

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK
PERTEMUAN 7



Disusun oleh :

Nama : Fatimah Az Zahra

NIM : 2211102160

Kelas : IF-10-K

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024

MODUL 0

A. Guided

1. Manusia.java

Source Code

```
5 package GUIDED1;
6 //Fatimah Az Zahra_2211102160_IF10K
7 public class Manusia {
8     //mendefinisikan atribut
9     private String nama;
10    private int umur;
11    //mendefinisikan relasi
12    private Manusia ibu;
13    private Manusia anak;
14
15    //konstruktor
16    public Manusia() {
17    }
18    //konstruktor nama dan umur
19    public Manusia(String nm, int umr) {
20        nama = nm;
21        umur = umr;
22        ibu = new Manusia(); //fungsi membuat objek baru
23        anak = new Manusia();
24        ibu = null; //fungsi set nilai relasi (null)
25        anak = null;
26    }
27
28    //inisialisasi untuk keluarga angkat
29    public Manusia(String nm, int umr, Manusia ibu_angkat) {
30        nama = nm;
31        umur = umr;
32        ibu = new Manusia(); //fungsi membuat objek baru
33        anak = new Manusia();
34        ibu = ibu_angkat; //menetapkan nilai relasi ibu_angkat sesuai parameter
35        ibu_angkat.anak = this; //relasi kita atur ke objek ini
36    }
```

IbuAnak.java

Source Code

```
5 package GUIDED1;
6 //Fatimah Az Zahra_2211102160_IF10K
7 //kelas main dari manusia
8 public class IbuAnak {
9     public static void main(String[] args) {
10        Manusia ibu1 = new Manusia("Budi", 30);
11        Manusia anak1 = new Manusia("Ani", 4);
12        Manusia ibu2 = new Manusia("Diana", 40);
13        Manusia anak2 = new Manusia("Andi", 5, ibu2);
14        //Relasi antara Manusia dengan Manusia
15        System.out.println("=====\n");
16        //hanya cetak
17        ibu1.cetak();
18        anak1.cetak();
19        System.out.println("=====\n");
20        ibu1.adopsi(anak1);
21        ibu1.cetak();
22        anak1.cetak();
23        System.out.println("=====\n");
24        ibu2.cetak();
25        anak2.cetak();
26    }
27 }
```

Output :

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul7 ---
=====

- Data Pribadi -
Nama : Budi
Umur : 30

- Data Pribadi -
Nama : Ani
Umur : 4
=====

- Data Pribadi -
Nama : Budi
Umur : 30
Nama anak : Ani

- Data Pribadi -
Nama : Ani
Umur : 4
Nama ibu : Budi
=====

- Data Pribadi -
Nama : Diana
Umur : 40
Nama anak : Andi

- Data Pribadi -
Nama : Andi
Umur : 5
Nama ibu : Diana

-----
BUILD SUCCESS
-----

Total time:  3.187 s
Finished at: 2024-05-06T10:27:50+07:00
-----
```

Penjelasan program:

Program di atas mengilustrasikan hubungan antara ibu dan anak dalam pemrograman berorientasi objek. Kelas `IbuAnak` bertindak sebagai titik masuk program, yang menciptakan objek-objek `Manusia` untuk merepresentasikan ibu dan anak. Melalui kelas `Manusia`, hubungan antara ibu dan anak diuji dan ditampilkan. Kelas `Manusia` memiliki atribut seperti `nama`, `umur`, `ibu`, dan `anak`, yang mencerminkan hubungan antara individu dengan ibu dan anaknya. Metode `adopsi()` digunakan untuk menetapkan hubungan antara ibu dan anak, sementara metode `cetak()` menampilkan informasi pribadi tentang manusia, termasuk nama, umur, dan hubungan dengan ibu atau anaknya. Relasi antara kelas `IbuAnak` dan `Manusia` adalah hubungan asosiasi, di mana kelas-kelas tersebut berinteraksi satu sama lain tanpa ketergantungan yang kuat.

2. Perusahaan.Java

Source Code

