**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL VII**

**RELASI ANTAR KELAS**

****

**DOSEN PENGAMPU :**

Agus Priyanto, S. Kom., M.Kom.

**DISUSUN OLEH :**

Filfimo Yulfiz Ahsanul Hulqi

19102143

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**FAKULTAS INFORMATIKA**

**PRODI S1 INFORMATIKA**

**2020**

**BAB I**

**Tujuan Praktikum**

Mahasiswa diharapkan mampu untuk :

1. Mengerti dan memahami tentang konsep relasi antar kelas.
2. Mampu mewujudkan berbagai jenis relasi kelas dalam Java.

**BAB II**

**Dasar Teori**

Permrograman OO (*Object Oriented*) mengambil realita masalah dalam kehidpan sehari-hari. Dalam OO, sering terjadi relasi antara satu objek dengan objek lainnya.

Ada beberapa relasi yang mungkin terjadi antara suatu kelas dengan kelas yang lain, yaitu :

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

Terdiri dari pewarisan tunggal (*single inherintance*) dan pewarisan jamak (*multiple inheritance*). Kelas yang diciptakan disebut dengan kelas anak dan kelas asalnya disebut kelas induk. Kelas anak akan mewarisi seluruh *method* dan properti yang dimiliki oleh kelas induknya. *Inheritance* sudah dijelaskan dan dikerjakan lebih lanjut pada modul-modul sebelumnya.

1. **Agregasi**

Agregasi adalah relasi antara dua buah objek dengan mengatakan bahwa satu objek memiliki atau mengandung atau berisi objek lain. Apabila suatu objek tersusun atas objek-objek lain, maka objek tersebut dikatakan sebagai objek agregat atau objek komposit (*agregate*, *composite object*). Relasi ini sering disebut juga dengan relasi ”*has*-*a*” atau relasi ”*whole*-*part*”.

1. **Asosiasi**

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

1. **Dependency / Kebergantungan**

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

1. **Composition / Komposisi**

Komposisi (*composition*) merupakan relasi yang lebih spesifik dari relasi Agregasi (*Strong Aggregation*). Karena setiap kelas tidak dapat berdiri sendiri

**BAB III**

**PENJELASAN**

**GUIDED**

1. Buat program untuk menampilkan informasi ibu dan anak, dengan menggunakan implementasi relasi asosiasi.

package ibuanak;

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;

System.out.println("Konstruktor Manusia dijalankan");

}

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;

System.out.println("Konstruktor Manusia dengan Ibu Angkat dijalankan");

}

public void adopsi(Manusia anak\_angkat){

anak = anak\_angkat;

anak\_angkat.ibu = this;

}

public void cetak(){

System.out.println("- 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);

}

}

*Source code class* Manusia

*Source code main class* IbuAnak untuk *running program*

package ibuanak;

public class IbuAnak {

public static void main(String[] args) {

Manusia ibu1 = new Manusia("Amel", 30);

Manusia anak1 = new Manusia("Raihan", 4);

Manusia ibu2 = new Manusia("Diana", 40);

Manusia anak2 = new Manusia("Andi", 5, ibu2);

System.out.println("======================");

ibu1.cetak();

anak1.cetak();

System.out.println("======================");

ibu1.adopsi(anak1);

ibu1.cetak();

anak1.cetak();

System.out.println("======================");

ibu2.cetak();

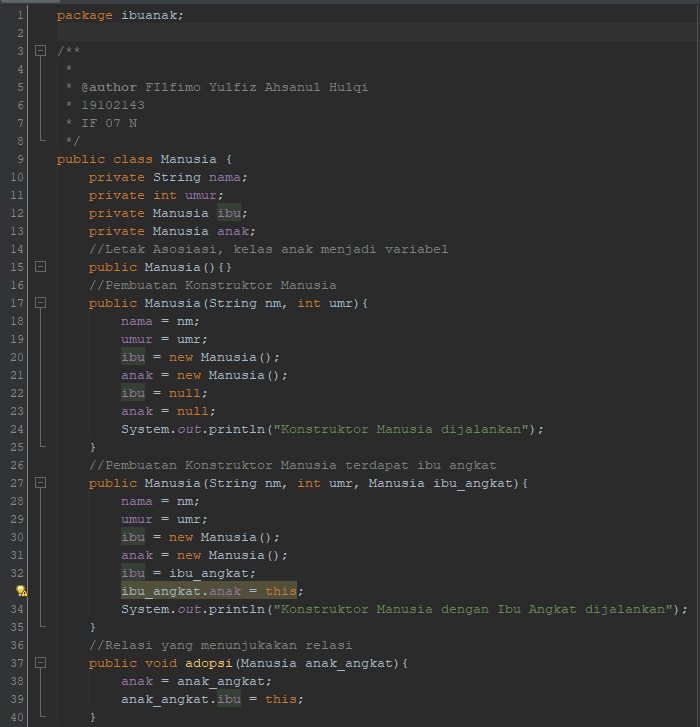
anak2.cetak();

