

**LAPORAN PRAKTIKUM**  
**MODUL VI**  
**RELASI ANTAR KELAS**



**Disusun Oleh**

Firmansyah Nuralif Rohman  
18102050

**Dosen**

Trihastuti Yuniati, S.Kom., M.T.

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS TEKNOLOGI INDUSTRI DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI TELKOM**  
**PURWOKERTO**  
**2020**

## A. TUJUAN

1. Mahasiswa mampu memahami tentang konsep relasi antar kelas.
2. Mahasiswa mampu menerapkan konsep relasi antar kelas pada pemrograman berorientasi objek.

## B. DASAR TEORI

Pemrograman berorientasi objek (PBO) mengambil realita masalah dalam kehidupan sehari-hari. Dalam PBO sering terjadi relasi antara satu objek dengan objek yang lainnya. Ada beberapa relasi yang mungkin terjadi antara suatu kelas dengan kelas yang lain, yaitu:

### 1. **Inheritance / Pewarisan**

Merupakan konsep pewarisan dari kelas induk ke kelas anak. Sudah dijelaskan di modul sebelumnya

### 2. **Exception**

Agregasi adalah relasi antara dua buah objek dengan mengatakan bahwa suatu objek memiliki atau mengandung atau berisi objek yang lain. Misalnya, sebuah mobil memiliki mesin, roda, body; sebuah rumah memiliki dapur, kamar mandi, ruang makan, dsb. Apabila suatu objek tertentu tersusun atas objek-objek lain, maka objek tersebut dikatakan sebagai objek agregat atau objek komposit (aggregate, composite object). Relasi ini sering disebut dengan relasi 'has-a' atau relasi 'whole-part'. Dalam diagram UML, relasi agregat ini digambarkan dengan symbol diamond. Symbol ini menunjukkan adanya inklusi structural sebuah objek terhadap objek yang lain yang diwujudkan dengan cara membuat kelas agregat yang memiliki atribut yang bertipe kelas penyusun.

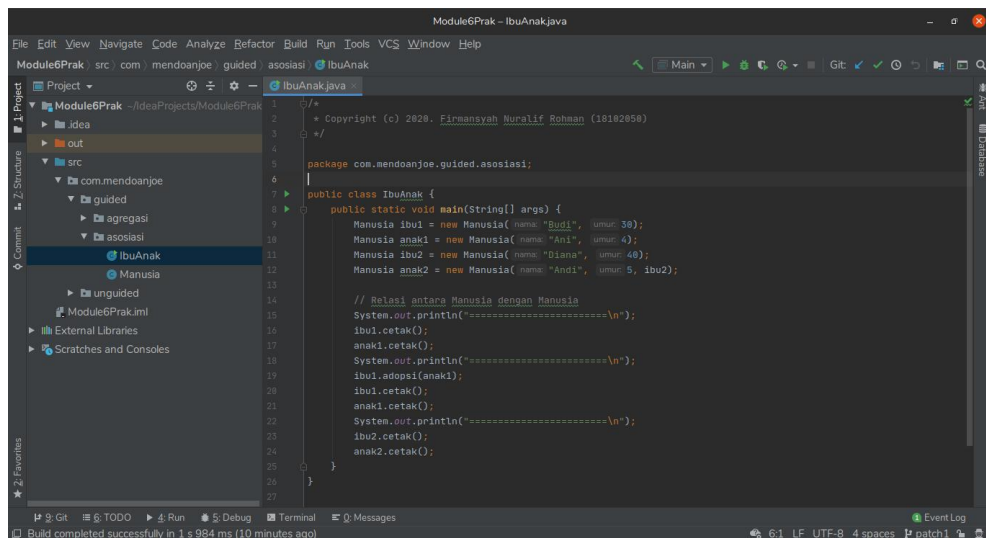
### 3. **Asosiasi**

Relasi asosiasi menyatakan suatu hubungan struktural antara beberapa objek yang menggambarkan objek dari suatu kelas dihubungkan dengan objek lain. Relasi ini menunjukkan variable dalam suatu kelas yang menyimpan rujukan bertipe kelas lain.