```
5 package GUIDED2;
6 //Fatimah Az Zahra_2211102160_IF10K
7 // Class Perusahaan
8
9 class Perusahaan {
10
11     private String namaPer;
12     private Pegawai peg[]; //Agregasi antara pegawai dan perusahaan
13     private int counter;
14
15     public Perusahaan(String namaPer) {
16         this.namaPer = namaPer;
17         counter = 0;
18         peg = new Pegawai[3];
19         System.out.println("Konstruktor perusahaan dijalankan...");
20     }
21     //Relasi agregasi antara pegawai dan perusahaan
22     public void insertPegawai(Pegawai p) {
23         peg[counter] = p;
24         counter++;
25     }
26
27     public void tampilPer() {
28         System.out.println("Perusahaan " + namaPer + " memiliki pegawai: ");
29         for (int i = 0; i < counter; i++) {
30             peg[i].tampilPeg();
31         }
32     }
33 }
```

Pegawai.Java

Source Code

```
5 package GUIDED2;
6 //Fatimah Az Zahra_2211102160_IF10K
7 // Class Pegawai
8
9 class Pegawai {
10
11     private String nama;
12     private String NIP;
13
14     public Pegawai() {
15         System.out.println("Konstruktor pegawai dijalankan...");
16     }
17
18     public Pegawai(String NIP, String nama) {
19         this.NIP = NIP;
20         this.nama = nama;
21         System.out.println("Konstruktor pegawai dijalankan...");
22     }
23
24     public void tampilPeg() {
25         System.out.println("NIP : " + NIP + " ,Nama : " + nama);
26     }
27 }
```

Relasi.Java

Source Code

```
5 package GUIDED2;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public class Relasi {
9
10     public static void main(String[] args) {
11         Perusahaan Per = new Perusahaan("Nusantara Jaya");
12         Pegawai Peg1, Peg2, Peg3;
13         Peg1 = new Pegawai("P001", "Rudi");
14         Peg2 = new Pegawai("P002", "Toni");
15         Peg3 = new Pegawai("P003", "Ani");
16         //Relasi antara Class Pegawai dengan Class Perusahaan
17         Per.insertPegawai(Peg1);
18         Per.insertPegawai(Peg2);
19         Per.insertPegawai(Peg3);
20         System.out.println();
21         Per.tampilPer();
22     }
23 }
```

Output:

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul7 ---
Konstruktor perusahaan dijalankan...
Konstruktor pegawai dijalankan...
Konstruktor pegawai dijalankan...
Konstruktor pegawai dijalankan...

Perusahaan Nusantara Jaya memiliki pegawai:
NIP : P001 ,Nama : Rudi
NIP : P002 ,Nama : Toni
NIP : P003 ,Nama : Ani
-----
BUILD SUCCESS
-----
Total time: 1.905 s
Finished at: 2024-05-06T10:40:39+07:00
-----
```

Penjelasan program:

Program di atas terdiri dari 2 kelas `Perusahaan` memiliki atribut `namaPer` sebagai nama perusahaan dan `peg` sebagai array objek dari kelas `Pegawai`, yang merepresentasikan hubungan agregasi antara pegawai dan perusahaan. Konstruktor kelas `Perusahaan` digunakan untuk menginisialisasi atribut dan array objek `peg`. Metode `insertPegawai()` digunakan untuk menambahkan objek pegawai ke dalam array `peg`, dan `tampilPer()` digunakan untuk menampilkan informasi perusahaan beserta pegawai-pegawai yang bekerja di dalamnya.

Kelas `Pegawai` memiliki atribut `nama` dan `NIP` (Nomor Induk Pegawai). Terdapat dua konstruktor di kelas `Pegawai`, salah satunya untuk menginisialisasi atribut `NIP` dan `nama`. Metode `tampilPeg()` digunakan untuk menampilkan informasi pegawai.