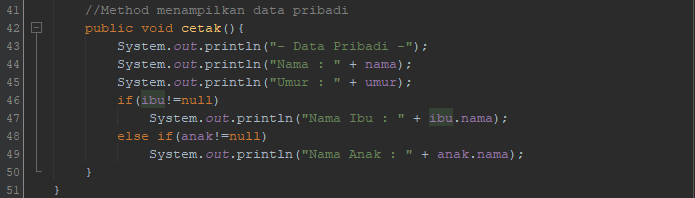
}

}

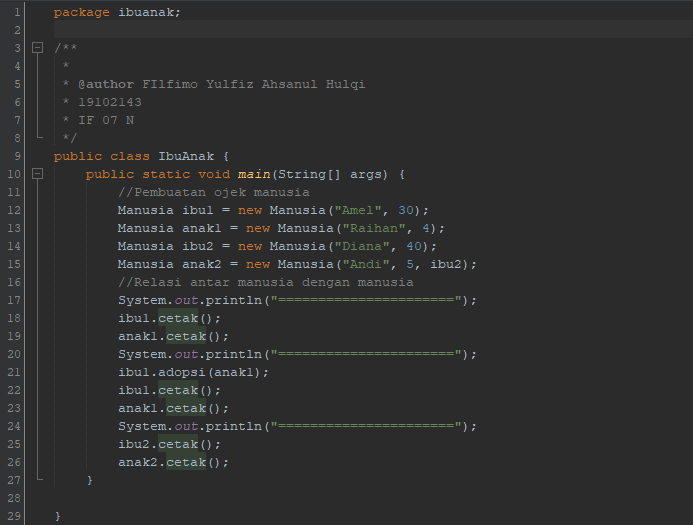
Screenshot program

* *Class* Manusia

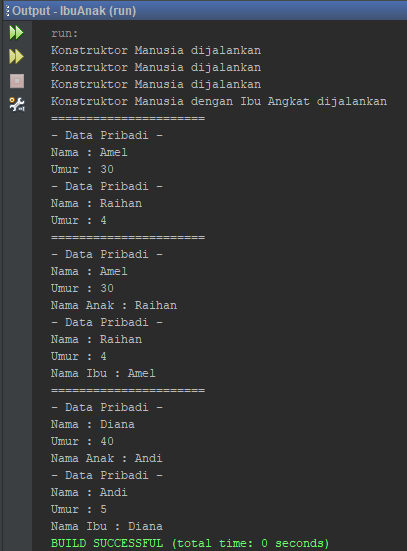




* *Main Class* IbuAnak



* *Output*



Penjelasan Program :

Pada program diatas adalah contoh implementasi relasi antar kelas, relasi yang digunakan adalah relasi asosiasi. Buat terlebih dahulu dua file Java, yaitu Manusia dan IbuAnak.

*Class* Manusia, didalamnya terdapat deklarasi *fields*, peletakan asosiasi kelas anak menjadi variabel, pebuatan konstruktor, pembuatan *method* adopsi yang akan menunjukkan relasi antar kelas, dan pembuatan *method* cetak untuk menampilkan data.

*Class* IbuAnak untuk *running program*, didalamnya terdapat pembuatan objek dari *class* Manusia yaitu ibu dan anak, selanjutnya adalah melakukan pemanggilan *method* cetak dari ibu dan anak untuk menampilkan *output*.