### 4. **Dependency**

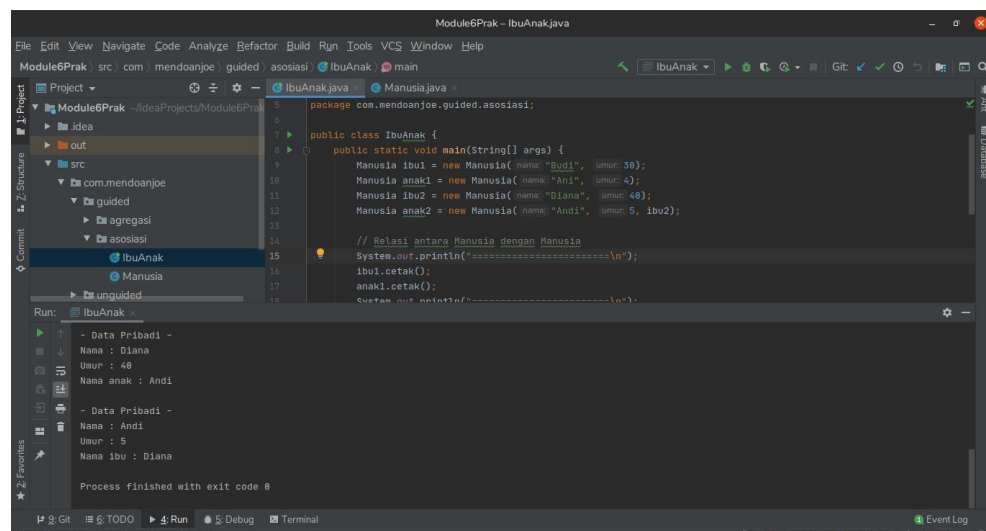
Relasi kebergantungan (dependency) menyatakan bahwa suatu kelas bergantung pada kelas yang lain, maka perubahan pada kelas yang menjadi ketergantungan dari kelas lain menyebabkan perubahan terhadap kelas yang tergantung tersebut. Misalnya, kelas tumbuhan membutuhkan kelas air. Jika kelas air mengalami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

## C. PEMBAHASAN PROGRAM GUIDED



Terdapat dua contoh modul program yaitu program dengan fungsi agregasi dan asosiasi. Secara singkat asosiasi adalah model yang berkaitan dengan model itu sendiri untuk memangun model baru. Agregasi adalah model yang berkaitan dengan model lain untuk melengkapi model yang telah ada.

### 1. Asosiasi



Disini terdapat 2 file java, yaitu kelas Manusia dan IbuAnak. Kelas Manusia digunakan sebagai model sedangkan kelas IbuAnak digunakan sebagai main program.

```

)/*
 * Copyright (c) 2020. Firmansyah Muralif Rohman (18102050)
 */

package com.mendoanjo.guided.asosiasi;
|
public class Manusia {
    private String nama;
    private int umur;
    private Manusia ibu;
    private Manusia anak;

    public Manusia() {

    }

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

    public Manusia (String nama, int umur, Manusia ibuAngkat) {
        this.nama = nama;

```

```

        this.umur = umur;
        ibu = new Manusia();
        anak = new Manusia();
        ibu = ibuAngkat;
        ibuAngkat.anak = this;
    }

    public void adopsi(Manusia anakAngkat) {
        anak = anakAngkat;
        anakAngkat.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);
    }
}

```

```

/*
 * Copyright (c) 2020. Firmansyah Nuralif Rohman (18102050)
 */

package com.mendoanjoe.guided.asosiasi;

public class IbuAnak {
    public static void main(String[] args) {
        Manusia ibu1 = new Manusia( nama: "Budi", umur: 30);
        Manusia anak1 = new Manusia( nama: "Ani", umur: 4);
        Manusia ibu2 = new Manusia( nama: "Diana", umur: 40);
        Manusia anak2 = new Manusia( nama: "Andi", umur: 5, ibu2);

        // Relasi antara Manusia dengan Manusia
        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();
    }
}

```

## 2. Agregasi

```

Module6Prak - Relasi.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Module6Prak src / com / mendoanjoe / guided / agregasi / Relasi
Project
Module6Prak -IdeaProjects/Module6Prak
  .idea
  out
  src
    com.mendoanjoe
      guided
        agregasi
          Pegawai
          Perusahaan
          Relasi
Run: Relasi x
  /usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
  Konstruktor perusahaan dijalankan...
  Konstruktor pegawai dijalankan...
  Konstruktor pegawai dijalankan...
  Konstruktor pegawai dijalankan...
  Perusahaan Nusantara Jaya memiliki pegawai :
  NIP : P801, Nama : Rudi
  NIP : P802, Nama : Toni
  NIP : P803, Nama : Ani
  Process finished with exit code 0

```

Pada module agregasi terdapat 3 file java, yaitu: Pegawai, Perusahaan dan Relasi. Relasi java digunakan sebagai main program dimana memanggil model Perusahaan dan Pegawai.

```

5/*
 * Copyright (c) 2020. Firmansyah Nuralif Rohman (18102050)
 */

package com.mendoanjoe.guided.agregasi;

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 tampilPeg() { System.out.println("NIP : " + NIP +", Nama : " + nama); }
}