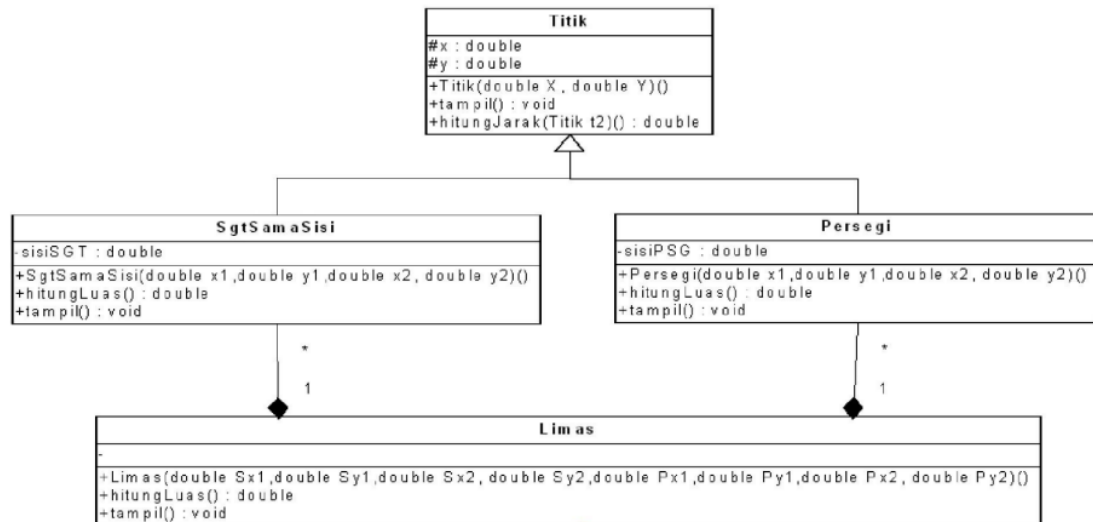
Pada kelas `Relasi`, objek `Perusahaan` dan tiga objek `Pegawai` diciptakan. Kemudian, objek-objek pegawai tersebut dimasukkan ke dalam perusahaan menggunakan metode `insertPegawai()`. Selanjutnya, informasi perusahaan beserta pegawai-pegawainya ditampilkan menggunakan metode `tampilPer()`. Relasi yang terbentuk adalah agregasi antara objek `Pegawai` dan `Perusahaan`, di mana objek perusahaan memiliki objek pegawai sebagai bagian darinya. Program ini menggambarkan bagaimana hubungan antara kelas-kelas diimplementasikan dalam pemrograman berorientasi objek, khususnya dalam konteks agregasi antara perusahaan dan pegawai.

B. Unguided

1. Unguided1

Soal:

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



Kelas Titik memiliki spesifikasi sebagai berikut :

- Memiliki atribut garis x dan y bertipe double (5%)
- Memiliki method tampil() untuk menampilkan nilai atribut kelas Titik (10%)
- Memiliki method hitungJarak(Titik t2) untuk menampilkan nilai atribut kelas Titik (10%)

Kelas SgtSamaSisi memiliki spesifikasi sebagai berikut :

- Memiliki atribut sisiSGT bertipe double (5%).
sisiSGT didapatkan dari perhitungan jarak 2 titik yang diinputkan melalui konstruktornya.
- Memiliki method hitungLuas() untuk menghitung luas segitiga (10%)
- Memiliki method tampil() yang meredefinisi method kelas induk untuk menampilkan nilai atribut kelas induk dan nilai atribut kelas SgtSamaSisi (10%)

Kelas Persegi memiliki spesifikasi sebagai berikut :

- Memiliki atribut sisiPSG bertipe double (5%)
sisiPSG didapatkan dari perhitungan jarak 2 titik yang diinputkan melalui konstruktornya.
- Memiliki method hitungLuas() untuk menghitung luas persegi (10%)
- Memiliki method tampil() yang meredefinisi method kelas induk untuk menampilkan nilai atribut kelas induk dan nilai atribut kelas Persegi (10%)

Kelas Limas memiliki spesifikasi sebagai berikut :