1. Buat program untuk menampilkan informasi data pegawai suatu perusahaan, dengan menggunakan implementasi relasi agregasi.

package relasi;

public class Pegawai {

private String nama, 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 dengan NIP&Nama dijalankan....");

}

public void tampilPeg(){

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

}

}

*Source code class* Pegawai

package relasi;

public class Perusahaan {

private String namaPer;

private Pegawai peg[];

private int counter;

public Perusahaan(String namaPer){

this.namaPer = namaPer;

counter = 0;

peg = new Pegawai[4];

System.out.println("Konstruktor Perusahaan dijalankan....");

}

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();

}

}

}

*Source code class* Perusahaan

package relasi;

public class Relasi {

public static void main(String[] args) {

//Pembuatan objek perusahaan dan input nama didalam konstruktor

Perusahaan Per = new Perusahaan("IF 07 N PRAK PBO");

//Pembuatan objek peagawai

Pegawai Peg1, Peg2, Peg3, Peg4;

//Input nama didalam konstruktor

Peg1 = new Pegawai("IF07","Filfimo");

Peg2 = new Pegawai("1910","Yulfiz");

Peg3 = new Pegawai("2143","Ahsanul");

Peg4 = new Pegawai("2019","Samsul");

//Relasi antar Class Pegawai dengan Class Perusahaan

Per.insertPegawai(Peg1);

Per.insertPegawai(Peg2);

Per.insertPegawai(Peg3);

Per.insertPegawai(Peg4);

//Pemanggilan output

Per.tampilPer();

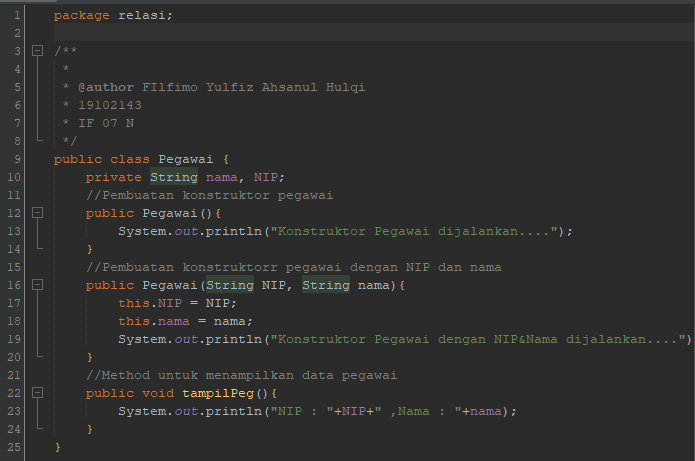
}

}

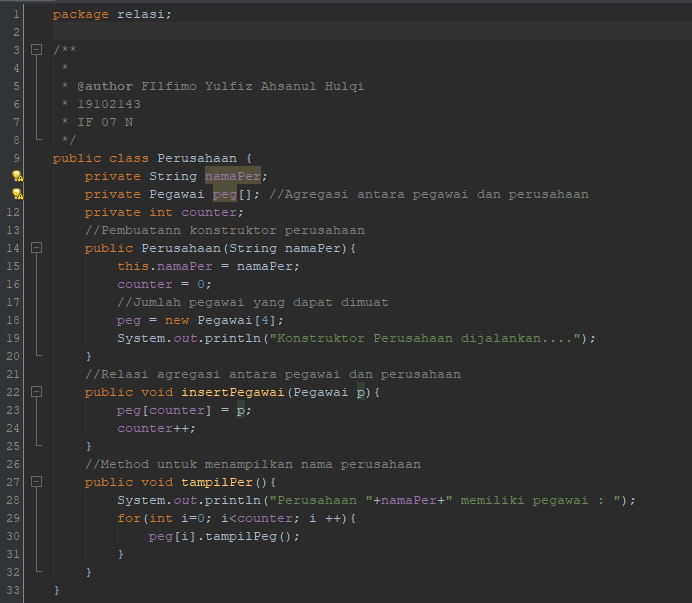
*Source code main class* Relasi untuk *running program*

