

**LAPORAN PRAKTIKUM PEMROGRAMAN
BERBASIS OBJEK
MODUL 7 RELASI ANTAR KELAS**



**Oleh :
Alif Alpian Sahrul Muharom
(20102007)**

**Dosen:
Agus Priyanto, M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
PURWOKERTO**

2022

I. TUJUAN

- Mengerti prinsip polimorfisme dalam bahasa C++ dan Java
- Mengerti tentang prinsip polimorfisme dan penggunaannya dalam membentuk suatu kelas.

II. TOOL

- Apache NetBeans IDE 13
- Java SE Development Kit 18

III. DASAR TEORI

- Asosiasi

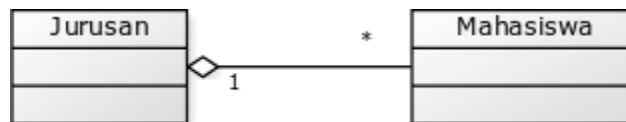


kelas diagram dengan hubungan asosiasi

Dapat dilihat pada gambar kelas diagram di atas, hubungan asosiasi digambarkan dengan satu garis tidak putus - putus dan tidak memiliki anak panah pada kedua ujungnya. Kemudian juga dibubuhkan kardinal pada garis tersebut.

Asosiasi adalah hubungan yang bisa saling menggunakan di dalam sebuah kelas, dan tidak saling memiliki. Misal pada gambar di atas terdapat kelas Dosen dan kelas Mahasiswa, dilihat dari kardinalitasnya, satu dosen bisa memiliki banyak mahasiswa. Apabila dosen dihilangkan, mahasiswa masih bisa digunakan karena tidak saling memiliki.

- Agregasi



kelas diagram dengan hubungan agregasi

Hubungan agregasi digambarkan dengan diamond putih, yang ditempelkan pada kelas yang memiliki, tidak dibubuhkan panah pada ujung yang tidak memiliki simbol diamond putih. Kemudian juga dibubuhkan kardinalitas seperti pada hubungan asosiasi.

Jika hubungan asosiasi adalah saling menggunakan, di sini hubungan yang terjadi adalah memiliki. Dilihat pada gambar kelas diagram di atas, bahwa kelas Jurusan memiliki kelas Mahasiswa sebagai variabelnya. Meskipun kelas Jurusan memiliki kelas Mahasiswa, namun kedua kelas tersebut dapat dibuat secara independen.

- Dependency**

Relasi kebergantungan (dependency) menyatakan bahwa suatu kelas bergantung pada kelas yang lain, maka perubahan pada kelas yang menjadi ketergantungan dari kelas lain menyebabkan perubahan terhadap kelas yang tergantung tersebut. Misalnya, kelas tumbuhan membutuhkan kelas air, jika kelas air mengalami perubahan, maka menyebabkan perubahan pada kelas tumbuhan.

IV. GUIDED

1. RELASI ASOSIASI

Manusia.java

```
package main.java.com.Alpian.Pertemuan_8.Guided.relasiAsosiasi;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Manusia {
    private String nama;
    private int umur;
    private Manusia ibu;
    private Manusia anak;

    public Manusia(){ }

    public Manusia(String nm, int umr){
        nama = nm;
        umur = umr;
        ibu = new Manusia();
        anak = new Manusia();

        ibu = null;
        anak = null;
    }

    public Manusia(String nm, int umr, Manusia ibu_angkat){
        nama = nm;
        umur = umr;
        ibu = new Manusia();
        anak = new Manusia();
        ibu = ibu_angkat;
        ibu_angkat.anak = this;
    }

    public void adopsi(Manusia anak_angkat){
        anak = anak_angkat;
        anak_angkat.ibu = this;
    }

    public void cetak(){
        System.out.println("\n- Data Pribadi -");
        System.out.println("Nama : " + nama);
        System.out.println("Umur : " + umur);

        if (ibu!=null) {
            System.out.println("Nama ibu : " + ibu.nama);
        }
    }
}
```

```

        } else if (anak!=null) {
            System.out.println("Nama anak : " + anak.nama);
        }
    }
}

```

Penjelasan:

Terdapat 2 atribut dalam kelas manusia, yaitu nama (String) dan umur (int). Private Manusia ibu dan Manusia anak merupakan deklarasi objek dari method Manusia(). Hubungan kedua kelasnya merupakan relasi asosiasi di mana kedua kelas menunjukkan variabel dalam suatu kelas yang menyimpan rujukan bertipe kelas lain.