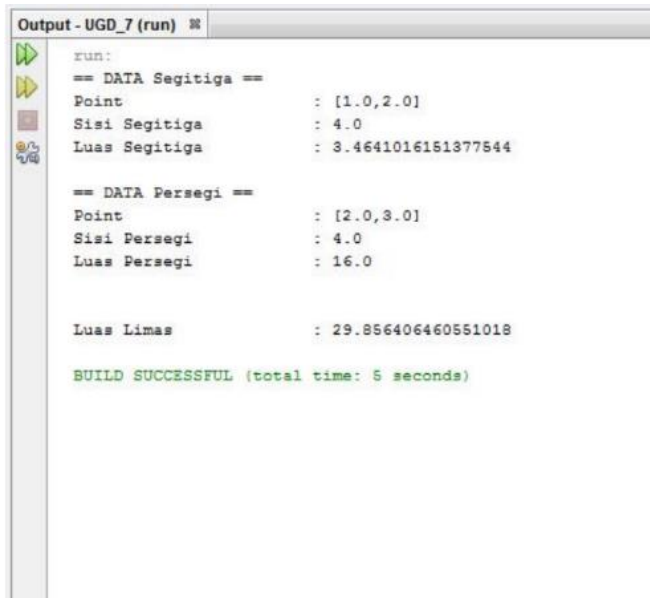
- Limas terdiri dari 4 buah SgtSamaSisi dengan ukuran yang sama dan 1 buah persegi (5%)
- Memiliki method hitungLuas() untuk menghitung luas Limas. (10%)
- Memiliki method tampil() untuk menampilkan nilai atribut SgtSamaSisi, nilai atribut Persegi dan Luas Permukaan Limas (10%)

Ketentuan Program :

1. Program diimplementasikan dalam Java dengan menggunakan prinsip Relasi

Kelas.

2. Cara Input Data → Melalui konsturktornya
Limas P = new Limas(1.0,2.0,5.0,2.0,2.0,3.0,2.0,7.0);
3. Rumus : Diasumsikan Titik1(x1,y1) dan Titik2(x2,y2)
 - ✓ Jarak 2 Titik : $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
 - ✓ Luas Persegí : *sisiPSG * sisi PSG*
 - ✓ Luas Segitiga Sama Sisi : $0.5 * sisiSGT * \sqrt{3}$
 - ✓ Luas Limas : $(4 * Luas Segitiga) + Luas Persegi$
4. Perhatikan Pengujian Unguidednya !! Unguided yang berhasil harus dapat dieksekusi sehingga tertampil seperti tampilan berikut ini.



```
run:
== DATA Segitiga ==
Point           : [1.0,2.0]
Sisi Segitiga    : 4.0
Luas Segitiga    : 3.4641016151377544

== DATA Persegi ==
Point           : [2.0,3.0]
Sisi Persegi     : 4.0
Luas Persegi     : 16.0

Luas Limas       : 29.856406460551018

BUILD SUCCESSFUL (total time: 5 seconds)
```

5. Sertakan comment yang terdiri dari NAMA dan NIM di bagian atas program Anda

Titik.Java

Source code

```
5 package UNGUIDED1;
6 //Fatimah Az Zahra - 2211102160 - IF10K
7
8
9 class Titik {
10     // Variabel anggota kelas untuk menyimpan koordinat titik.
11     double x, y;
12
13     // Konstruktor untuk kelas Titik, menerima koordinat x dan y sebagai parameter.
14     Titik(double x, double y) {
15         this.x = x;
16         this.y = y;
17     }
18
19     // Metode untuk menampilkan koordinat titik ke layar.
20     void tampil() {
21         System.out.println("Point \t\t: [" + x + ", " + y + "]");
22     }
23
24     double hitungJarak(Titik t2) {
25         return Math.sqrt(Math.pow((t2.x - x), 2) + Math.pow((t2.y - y), 2));
26     }
27 }
```