Screenshot Program

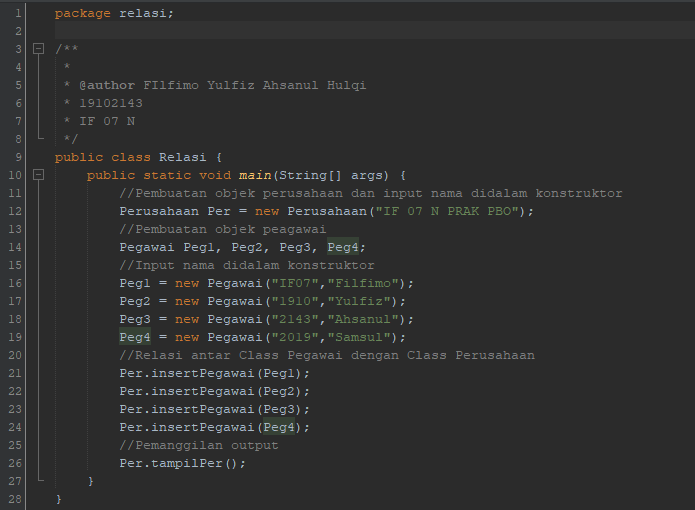
* *Class* Pegawai



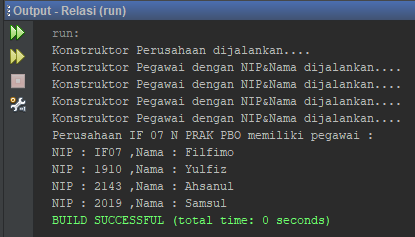
* *Class* Perusahaan



* *Main Class* Relasi



* *Output*



Penjelasan Program :

Pada program diatas adalah contoh implementasi relasi antar kelas, relasi yang digunakan adalah relasi agregasi. Buat terlebih dahulu tiga file Java, yaitu Pegawai, Perusahaan dan Relasi.

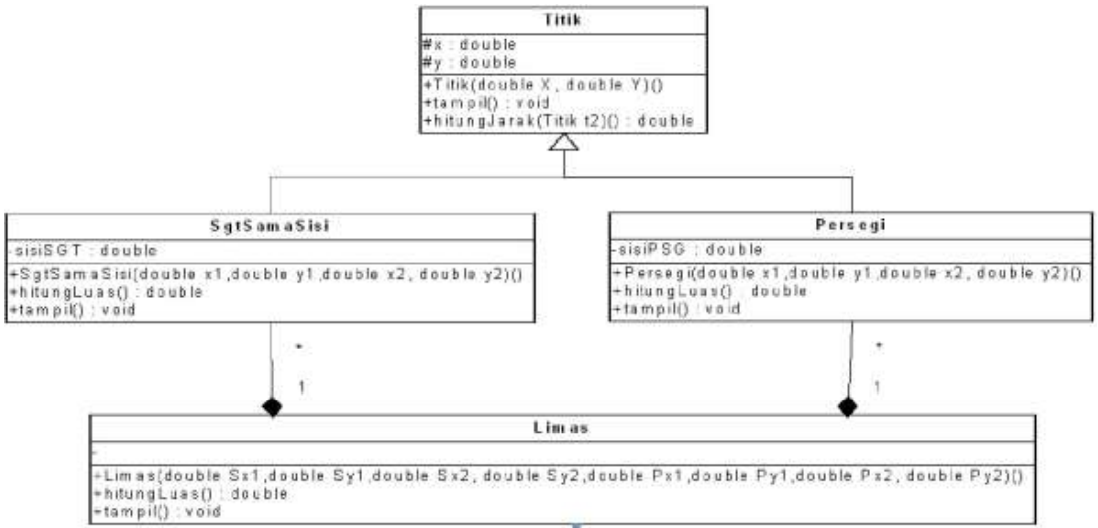
*Class* Pegawai, didalamnya terdapat deklarasi *fields*, pebuatan konstruktor, pembuatan *method* tampilPeg untuk menampilkan data dari pegawai perusahaan.

*Class* Perusahaan, didalamnya terdapat deklarasi *fields*, pembuatan objek dari *class* Pegawai, melakukan pembuatan agregasi antara *class* Pegawai dan *class* Perusahaan, pembuatan konstruktor perusahaan didalamnya terdapat jumlah data pegawai yang dapat dimuat, pembuatan *method insert*Pegawai untuk melakukan *insert* data pegawai, terakhir pembuatan *method* tampilPer untuk menampilkan nama dari perusahaan.