```

```

package com.mendoanjoe.guided.agregasi;

class Perusahaan {
    private String namaPer;
    private Pegawai peg[]; // Agregasi antara pegawai dan perusahaan
    private int counter;

    public Perusahaan(String namaPer){
        this.namaPer = namaPer;
        counter = 0;
        peg = new Pegawai[3];
        System.out.println("Konstruktor perusahaan dijalankan...");
    }

    // Relasi agregasi antara pegawai dan perusahaan
    public void insertPegawai(Pegawai p){
        peg[counter] = p;
        counter++;
    }

    public void tampilPer(){
        System.out.println("Perusahaan "+namaPer+" memiliki pegawai : ");
        for (int i = 0; i < counter; i++) {
            peg[i].tampilPeg();
        }
    }
}

```

```

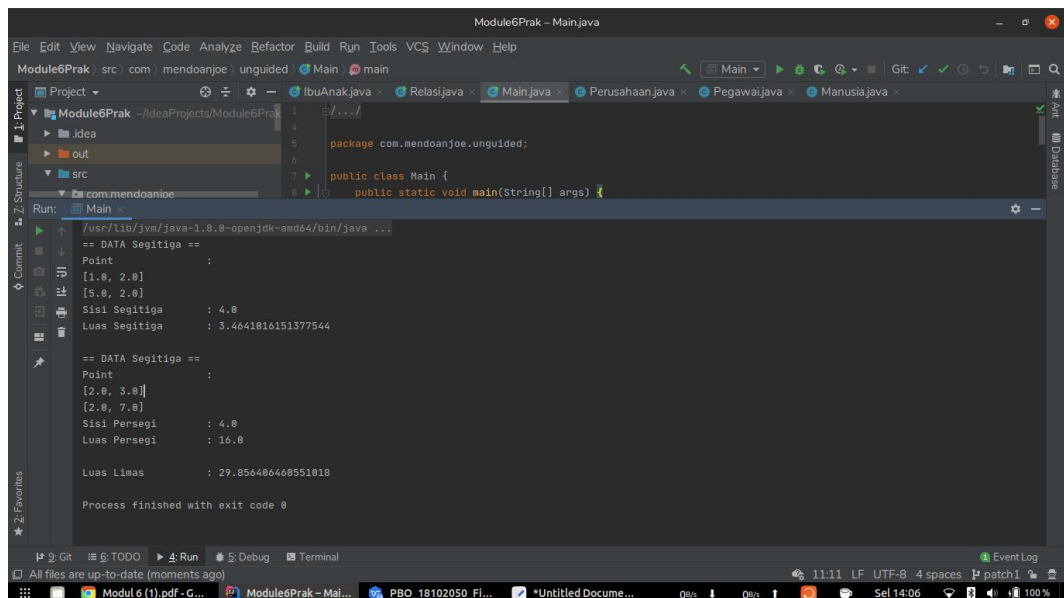
5/*
 * Copyright (c) 2020. Firmansyah Nuralif Rohman (18102050)
 */

package com.mendoanjoe.guided.agregasi;

public class Relasi {
    public static void main(String[] args) {
        Perusahaan Per = new Perusahaan( namaPer: "Nusantara Jaya");
        Pegawai Peg1, Peg2, Peg3;
        Peg1 = new Pegawai( NIP: "P001", nama: "Rudi");
        Peg2 = new Pegawai( NIP: "P002", nama: "Toni");
        Peg3 = new Pegawai( NIP: "P003", nama: "Ani");
        // Relasi antara Class Pegawai dengan Class Perusahaan
        Per.insertPegawai(Peg1);
        Per.insertPegawai(Peg2);
        Per.insertPegawai(Peg3);
        System.out.println();
        Per.tampilPer();
    }
}

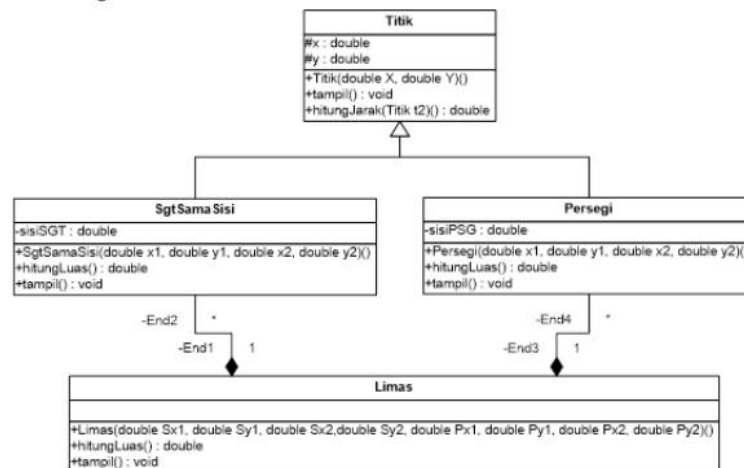
```

