PRAKTIKUM 2

Enkapsulasi dan Relasi Antar Objek

1. Tujuan

Tujuan praktikum ini adalah:

a. Mahasiswa dapat menerapkan konsep enkapsulasi data pada program.

b. Mahasiswa dapat menerapkan konsep enkapsulasi pada program.

c. Mahasiswa dapat memahami hubungan antar objek dan menerapkannya dalam program.

d. Mahasiswa dapat memahami interaksi antar objek dan menerapkannya dalam program.

2. Landasan Teori

2.1. Enkapsulasi

Enkapsulasi merupakan usaha untuk menyembunyikan detail implementasi dari objek, untuk membuat objek seindependen mungkin. Enkapsulasi dapat diwujudkan dengan data maupun *information hiding*.

Pada pemrograman Java, *information hiding* diwujudkan dengan membuat atribut memiliki *access modifier* `private`, sedangkan method yang mengakses atribut tersebut memiliki *access modifier* `public`.

Contoh class Titik pada praktikum 1 dapat dilengkapi dengan *access modifier* pada atribut dan method-nya menjadi seperti berikut:

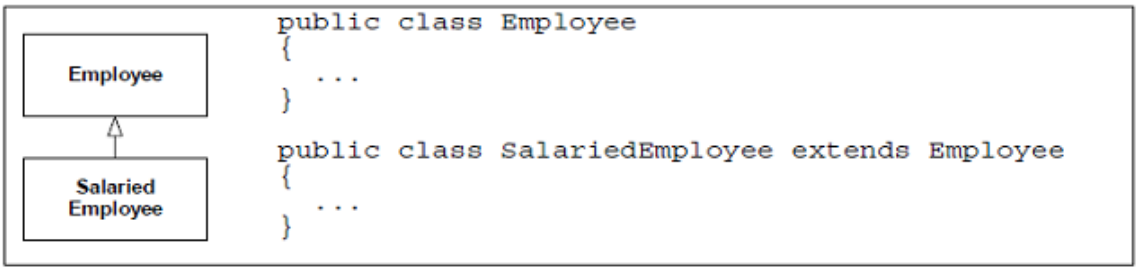
```
10  public class Titik {
11      //atribut
12      private double absis;
13      private double ordinat;
14      //static int numberOfPoints;
15
16      //konstruktor
17      //membuat objek titik dengan inisialisasi nilai absis dan ordinat
18      public Titik(double absis, double ordinat){
19          this.absis = absis;
20          this.ordinat = ordinat;
21      }
22
23      //membuat objek titik dengan inisialisasi absis 0 dan ordinat 0
24      public Titik(){
25          this(0,0);
26      }
```

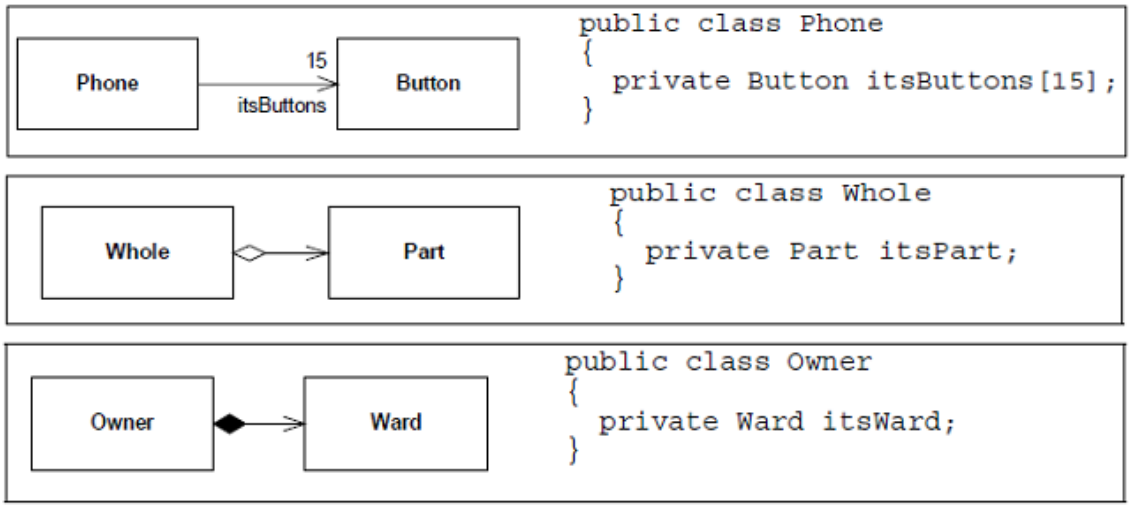
```
28          //method
29          //fungsi selektor untuk mendapatkan nilai atribut absis
30          public double getOrdinat(){
31              return this.ordinat;
32          }
33
34          //fungsi selektor untuk mendapatkan nilai atribut ordinat
35          public double getAbsis(){
36              return this.absis;
37          }
38
39          //prosedur untuk mengeset nilai atribut absis dengan nilai yang baru
40          public void setAbsis(double absis){
41              this.absis = absis;
42          }
43
44          //prosedur untuk mengeset nilai atribut ordinat dengan nilai yang baru
45          public void setOrdinat(double ordinat){
46              this.ordinat = ordinat;
47          }
48      }
```