*Class* Relasi untuk *running program*, didalamnya terdapat pembuatan objek perusahan dan input nama perusahaan didalam konstruktor, pembuatan objek pegawai, ada empat objek pegawai, *input* nama pegawai didalam konstruktor, kemudian melakukan *insert*Pegawai, terakhir melakukan pemanggilan *output* dengan *method* tampilPer dari *class* Perusahaan.

**UNGUIDED**

Buatlah kelas Limas yang terdiri dari kelas SgtSamaSisi dan kelas Persegi. Kelas SgtSamaSisi dan kelas Persegi adalah turunan dari kelas Titik



Dengan ketentuan dan spesifikasi dari masing-masing kelas, sesuai dengan Modul Praktikum 7 – Relasi Antar Kelas.

package hitung;

public class Titik {

protected double x;

protected double y;

Titik(){

}

Titik(double x, double y) {

this.x = x;

this.y = y;

}

void tampil(Titik t2, Titik t1){

System.out.println("Titik 1 : " +t1.x+", " +t1.y);

System.out.println("Titik 2 : " +t2.x+", " +t2.y);

System.out.println("Jarak titik : " + hitungJarak(t2,t1));

}

double hitungJarak(Titik t2, Titik t1){

double jarakX = Math.pow((t2.x - t1.x), 2);

double jarakY = Math.pow((t2.y - t1.y), 2);

return Math.sqrt((jarakX + jarakY));

}

*Source code class* Titik bagian 1

*Source code class* Titik bagian 2

package hitung;

public class Persegi extends Titik {

public double sisiPSG;

Titik tp1, tp2;

public Persegi(){

}

public Persegi(double x1, double y1, double x2, double y2) {

tp1 = new Titik(x1,y1);

tp2 = new Titik(x2,y2);

this.sisiPSG = super.hitungJarak(tp2, tp1);

}

public double hitungLuas(){

return Math.pow(this.sisiPSG, 2);

}

void tampil(){

System.out.println("===== DATA PERSEGI =====");

System.out.print("Point : "); tampilTitik(this.tp1, this.tp2);

System.out.println("Sisi Persegi : " + this.sisiPSG );

System.out.println("Luas Persegi : " + hitungLuas() + "\n" );

}

}

void tampilTitik(Titik t2, Titik t1){

System.out.println("[" + t1.x +", "+ t1.y+ "]," + "[" + t2.x +","+ t2.y +"]");

}

}

*Source code* class Persegi

package hitung;

public class SgtSamaSisi extends Titik {

public double sisiSGT;

Titik ts1, ts2;

public SgtSamaSisi(){

}

public SgtSamaSisi(double x1, double y1, double x2, double y2) {

ts1 = new Titik(x1,y1);

ts2 = new Titik(x2,y2);

this.sisiSGT = super.hitungJarak(ts2, ts1);

}

*Source code class* SgtSamaSisi bagian 1

*Source code class* SgtSamaSisi bagian 2

package hitung;

public class Limas {

SgtSamaSisi s;

Persegi p;

public Limas(){

}

public Limas(double xx1, double xy1, double xx2, double xy2, double yx1, double yy1, double yx2, double yy2){

this.s = new SgtSamaSisi(xx1, yy1, xx2, xy2);

this.p = new Persegi(yx1, yy1, yx2, yy2);

}

public double hitungLuas(){

return (4\*this.s.hitungLuas()) + this.p.hitungLuas();

}

public void tampilLimas(){

this.s.tampil();

this.p.tampil();

System.out.println("===== DATA LIMAS =====");

System.out.println("Luas Limas : " + this.hitungLuas() + "\n");

}

}

public double hitungLuas(){

return 0.5 \* this.sisiSGT \* Math.sqrt(3);

}

void tampil(){

System.out.println("===== DATA SEGITIGA =====");

System.out.print("Point : "); tampilTitik(this.ts1, this.ts2);

System.out.println("Sisi Segitiga : " + this.sisiSGT);

System.out.println("Luas Segitiga : " + hitungLuas() + "\n");

