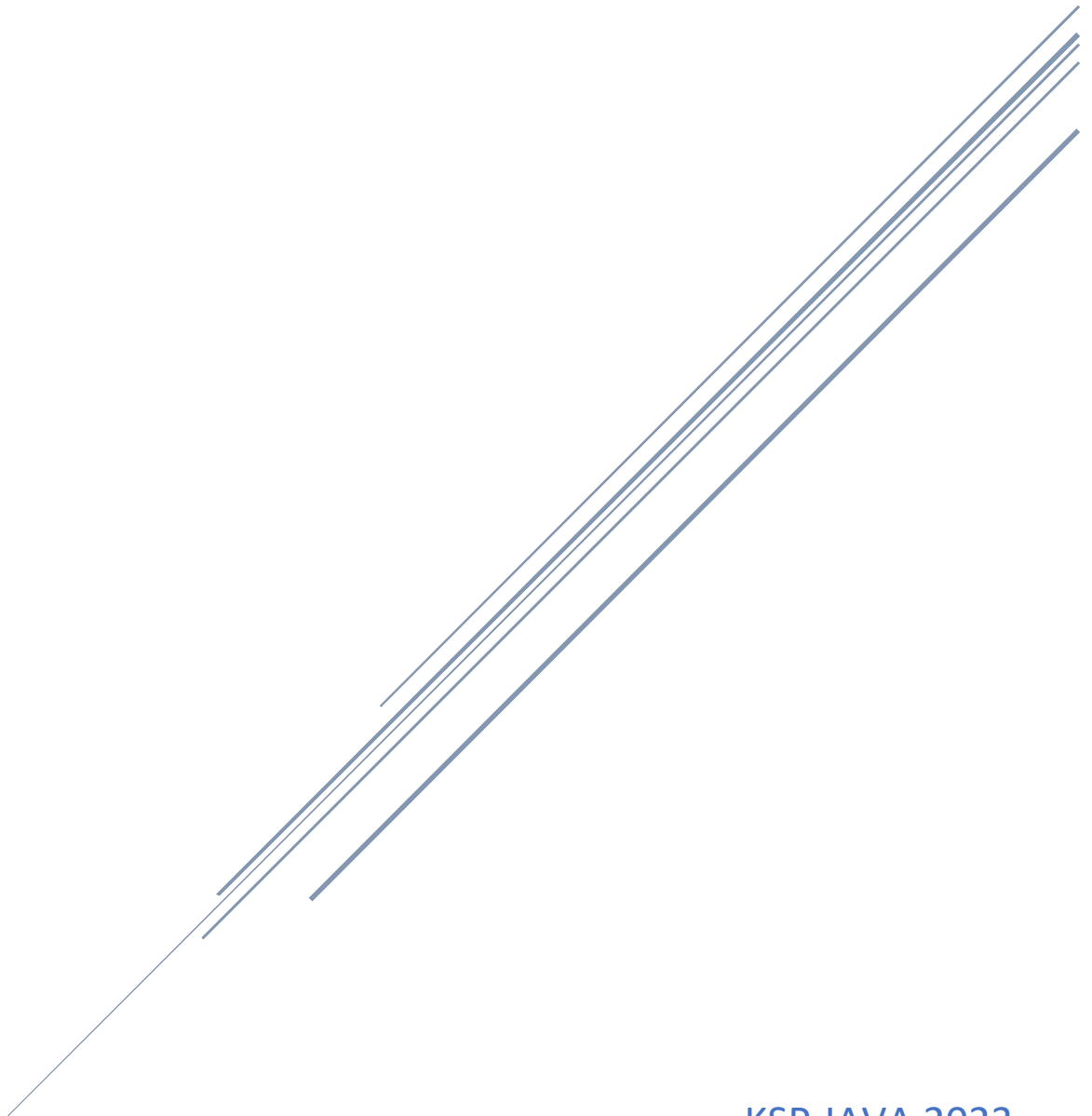


MODUL 5

Relasi <3



KSP JAVA 2023

RELASI KELAS

Dalam menggambarkan relasi kelas dalam PBO, dibutuhkan UML (Unified Modelling Language). UML merupakan spesifikasi pemodelan yang banyak digunakan untuk memodelkan struktur dan perilaku aplikasi. Memiliki jenis diagram yang dapat digunakan untuk memodelkan aplikasi, namun dalam modul ini akan dibahas kelas diagramnya saja yang merupakan UML yang digunakan untuk memodelkan kelas dalam PBO.