2.2. Relasi Antar Objek

Pemrograman berorientasi obejk menggambarkan hubungan antar objek dan interaksinya. Diagram kelas digunakan untuk menggambarkan hubungan antar kelas atau objek. Terdapat beberapa hubungan antar objek seperti pewarisan, asosiasi, agregasi, komposisi, dan dependency. Khusus hubungan pewarisan tidak dibahas pada praktikum ini.

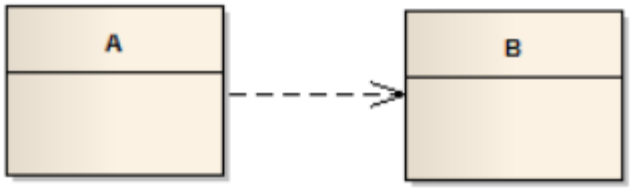
Representasi hubungan antar objek tersebut adalah sebagai berikut:





Gambar 1. Representasi hubungan antar objek, berturut-turut pewarisan, asosiasi, agregasi, dan komposisi (sumber : UML For Java Programmers, Robert Cecil Martin (2002))

Dependensi adalah hubungan yang menunjukkan bahwa sebuah objek bergantung terhadap objek lain. Contoh:



Gambar 2. Contoh Hubungan Dependency

3. Langkah Praktikum

3.1. Enkapsulasi

1. Implementasikan konsep enkapuslasi pada class titik yang telah dibuat pada praktikum 1.
2. Tambahkan beberapa method, antara lain:
    a. getJarakPusat() untuk menghitung jarak sebuah titik dengan titi pusat (0,0).
    b. refleksiX() melakukan pencerminan titik terhadap sumbu X.
    c. refleksiY() melakukan pencerminan titik terhadap sumbu Y.
    d. getRefleksiX() menghasilkan titik baru yang merupakan hasil pencerminan terhadap sumbu X.
    e. getRefleksiY() menghasilkan titik baru yang merupakan hasil pencerminan terhadap sumbu Y.

    (Perhatikan perbedaan antara method pada point b & c dengan d & e!)

3. Buatlah sebuah main class untuk membuat sejumlah objek titik. Lakukan eksperimen terhadap pemanggilan atribut dan method dari objek titik tersebut.
4. Lakukan pula eksperimen dengan mengubah accress modifier yang telah dibuat dari private dan public ataupun sebaliknya pada atribut, konstruktor, maupun method.
5. Tulislah kesimpulan dari hasil eksperimen yang didapatkan dengan kalimat sendiri denga jelas dan singkat!

3.2. Relasi Antar Objek

1. Buatlah sebuah kelas bernama Garis yang terdiri atas 2 buah atribut, yaitu titikAwal dan titikAkhir bertipe Titik (Garis memiliki hubungan agregasi dengan Titik).
2. Lengkapi method getter dan setter pada class Garis.
3. Tambahkan beberapa method lainnya, antara lain:
    a. getPanjang() menghitung Panjang garis.
    b. getGradien() menghitung gradien garis.
    c. getRefleksiY() menghasilkan garis baru yang merupakan hasil pencerminan dengan sumbu Y.
    d. isTegakLurus(G: Garis) menghasilkan nilai True jika objek garis tegak lurus dengan garis G.
4. Buatlah sebuah main class untuk membuat sejumlah objek garis. Lakukan eksperimen terhadap pemanggilan atribut dan method dari objek garis tersebut.

===== Selesai =====