IbuAnak.java

```

package main.java.com.Alpian.Pertemuan_8.Guided.relasiAsosiasi;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class IbuAnak {
    public static void main(String[] args) {
        Manusia ibu1 = new Manusia("Linda",18);
        Manusia anak1 = new Manusia("Cece", 20);

        Manusia ibu2 = new Manusia("Diana", 40);
        Manusia anak2 = new Manusia("Khoiron",17, ibu2);

        System.out.println("=====\n");
        ibu1.cetak();
        anak1.cetak();

        System.out.println("=====\n");
        ibu1.adopsi(anak1);
        ibu1.cetak();
        anak1.cetak();

        System.out.println("=====\n");
        ibu2.cetak();
        anak2.cetak();
    }
}

```

Penjelasan:

IbuAnak merupakan kelas Main dari Manusia.

2. RELASI AGREGASI

Pegawai.java

```
package main.java.com.Alpian.Pertemuan_8.Guided.relasiAgregasi;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Pegawai {
    private String nama;
    private String NIP;

    public Pegawai(){
        System.out.println("Konstruktor pegawai dijalankan");
    }

    public Pegawai(String NIP, String nama){
        this.NIP = NIP;
        this.nama = nama;
        System.out.println("Konstruktor pegawai dijalankan");
    }

    public void tampilPegawai(){
        System.out.println("NIP : "+NIP+" , nama : "+nama);
    }
}
```

Perusahaan.java

```
package main.java.com.Alpian.Pertemuan_8.Guided.relasiAgregasi;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Perusahaan {
    private String namaPerusahaan;
    private Pegawai Pegawai[];
    private int counter;

    public Perusahaan(String namaPerusahaan){
        this.namaPerusahaan = namaPerusahaan;
        counter=0;
        Pegawai = new Pegawai[3];
    }

    public void insertPegawai(Pegawai p){
```

```

        Pegawai[counter]=p;
        counter++;
    }

    public void tampilPerusahaan(){
        System.out.println("Perusahaan "+ namaPerusahaan +
"\nMemiliki pegawai : ");
        for (int i = 0; i < counter; i++) {
            Pegawai[i].tampilPegawai();
        }
    }
}

```

Penjelasan:

Kelas Pegawai dan kelas Perusahaan merupakan asosiasi agregasi. Jadi, kedua kelas saling membutuhkan dan saling melengkapi atribut dan method-methodnya.

Relasi.java

```

package main.java.com.Alpian.Pertemuan_8.Guided.relasiAgregasi;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Relasi {
    public static void main(String[] args) {
        Perusahaan Per = new Perusahaan("Stark Industries");
        Pegawai pegawai1,pegawai2,pegawai3;

        pegawai1 = new Pegawai("A001","Nazar");
        pegawai2 = new Pegawai("A002","Zayyan");
        pegawai3 = new Pegawai("A003","Farhan");

        Per.insertPegawai(pegawai1);
        Per.insertPegawai(pegawai2);
        Per.insertPegawai(pegawai3);

        System.out.println("");

        Per.tampilPerusahaan();
    }
}

```

Penjelasan:

Kelas main dari kelas Pegawai dan Perusahaan. Dalam kelas Main terdapat objek dari kelas Pegawai yang diisi sesuai method yang ada pada kelas pegawai.

V. UNGUIDED

Titik.java

```
package main.java.com.Alpian.Pertemuan_8.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Titik {
    double x, y;
    Titik t2;

    public Titik(){}

    public Titik(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public double hitungJarak(Titik t2){
        double jarak = Math.sqrt(Math.pow(t2.x - this.x, 2) +
Math.pow(t2.y - this.y, 2));
        return jarak;
    }

    public void tampil() {
        System.out.println("Titik x = " + x);
        System.out.println("Titik y = " + y);
    }
}
```

Penjelasan:

Kelas induk dari kelas-kelas lainnya. Terdapat atribut double x dan y serta Titik t2 sebagai objek yang digunakan. Pada kelas ini terdapat method untuk menghitung jarak antar titik x dan y bertipe data double.