Jenis Jenis Relasi

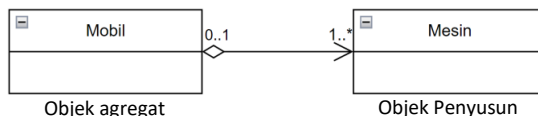
1. Pewarisan (inheritance)

Merupakan relasi turunan dimana sebuah kelas diciptakan berdasarkan kelas lainnya. Terdiri dari subclass dan superclass.

2. Agregasi (Aggregation)

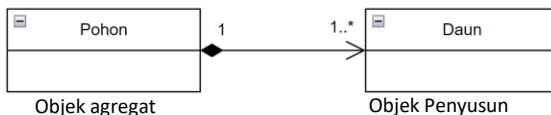
a. Agregasi

Sebuah objek tersusun atas objek lain. Relasi antar objek bersifat **lemah**. Objek penyusun **bisa ada** tanpa harus ada objek agregat. Misalnya Mesin akan tetap ada meskipun tidak ada mobil



b. Komposisi

Adalah sebuah varian dari agregasi. Komposisi adalah sebuah objek tersusun atas objek lain. Relasi antar objek bersifat **kuat**. Objek penyusun **tidak bisa ada** tanpa keberadaan objek komposit. Contohnya jika pohon dihancurkan maka daun akan ikut hancur karena daun hanya bisa hidup dari berdirinya pohon.



Dari gambar di atas yang membedakan agregasi dan komposisi adalah pada gambar diamondnya. Komposisi akan digambarkan dengan diamond dengan terarsir penuh sedangkan agregasi tidak diarsir sama sekali. Untuk membedakan agregasi dan komposisi bisa juga dilihat dari diagramnya, ketika multiplicity pada sisi wholenya adalah 1 maka relasinya adalah komposisi, jika sisi wholenya 0 - 1 maka disebut dengan agregasi, dari sisi code dapat dibedakan dengan:

Agregasi

```
public class Mobil {  
    private String merkMobil;  
    private double hargaMobil;  
    private Mesin mesin;  
  
    public Mobil(String merkMobil, double hargaMobil, Mesin mesin) {  
        this.merkMobil = merkMobil;  
        this.hargaMobil = hargaMobil;  
        this.mesin = mesin;  
    }  
  
    public String getMerkMobil() {  
        return merkMobil;  
    }  
  
    public void setMerkMobil(String merkMobil) {  
        this.merkMobil = merkMobil;  
    }  
}
```

Komposisi

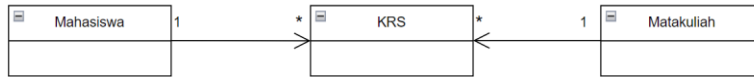
```
public class Pohon {  
    private String namaPohon;  
    private int tinggiPohon;  
    private Daun daun;  
  
    public Pohon(String namaPohon, int tinggiPohon, String tulangDaun, String warnaDaun) {  
        this.namaPohon = namaPohon;  
        this.tinggiPohon = tinggiPohon;  
        this.daun = new Daun(tulangDaun, warnaDaun);  
    }  
  
    public String getNamaPohon() {  
        return namaPohon;  
    }  
  
    public void setNamaPohon(String namaPohon) {  
        this.namaPohon = namaPohon;  
    }  
}
```

3. Asosiasi (Association)

Merupakan suatu hubungan struktural antara beberapa objek yang menggambarkan objek dari suatu kelas dihubungkan dengan objek lain