}

}

*Source code class* Limas

package hitung;

public class Hitung {

public static void main(String[] args) {

System.out.println("FILFIMO YULFIZ AHSANUL HULQI (19102143) IF 07 N" + "\n");

Limas P = new Limas(1.0,2.0,5.0,2.0,2.0,3.0,2.0,7.0);

P.tampilLimas();

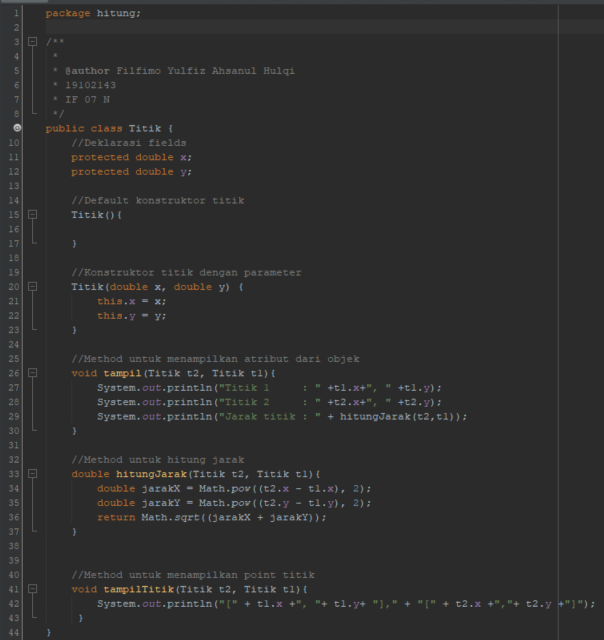
}

}

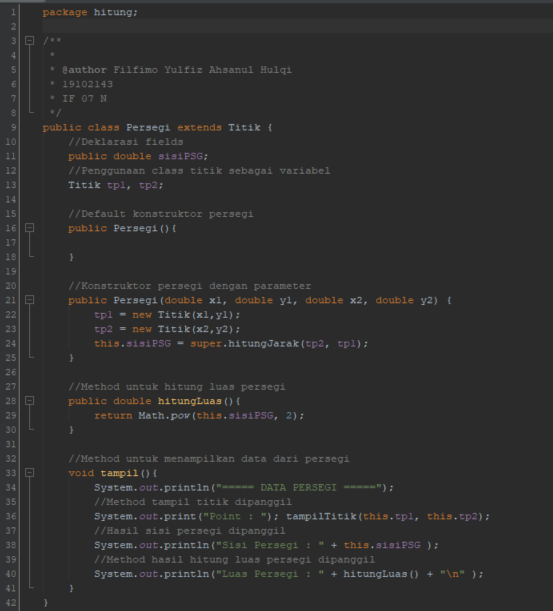
*Source code class* Hitung untuk *running program*