SgtSamaSisi.Java

Source Code

```
5 package UNGUIDED1;
6 //Fatimah Az Zahra - 2211102160 - IF10K
7
8 import java.lang.Math;
9
10 // Deklarasi kelas SgtSamaSisi yang merupakan turunan dari kelas Titik.
11 class SgtSamaSisi extends Titik {
12
13     // Variabel untuk menyimpan panjang sisi segitiga.
14     double sisiSGT;
15
16     // Konstruktor kelas SgtSamaSisi, menerima koordinat dua titik untuk menghitung sisi segitiga.
17     SgtSamaSisi(double x1, double y1, double x2, double y2) {
18         // Memanggil konstruktor kelas induk (Titik) dengan koordinat titik pertama (x1, y1).
19         super(x1, y1);
20         // Menghitung sisi segitiga
21         this.sisiSGT = super.hitungJarak(new Titik(x2, y2));
22     }
23
24     // Metode untuk menghitung luas segitiga.
25     double hitungLuas() {
26         // Menggunakan rumus luas segitiga equilateral: luas = 0.5 * sisi * sqrt(3).
27         return 0.5 * sisiSGT * Math.sqrt(3);
28     }
29
30     // Metode untuk menampilkan informasi titik dan sisi segitiga ke layar.
31     @Override
32     void tampil() {
33         // Memanggil metode tampil() dari kelas induk (Titik) untuk menampilkan koordinat titik.
34         super.tampil();
35         // Menampilkan panjang sisi segitiga ke layar.
36         System.out.println("Sisi Segitiga \t: " + sisiSGT);
37     }
38 }
```

Persegi.Java

Source Code

```
5 package UNGUIDED1;
6 //Fatimah Az Zahra - 2211102160 - IF10K
7 // Deklarasi kelas Persegi yang merupakan turunan dari kelas Titik.
8 class Persegi extends Titik {
9
10     // Variabel untuk menyimpan panjang sisi persegi.
11     double sisiPSG;
12
13     // Konstruktor kelas Persegi, menerima koordinat dua titik untuk menghitung sisi persegi.
14     Persegi(double x1, double y1, double x2, double y2) {
15         // Memanggil konstruktor kelas induk (Titik) dengan koordinat titik pertama (x1, y1).
16         super(x1, y1);
17         // Menghitung sisi persegi
18         this.sisiPSG = super.hitungJarak(new Titik(x2, y2));
19     }
20
21     // Metode untuk menghitung luas persegi.
22     double hitungLuas() {
23         // Menggunakan rumus luas persegi: luas = sisi * sisi.
24         return (sisiPSG * sisiPSG);
25     }
26
27     // Metode untuk menampilkan informasi titik dan sisi persegi ke layar.
28     @Override
29     void tampil() {
30         // Memanggil metode tampil() dari kelas induk (Titik) untuk menampilkan koordinat titik.
31         super.tampil();
32         // Menampilkan panjang sisi persegi ke layar.
33         System.out.println("Sisi Persegi \t: " + sisiPSG);
34     }
35 }
```

Limas.Java

Source Code

```
5 package UNGUIDED1;
6 //Fatimah Az Zahra - 2211102160 - IF10K
7 // Deklarasi kelas Limas.
8 class Limas {
9     // Variabel untuk menyimpan objek segitiga dan persegi.
10    SgtSamaSisi segitiga;
11    Persegi persegi;
12
13    // Konstruktor kelas Limas, menerima koordinat untuk dua titik segitiga dan dua titik persegi.
14    Limas(double Sx1, double Syl, double Sx2, double Sy2, double Px1, double Px2, double Py2) {
15        // Membuat objek segitiga dengan menggunakan konstruktor kelas SgtSamaSisi.
16        segitiga = new SgtSamaSisi(Sx1, Syl, Sx2, Sy2);
17        // Membuat objek persegi dengan menggunakan konstruktor kelas Persegi.
18        persegi = new Persegi(Px1, Py2, Px2, Py2);
19    }
20
21    // Metode untuk menghitung luas limas.
22    double hitungLuas() {
23        // Menghitung luas segitiga dari objek segitiga.
24        double luasSegitiga = segitiga.hitungLuas();
25        // Menghitung luas persegi dari objek persegi.
26        double luasPersegi = persegi.hitungLuas();
27        // Menggunakan rumus luas limas: 4 * luas segitiga + luas persegi.
28        return (4 * luasSegitiga) + luasPersegi;
29    }
30
31    // Metode untuk menampilkan informasi segitiga, persegi, dan luas limas ke layar.
32    void tampil() {
33        // Menampilkan header dan data segitiga.
34        System.out.println("== DATA Segitiga ==");
35        segitiga.tampil();
36        System.out.printf("Luas Segitiga \t: %.7f\n\n", segitiga.hitungLuas());
37        // Menampilkan header dan data persegi.
38        System.out.println("== DATA Persegi ==");
39        persegi.tampil();
40        System.out.printf("Luas Persegi \t: %.7f\n\n", persegi.hitungLuas());
41        // Menampilkan luas limas.
42        System.out.printf("Luas Limas \t: %.7f\n", hitungLuas());
43    }
44 }
```