Dalam implementasi, secara sintaks tidak memiliki perbedaan dengan implementasi agregasi, kecuali asosiasi bersifat dua arah. Kita bisa saja mendefinisikan asosiasi 2 arah yaitu objek objek dari kelas yang saling berhubungan bisa melakukan rujukan. Akan tetapi penggunaan asosiasi jenis ini dapat meningkatkan keterkaitan antar kelas atau dikenal dengan istilah **coupling**. Semakin tinggi coupling maka perancangan kelasnya kurang baik. Sedangkan program yang baik haruslah memiliki coupling yang rendah, sehingga untuk mengatasi permasalahan tersebut dibentuklah dengan menambah 1 kelas khusus untuk menjadi penengah



4. Kebergantungan (dependency)

Menyatakan bahwa suatu kelas bergantung pada kelas lain, maka perubahan pada kelas yang menjadi kebergantungan dari kelas lain menyebabkan perubahan terhadap kelas yang tergantung tersebut. Misalnya:



Misalnya jika kelas tumbuhan membutuhkan kelas air, jika kelas air mengalami perubahan maka menyebabkan perubahan pada kelas tumbuhan

Perwujudan relasi ini dapat dilakukan dalam 3 bentuk :

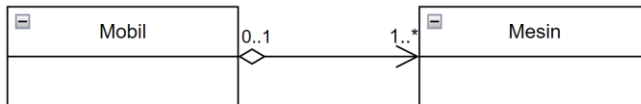
- Penggunaan kelas B sebagai parameter pada fungsi di kelas A.
- Penggunaan kelas B sebagai nilai balikan pada fungsi di kelas A.
- Penggunaan kelas B sebagai variabel lokal pada fungsi di kelas A.

Guided

Buatlah project baru dengan nama GD5_KSP_XXXX (XXXX npm terakhir kalian) dan jangan lupa centang create main class

☒ Create Main Class

a. Relasi Agregasi



Buatlah kelas bernama **Mobil.java** dan **Mesin.java** yang berisi seperti berikut

```
package gd5_ksp_XXXX;

/**
 *
 * @author daniel
 */
public class Mesin {
    private String namaMesin;
    private int tahunProduksi;

    public Mesin(String namaMesin, int tahunProduksi) {
        this.namaMesin = namaMesin;
        this.tahunProduksi = tahunProduksi;
    }

    public String getNamaMesin() {
        return namaMesin;
    }

    public void setNamaMesin(String namaMesin) {
        this.namaMesin = namaMesin;
    }

    public int getTahunProduksi() {
        return tahunProduksi;
    }

    public void setTahunProduksi(int tahunProduksi) {
        this.tahunProduksi = tahunProduksi;
    }

    public void tampilData(){
        System.out.println("Data Mesin");
        System.out.println("Nama Mesin : " +getNamaMesin());
        System.out.println("Tahun Produksi: "+getTahunProduksi());
    }
}
```

Mobil.java

```
package gd5_ksp_xxxx;

/**
 *
 * @author daniel
 */
public class Mobil {
    private String merkMobil;
    private double hargaMobil;
    private Mesin mesin;

    public Mobil(String merkMobil, double hargaMobil, Mesin mesin) {
        this.merkMobil = merkMobil;
        this.hargaMobil = hargaMobil;
        this.mesin = mesin;
    }

    public String getMerkMobil() {
        return merkMobil;
    }

    public void setMerkMobil(String merkMobil) {
        this.merkMobil = merkMobil;
    }

    public double getHargaMobil() {
        return hargaMobil;
    }

    public void setHargaMobil(double hargaMobil) {
        this.hargaMobil = hargaMobil;
    }

    public Mesin getMesin() {
        return mesin;
    }

    public void setMesin(Mesin mesin) {
        this.mesin = mesin;
    }

    public void tampilDataMobil(){
        System.out.println("Data Mobil");
        System.out.println("Merk Mobil : " +getMerkMobil());
        System.out.println("Harga Mobil: " +getHargaMobil());
        System.out.println("-----");
        mesin.tampilData();
    }
}
```

Lalu dalam main kita tambahkan code ini

```
public static void main(String[] args) {
    System.out.println("Relasi Agregasi");
    System.out.println("=====");
    Mesin mesin = new Mesin("Piston", 2002);
    Mobil mobil = new Mobil("Avansa", 1500000, mesin);
    mobil.tampilDataMobil();
}
```