Screenshot Program

* *Class* Titik



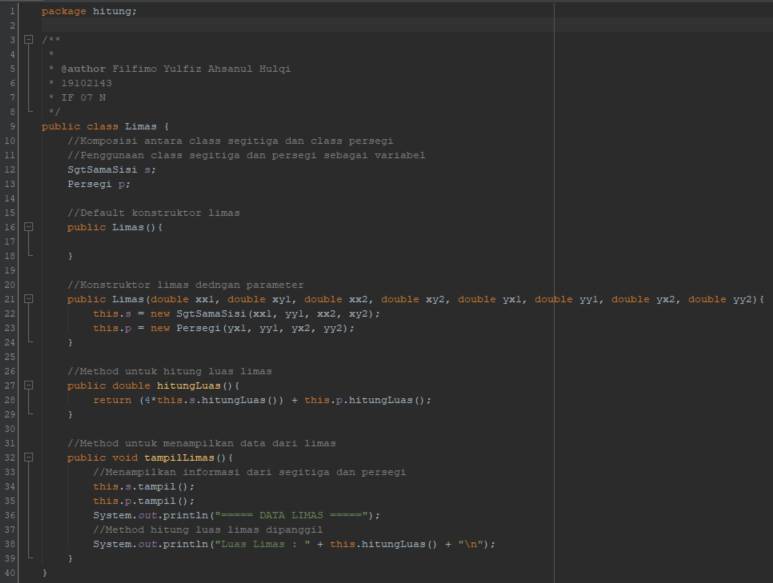
* *Class* Persegi



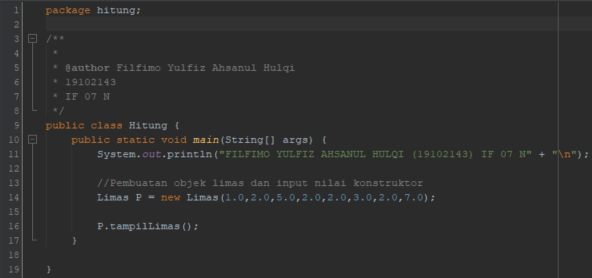
* *Class* SgtSamaSisi



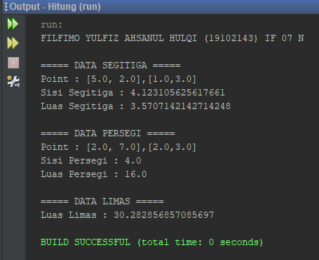
* *Class* Limas



* *Class* Hitung untuk *running program* (*main class*)



* *Output*



Penjelasan Program :

Pada program kali ini diminta untuk mengimplementasikan relasi antar kelas, relasi yang digunakan yaitu komposisi untuk menghitung Limas.

Buat terlebih dahulu lima file Java, yaitu Titik, SgtSamaSisi, Persegi, Limas, dan Hitung (*main*).

*Class* Titik, kelas ini nantinya akan diturunkan ke kelas SgtSamaSisi dan Persegi, didalamnya terdapat deklarasi *fields* pembuatan konstruktor *default* dan konstruktor dengan parameter, pembuatan *method* untuk menampilkan atribut dari objek, pembuatan *method* untuk menghitung jarak antar titik, dan pembuatan *method* untuk menampilkan *point* titik.

*Class* SgtSamaSisi, kelas ini adalah turunan dari kelas Titik, didalamya terdapat deklarasi *fields*, penggunaan kelas Titik sebagai variabel, melakukan pembuatan konstruktor *default* dan konstruktor dengan parameter, didalam konstruktor parameter ini terdapat *keyword* *super* untuk mengakses atribut dari *super class* atau kelas induknya, selanjutnya pembuatan *method* untuk menghitung luas SgtSamaSisi, dan terakhir adalah *method* untuk menampilkan seluruh data dari SgtSamaSisi.

*Class* Persegi, kelas ini adalah turunan dari kelas Titik, didalamnya terdapat deklarasi *fields*, penggunaan kelas Titik sebagai variabel, melakukan peembuatan konstruktor *default* dan konstruktor dengan parameter, sama seperti sebelumnya didalam konstruktor dednganparameter, didalam konstruktor parameter ini terdapat *keyword super* untuk mengaksees atribut *super class* atau kelas induknya, selanjutnya pembuatan *method* untuk menghitung luas Persegi, dan pembuatan *method* untuk menampilkan seluruh data dari Persegi.

*Class* Limas, kelas ini akan mengimplemntasikan komposisi antara kelas SgtSsamaSisi dan kelas Persegi, didalamnya diawali dengan pendeklarian penggunaan kelas SgtSamaSisi dan Persegi sebagai variabbel, kemudian dilanjut pembuatan konstruktor *default* dan konstruktor dengan parameter, seelanjutnya pembuatan *method* untuk menghitung luas Limas, dan pembuatan *method* untuk menampilkan informasi data dari Limas, tetapi didalam *method* ini juga akan menampilkan informasi data dari SgtSamaSisi dan Persegi.

*Class* Hitung untuk *running program*, didalamnya sangat *simple* hanya *print* nama dari mahasiswa, pembuatan objek dari Limas lalu isi juga konstruktornya dan terakhir lakukan pemanggilan *method* tampil untuk menampilkan *output*.

**BAB IV**

**Kesimpulan**

Pada praktikum kali ini dapat diambil beberapa kesimpulan :

1. Implementasi program akan berbeda-beda sesuai dengan relasi apa yang digunakan.
2. Struktur dari masing-masing relasi antar kelas berbeda-beda.
3. Memahami lebih dalam konsep dari *Object Oriented* karena relasi antar kelas juga mengambil/bagian dari konsep dari *Object Oriented*.