Main.Java

Source Code

```
5 package UNGUIDED1;
6 //Fatimah Az Zahra - 2211102160 - IF10K
7 public class Main {
8     public static void main(String[] args) {
9         // Deklarasi dan inisialisasi variabel nama dengan nilai "FATIMAH AZ AHRA".
10        String nama = "FATIMAH AZ AHRA";
11        // Deklarasi dan inisialisasi variabel NIM dengan nilai "2211102160".
12        String NIM = "2211102160";
13
14        // Menampilkan informasi nama ke layar.
15        System.out.println("NAMA \t: " + nama); // Menampilkan nama di atas
16        // Menampilkan informasi NIM ke layar.
17        System.out.println("NIM \t: " + NIM);
18
19        // Membuat objek Limas P dengan menggunakan konstruktor Limas.
20        // Titik-titik yang digunakan untuk membentuk limas, segitiga, dan persegi.
21        Limas P = new Limas(1.0, 2.0, 5.0, 2.0, 6.0, 2.0, 7.0);
22
23        // metode tampil() dari objek P
24        P.tampil();
25    }
26 }
```

Output :

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul7 ---
NAMA      : FATIMAH AZ AHRA
NIM       : 2211102160
== DATA Segitiga ==
Point     : [1.0, 2.0]
Sisi Segitiga : 4.0
Luas Segitiga : 3.4641016

== DATA Persegi ==
Point     : [6.0, 7.0]
Sisi Persegi : 4.0
Luas Persegi : 16.0000000

Luas Limas : 29.8564065

-----
BUILD SUCCESS
-----

Total time: 3.447 s
Finished at: 2024-05-06T19:56:12+07:00
-----
```

Penjelasan Program:

Program Java tersebut merupakan program yang menghitung luas limas. Dalam program ini, terdapat beberapa kelas yang saling terkait, yakni kelas `Titik`, `SgtSamaSisi`, `Persegi`, `Limas`, dan `Main`. Kelas `Titik` digunakan untuk merepresentasikan titik dalam koordinat. Kelas `SgtSamaSisi` dan `Persegi` bertindak sebagai turunan dari kelas `Titik`, yang menghitung luas segitiga dan persegi. Kelas `Limas` menggunakan objek dari kelas `SgtSamaSisi` dan `Persegi` untuk menghitung luas limas. Kelas `Main` merupakan kelas utama yang digunakan untuk menjalankan program. Dalam program ini, hubungan antar kelas terbentuk melalui konsep pewarisan dan asosiasi. Pewarisan terjadi antara kelas `SgtSamaSisi` dan `Persegi` dengan kelas `Titik`, di mana kelas anak mewarisi sifat dan metode dari kelas induk. Sedangkan asosiasi terjadi antara kelas `Limas` dengan kelas `SgtSamaSisi` dan `Persegi`, di mana objek dari kelas `SgtSamaSisi` dan `Persegi` digunakan dalam kelas `Limas` untuk menghitung luas limas.