Persegi.java

```
package main.java.com.Alpian.Pertemuan_8.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Persegi extends Titik{
    public double sisiPSG;
```

```

        public Persegi(double x1, double y1, double x2, double y2){
            super.x = x1;
            super.y = y1;
            t2 = new Titik (x2, y2);
        }
        public double hitungLuas(){
            sisiPSG = super.hitungJarak(t2);
            double Luas = sisiPSG * sisiPSG;
            return Luas;
        }
        public void tampil(){
            System.out.println("== DATA Persegi ==");
            System.out.println("Point          : [" + super.x + ", " +
super.y + "]");
            System.out.println("Sisi Segitiga : " + sisiPSG);
            System.out.println("Luas Segitiga : " + hitungLuas());
            System.out.println("");
        }
    }
}

```

Penjelasan:

Kelas Persegi merupakan turunan dari kelas Titik. Berarti, kelas ini mewarisi atribut dan method dari kelas Titik. Objek t2 dipanggil pada kelas ini dan dideklarasikan kembali objeknya khusus untuk kelas Persegi, yaitu x2 dan y2. Terdapat method menghitung luas persegi dan method untuk mencetak nilainya

SgtSamaSisi.java

```

package main.java.com.Alpian.Pertemuan_8.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class SgtSamaSisi extends Titik{
    public double sisiSGT;

    public SgtSamaSisi(double x1, double y1, double x2, double
y2){
        super.x = x1;
        super.y = y1;
        t2 = new Titik (x2, y2);
    }
    public double hitungLuas(){
        sisiSGT = super.hitungJarak(t2);
        double Luas = 0.5 * sisiSGT * Math.sqrt(3);
        return Luas;
    }
    public void tampil(){
        System.out.println("== DATA Segitiga ==");
        System.out.println("Point          : [" + super.x + ", " +

```



```

        super.y + "]");
        System.out.println("Sisi Segitiga : " + sisiSGT);
        System.out.println("Luas Segitiga : " + hitungLuas());
        System.out.println("");
    }
}

```

Penjelasan:

Sama seperti kelas Persegi, kelas ini juga merupakan turunan dari kelas titik. Method-methodnya juga sama dengan kelas Persegi.

Limas.java

```

package main.java.com.Alpian.Pertemuan_8.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Limas extends Persegi{
    private SgtSamaSisi Sgt;
    private Persegi Psg;

    public Limas(double Sx1, double Sy1, double Sx2, double Sy2,
double Px1, double Py1, double Px2, double Py2){
        super(Sx1, Sy1, Sx2, Sy2);
        Sgt = new SgtSamaSisi (Sx1, Sy1, Sx2, Sy2);
        Psg = new Persegi (Px1, Py1, Px2, Py2);
    }
    public double hitungLuas(){
        double luasSegitiga = Sgt.hitungLuas();
        double luasPersegi = Psg.hitungLuas();
        double Luas = ( 4 * luasSegitiga ) + luasPersegi;
        return Luas;
    }
    public void tampil(){
        Sgt.tampil();
        Psg.tampil();
        System.out.println("Luas Limas : " + hitungLuas());
    }
}

```

Penjelasan:

Beda dari kedua kelas sebelumnya. Kelas Limas merupakan turunan dari kelas Persegi. Jadi, kelas Limas tidak bisa memanggil atribut dari kelas Titik.

Relasi.java

```
package main.java.com.Alpian.Pertemuan_8.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Main
{
    public static void main(String[] args)
    {
        Limas P = new Limas(1.0,2.0,5.0,2.0,2.0,3.0,2.0,7.0);
        P.hitungLuas();
        P.tampil();
    }
}
```

Penjelasan:

Pada kelas Main hanya terdapat pengisian objek dari method di kelas Limas. Setelah di run, akan muncul data dari kelas Persegi, SgtSamaSisi, dan Limas.

VI. KESIMPULAN

Asosiasi adalah hubungan yang bisa saling menggunakan di dalam sebuah kelas, dan tidak saling memiliki. Hubungan agregasi digambarkan dengan diamond putih, yang ditempelkan pada kelas yang memiliki, tidak dibubuhkan panah pada ujung yang tidak memiliki simbol diamond putih. Kemudian juga dibubuhkan kardinalitas seperti pada hubungan asosiasi.

Relasi kebergantungan (dependency) menyatakan bahwa suatu kelas bergantung pada kelas yang lain, maka perubahan pada kelas yang menjadi ketergantungan dari kelas lain menyebabkan perubahan terhadap kelas yang tergantung tersebut.