## D. PEMBAHASAN PROGRAM UNGUIDED



```
Module6Prak - Main.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Module6Prak src / com / mendoanjoe / unguided / Main / main
Project
Module6Prak -IdeaProjects/Module6Prak
src
com.mendoanjoe
Run: Main
Run: /usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
== DATA Segitiga ==
Point :
[1.0, 2.0]
[5.0, 2.0]
Sisi Segitiga : 4.0
Luas Segitiga : 3.4641816151377544
== DATA Segitiga ==
Point :
[2.0, 3.0]
[2.0, 7.0]
Sisi Persegi : 4.0
Luas Persegi : 16.0
Luas Limas : 29.856486468551818
Process finished with exit code 0
```

Terdapat empat kelas model dan satu kelas main, yaitu: Limas, Persegi, SgtSamaSisi, Titik, dan Main. Program di buat mengikuti map logika dibawah ini:



Dengan ketentuan:

1. Program di implementasikan menggunakan metode relasi
2. Input program sudah ditentukan
3. Rumus sudah di tentukan
4. Hasil output sudah ditentukan

Berikut porgramnya yang sudah dibuat:



```

/*
 * Copyright (c) 2020. Firmansyah Nuralif Rohman (18102050)
 */

package com.mendoanjoe.unguided;

public class Titik {
    private double x;
    private double y;

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

    public void tampil() {
        System.out.println("[ " + x + ", " + y + " ]");
    }

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

```

```

package com.mendoanjoe.unguided;

public class SgtSamaSisi {
    private double sisiSGT;
    private Titik titik;
    private Titik titik2;

    public SgtSamaSisi(double x1, double y1, double x2, double y2) {
        this.titik = new Titik(x1, y1);
        this.titik2 = new Titik(x2, y2);
        this.sisiSGT = titik.hitungJarak(titik2);
    }

    public double hitungLuas() {
        // return 0.5 * sisiSGT * (sisiSGT * 0.5 * Math.sqrt(3)); // 1/2 * alas * tinggi (alas * 1/2 * akar3)
        return 0.5 * sisiSGT * Math.sqrt(3); // menyesuaikan module
    }

    public void tampil() {
        System.out.println("== DATA Segitiga ==");
        System.out.println("Point \t\t\t\t: ");
        titik.tampil();
        titik2.tampil();
        System.out.println("Sisi Segitiga \t\t: " + sisiSGT);
        System.out.println("Luas Segitiga \t\t: " + this.hitungLuas());
        System.out.println();
    }
}

```

```

package com.mendoanjoe.unguided;

public class Persegi {
    private double sisiPSG;
    private Titik titik;
    private Titik titik2;

    public Persegi(double x1, double y1, double x2, double y2) {
        this.titik = new Titik(x1, y1);
        this.titik2 = new Titik(x2, y2);
        this.sisiPSG = titik.hitungJarak(titik2);
    }

    public double hitungLuas() {
        return sisiPSG * sisiPSG;
    }

    public void tampil() {
        System.out.println("== DATA Segitiga ==");
        System.out.println("Point \t\t\t\t: ");
        titik.tampil();
        titik2.tampil();
        System.out.println("Sisi Persegi \t\t: " + sisiPSG);
        System.out.println("Luas Persegi \t\t: " + hitungLuas());
        System.out.println();
    }
}

```

```

/*
 * Copyright (c) 2020. Firmansyah Nuralif Rohman (18182050)
 */

package com.mendoanjoe.unguided;

public class Limas {
    private static final int MAX_SEGITIGA_SAMA_SISI = 4;
    private SgtSamaSisi sgtSamaSisi;
    private Persegi persegi;

    public Limas(double sx1, double sy1, double sx2, double sy2, double px1, double py1, double px2, double py2) {
        this.sgtSamaSisi = new SgtSamaSisi(sx1, sy1, sx2, sy2);
        this.persegi = new Persegi(px1, py1, px2, py2);
    }

    public double hitungLuas() {
        return (MAX_SEGITIGA_SAMA_SISI * this.sgtSamaSisi.hitungLuas()) + this.persegi.hitungLuas();
    }

    public void tampil() {
        this.sgtSamaSisi.tampil();
        this.persegi.tampil();
        System.out.println("Luas Limas \t\t\t: " + this.hitungLuas());
    }
}

```

## Kesimpulan

Program dengan metode relasi memang dapat menghemat penulisan kode program pada main kelas, akan tetapi dalam pengkodean dengan menggunakan metode relasi harus lebih teliti dan model bersifat dapat diakses dimana saja.