di folder chapter08. Compile. Klik kanan dan pilih Open Editor untuk melihat source codenya.

Projek dome-v2 ini merupakan perbaikan dome-v1 dengan menggunakan inheritance. Coba projek dome-v2 dan pahami cara kerjanya. Langkah-langkah:

- Buatlah 2 atau 3 objek CD dan 2 atau 3 objek DVD.
- Buat objek Database.
- Masukkan semua objek CD dan DVD ke dalam Database dengan memilih method addItem.

- Kemudian list isi dari database.

Perhatikan dan pahami source code dari program. Kode yang berulang hilang dan program dapat dikembangkan lebih lanjut dengan mudah.

Lihat method constructor pada kelas CD:

```
public CD(String theTitle, String theArtist, int tracks, int time)
{
    super(theTitle, time);
    artist = theArtist;
    numberOfTracks = tracks;
}
```

Disini kata kunci **super** digunakan untuk memanggil method constructor dari super-kelas.

## LATIHAN 2

Buatlah class Pegawai seperti di bawah ini:

```
public class Pegawai {
    private String nama;
    private double gaji;

    public Pegawai(String nama, double gaji)
    {
        this.nama = nama;
        this.gaji = gaji;
    }

    public void setNama(String nama)
    {
        this.nama = nama;
    }

    public String getNama()
    {
        return this.nama;
    }

    public void setGaji(double gaji)
    {
        this.gaji = gaji;
    }

    public double getGaji()
    {
        return this.gaji;
    }
}
```

```
}
```

Kemudian buat class Manajer yang merupakan turunan class Pegawai.

```
public class Manajer extends Pegawai {  
    private String departemen;  
  
    public Manajer(String nama, double gaji, String departemen)  
    {  
        super(nama, gaji);  
        this.departemen = departemen;  
    }  
  
    public void setDepartemen(String departemen)  
    {  
        this.departemen = departemen;  
    }  
  
    public String getDepartemen()  
    {  
        return this.departemen;  
    }  
}
```

Kompilasi program di atas.

Buat class TesPegawai yang berisi method main yang menggunakan class Pegawai dan Manajer.

## SOAL-SOAL

1. Kembangkan proyek dome-v2 menjadi dome-v3 dengan menambahkan item koleksi berupa VideoGame. VideoGame merupakan sub-kelas dari kelas Item. Spesifikasi kelas VideoGame.

VideoGame
numberOfPlayers
platform
getNumberOfPlayers
getPlatform

Buatlah kode untuk class VideoGame dengan menggunakan spesifikasi di atas.

2. Buka proyek lab-class yang terletak dalam folder chapter01. Tambahkan kelas Instructor ke dalam proyek tersebut. Untuk setiap lab-class terdiri dari satu instructor dan banyak students (tergantung kapasitas yang diset di awal). Kelas instructor mempunyai nama dan detil kontak. Karena student dan instructor sama-sama mempunyai nama, gunakan konsep inheritance dengan menambahkan satu super-kelas. Pindahkan method yang sesuai pada student ke super-kelasnya.