Jika dirun maka akan menghasilkan

```
Relasi Agregasi
=====
Data Mobil
Merk Mobil : Avansa
Harga Mobil: 1500000.0
-----
Data Mesin
Nama Mesin : Piston
Tahun Produksi: 2002
```

b. Relasi Komposisi



Kita masih menggunakan project yang sama dan tambahkan kelas **Daun.java** dan **Pohon.java**
Daun.java

```
public class Daun {
    private String tulangDaun;
    private String warnaDaun;

    public Daun(String tulangDaun, String warnaDaun) {
        this.tulangDaun = tulangDaun;
        this.warnaDaun = warnaDaun;
    }

    public String getTulangDaun() {
        return tulangDaun;
    }

    public void setTulangDaun(String tulangDaun) {
        this.tulangDaun = tulangDaun;
    }

    public String getWarnaDaun() {
        return warnaDaun;
    }

    public void setWarnaDaun(String warnaDaun) {
        this.warnaDaun = warnaDaun;
    }

    public void tampilDaun(){
        System.out.println("Data Daun");
        System.out.println("-----");
        System.out.println("Tualng Daun : " +getTulangDaun());
        System.out.println("Warna Daun : " +getWarnaDaun());
    }
}
```

Pohon.java

```
public class Pohon {
    private String namaPohon;
    private int tinggiPohon;
    private Daun daun;

    public Pohon(String namaPohon, int tinggiPohon, String tulangDaun, String warnaDaun ) {
        this.namaPohon = namaPohon;
        this.tinggiPohon = tinggiPohon;
        this.daun = new Daun(tulangDaun, warnaDaun);
    }

    public String getNamaPohon() {
        return namaPohon;
    }

    public void setNamaPohon(String namaPohon) {
        this.namaPohon = namaPohon;
    }

    public int getTinggiPohon() {
        return tinggiPohon;
    }

    public void setTinggiPohon(int tinggiPohon) {
        this.tinggiPohon = tinggiPohon;
    }

    public void tampilPohon(){
        System.out.println("Data Pohon");
        System.out.println("Nama Pohon : " +getNamaPohon());
        System.out.println("Tinggi Pohon : " +getTinggiPohon());
        daun.tampilDaun();
    }
}
```

Lalu tinggal tambahkan code lagi di dalam mainnya seperti berikut

```
public static void main(String[] args) {
    System.out.println("Relasi Agregasi");
    System.out.println("=====");
    Mesin mesin = new Mesin("Piston", 2002);
    Mobil mobil = new Mobil("Avansa", 1500000, mesin);
    mobil.tampilDataMobil();

    System.out.println("\nRelasi Komposisi");
    System.out.println("=====");
    Pohon pohon = new Pohon("Mangga", 200, "Menyirip", "Kuning");
    pohon.tampilPohon();
}
```


Dan akan menghasilkan output seperti berikut

```
Relasi Komposisi
=====
Data Pohon
Nama Pohon   : Mangga
Tinggi Pohon : 200
Data Daun
-----
Tualng Daun : Menyirip
Warna Daun  : Kuning
```

c. Relasi Asosisasi 1 arah



Masih dalam project yang sama lalu tambahkan kelas **Dosen.java** dan **Matakuliah.java**

```
public class Matakuliah {
    private String namaMatkul;
    private String kodeMatkul;
    private int SKS;

    public Matakuliah(String namaMatkul, String kodeMatkul, int SKS) {
        this.namaMatkul = namaMatkul;
        this.kodeMatkul = kodeMatkul;
        this.SKS = SKS;
    }

    public String getNamaMatkul() {
        return namaMatkul;
    }

    public void setNamaMatkul(String namaMatkul) {
        this.namaMatkul = namaMatkul;
    }

    public String getKodeMatkul() {
        return kodeMatkul;
    }

    public void setKodeMatkul(String kodeMatkul) {
        this.kodeMatkul = kodeMatkul;
    }

    public int getSKS() {
        return SKS;
    }

    public void setSKS(int SKS) {
        this.SKS = SKS;
    }

    public void tampilMatakuliah(){
        System.out.println("Nama Matakuliah : " +getNamaMatkul());
        System.out.println("Kode Matakuliah : " +getKodeMatkul());
        System.out.println("Jumlah SKS      : " +getSKS());
    }
}
```

Dosen.java

Dalam kelas Dosen ini kita menambahkan Vector yang memiliki fungsi mirip dengan arraylist jadi jangan lupa untuk mengimport vectornya

```
package gd5_ksp_xxxx;

import java.util.Vector;

/**
 *
 * @author daniel
 */
public class Dosen {
    private String namaDosen;
    private String NIDN;
    private Vector<Matakuliah> daftarMatakuliah;

    public Dosen(String namaDosen, String NIDN) {
        this.namaDosen = namaDosen;
        this.NIDN = NIDN;
        this.daftarMatakuliah = new Vector<Matakuliah>();
    }

    public String getNamaDosen() {
        return namaDosen;
    }

    public void setNamaDosen(String namaDosen) {
        this.namaDosen = namaDosen;
    }

    public String getNIDN() {
        return NIDN;
    }

    public void setNIDN(String NIDN) {
        this.NIDN = NIDN;
    }

    public void tambahMatkul(Matakuliah M) {
        daftarMatakuliah.addElement(M);
    }
}
```

```

    public void tampilDosen() {
        System.out.println(getNamaDosen() + " || " + getNIDN());
        System.out.println("-----");
        for (int i = 0; i < daftarMatakuliah.size(); i++) {
            daftarMatakuliah.elementAt(i).tampilMatakuliah();
            System.out.println("");
        }
    }
}

```

Lalu tambahkan juga code di dalam mainnya

```

public static void main(String[] args) {
    System.out.println("Relasi Agregasi");
    System.out.println("=====");
    Mesin mesin = new Mesin("Piston", 2002);
    Mobil mobil = new Mobil("Avansa", 1500000, mesin);
    mobil.tampilDataMobil();

    System.out.println("\nRelasi Komposisi");
    System.out.println("=====");
    Pohon pohon = new Pohon("Mangga", 200, "Menyirip", "Kuning");
    pohon.tampilPohon();

    System.out.println("\nRelasi Asosiasi 1 Arah");
    System.out.println("=====");
    Dosen mahasiswa = new Dosen("Vincent", "200710700");

    Matakuliah Daspro, ISD, PBO;
    Daspro = new Matakuliah("Dasar Pemrograman", "DP", 4);
    ISD = new Matakuliah("Struktur Data", "ISD", 4);
    PBO = new Matakuliah("Pemrograman Berorientasi Objek", "PBO", 4);

    mahasiswa.tambahMatkul(Daspro);
    mahasiswa.tambahMatkul(ISD);
    mahasiswa.tambahMatkul(PBO);
    mahasiswa.tampilDosen();
}

```

Maka outpunya berupa

```
Relasi Asosiasi 1 Arah
=====
Vincent || 200710700
-----

Nama Matakuliah : Dasar Pemrograman
Kode Matakuliah : DP
Jumlah SKS      : 4

Nama Matakuliah : Struktur Data
Kode Matakuliah : ISD
Jumlah SKS      : 4

Nama Matakuliah : Pemrograman Berorientasi Objek
Kode Matakuliah : PBO
Jumlah SKS      : 4
```

Hasil Lengkapnya:

```
run:
Relasi Agregasi
=====
Data Mobil
Merk Mobil : Avansa
Harga Mobil: 1500000.0
-----

Data Mesin
Nama Mesin : Piston
Tahun Produksi: 2002

Relasi Komposisi
=====
Data Pohon
Nama Pohon   : Mangga
Tinggi Pohon : 200
Data Daun
-----

Tualng Daun : Menyirip
Warna Daun   : Kuning

Relasi Asosiasi 1 Arah
=====
Vincent || 200710700
-----

Nama Matakuliah : Dasar Pemrograman
Kode Matakuliah : DP
Jumlah SKS      : 4

Nama Matakuliah : Struktur Data
Kode Matakuliah : ISD
Jumlah SKS      : 4

Nama Matakuliah : Pemrograman Berorientasi Objek
Kode Matakuliah : PBO
Jumlah SKS      : 4
```