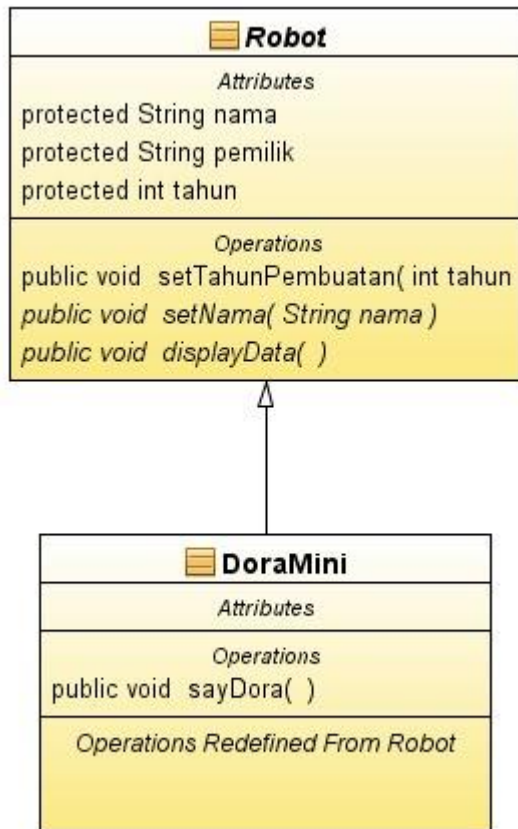


Nama : Ni Made Widiasanti
NIM : 19051397002
Prodi : D4 Manajemen Informatika 2019A
Mata Kuliah : Pemrograman Berorientasi Objek

Latihan Praktikum “Abstract Class”



1. Tugas:

1. Ubahlah class diagram di atas ke dalam **bentuk standar UML class diagram**
2. Buatlah 2 kelas berdasarkan diagram kelas di atas (beserta kelas Main).

Keterangan tambahan:

- a) Kelas Robot adalah kelas abstrak

- b) Method “sayDora” akan menampilkan “Halo, Saya Dora Mini” di layar
- c) Method displayData akan menampilkan setiap nilai dari atribut yang dimiliki ke layar
- d) Method setNama dan displayData adalah method abstract

Jawaban :

```
1  abstract class Robot
2
3      String nama;
4      String pemilik;
5      int tahun;
6
7      void setTahunpembuatan(int tahun)
8  {
9      System.out.println("Saya dibuat tahun " + tahun);
10 }
11
12 abstract void setNama(String nama);
13
14 abstract void displayData();
15 }
16
17 class DoraMini extends Robot
18 {
19     void sayDora()
20     {
21         System.out.println("Hi, saya Dora Mini");
22     }
23
24     @Override
25     void setNama(String nama)
26     {
27         System.out.println("Nama panggilan saya " + nama);
28     }
29
30     @Override
31     void displayData()
32     {
33         sayDora();
34         setNama("Dora");
35         setTahunpembuatan(2020);
36         System.out.print("\n");
37     }
38
39     Run | Debug
40     public static void main(String[] args) {
41         DoraMini DoraMini = new DoraMini();
42         DoraMini.displayData();
43     }
```

2. Seorang analyst membuat aplikasi simulasi permainan. Analyst membuat class diagram dengan abstract class sebagai berikut:

<i>Permainan</i>
-namaPemain : string -levelPemain : int
+setNamaPemain(in namaPemain : string) : void +setLevelPemain(in levelPemain : int) : void +getNamaPemain() : string +getLevelPemain() : int +jalankan() : void +hitungSkor(in hit : int, in miss : int) : int

Deskripsi:

- a) Atribut namaPemain dan levelPemain menyimpan nama dan level pemain.
- b) Nilai levelPemain berkisar 1-100 dengan ketentuan:
 - a. 1-20: normal
 - b. 21-80: medium
 - c. 81-100: hard
- c) Method jalankan() akan menjalankan skenario permainan (set nama dan level pemain, mengeluarkan data tersebut, dan menghitung skor pemain).
- d) Method hitungSkor merupakan abstract method.

Tugas:

- a) Buatlah kode berdasarkan abstract class di atas.
- b) Buatlah 2 classes lain yang menggunakan abstract class di atas dengan deskripsi:
 - a. “PermainanArcade” dengan aturan hitung skor: jumlah hit x 3 – jumlah miss x 1;
 - b. “PermainanStrategy” dengan aturan hitung skor: jumlah hit x 5;

Perhatikan bahwa algoritma hitungSkor ditentukan oleh subclass, bukan superclass.

- c) Buktikan bahwa abstract method memastikan bahwa method tersebut di-override oleh subclass.
- d) Buktikan bahwa objek dari abstract class tidak dapat dibentuk.

Jawaban :

```
1  abstract class Permainan
2  {
3      Scanner inputUser = new Scanner(System.in);
4      String namaPemain;
5      int levelPemain;
6
7      void setNamaPemain(String namaPemain)
8      {
9          System.out.print("Masukkan nama anda : ");
10         namaPemain = inputUser.nextLine();
11     }
12
13     void setLevelPemain(int levelPemain)
14     {
15         System.out.print("Pilih level anda : ");
16         levelPemain = inputUser.nextInt();
17         if (levelPemain >= 1 && levelPemain <=20 )
18         {
19             System.out.println("Level normal");
20         }
21         else if (levelPemain >= 21 && levelPemain <= 80)
22         {
23             System.out.println("Level medium");
24         }
25         else if (levelPemain >= 81 && levelPemain <= 100)
26         {
27             System.out.println("Level hard");
28         }
29         else
30         {
```

```

31         System.out.println("Level tidak tersedia!");
32         System.out.println("Level terendah : 1");
33         System.out.println("Level tertinggi : 100");
34     }
35 }
36
37 String getNamaPemain()
38 {
39     return namaPemain;
40 }
41 int getLevelPemain()
42 {
43     return levelPemain;
44 }
45
46 void jalankan()
47 {
48     setNamaPemain(namaPemain);
49     setLevelPemain(levelPemain);
50 }
51
52 abstract int hitungSkor(int hit, int miss);
53 }

```

```

54
55 class PermainanArcade extends Permainan
56 {
57     @Override
58     int hitungSkor(int hit, int miss)
59     {
60         int total = (hit*3) - (miss*1);
61         System.out.println("Maka, skor perolehan : " + total);
62         System.out.println("\n");
63         return total;
64     }
65 }
66
67 class PermainanStrategy extends Permainan
68 {
69     @Override
70     int hitungSkor(int hit, int miss)
71     {
72         int total = hit*5;
73         System.out.println("Maka, skor perolehan : " + total);
74         System.out.print("\n");
75         return total;
76     }
77 }
78

```

```

78
79 public class mainpermainan {
    Run | Debug
80     public static void main(String[] args) {
81         Scanner inputUser = new Scanner(System.in);
82         Permainan Arcade = new PermainanArcade();
83         Permainan Strategy = new PermainanStrategy();
84
85         System.out.println("--MODE PERMAINAN--");
86         System.out.println("1. Permainan Arcade");
87         System.out.println("2. Permainan Strategy");
88         System.out.print("Masukkan pilihan anda : ");
89         int pilmode = inputUser.nextInt();
90         if (pilmode == 1) {
91             Arcade.jalankan();
92             System.out.println("\n--PERMAINAN ARCADE--");
93             Arcade.hitungSkor(5, 3);
94         }
95         else if (pilmode == 2)
96         {
97             Strategy.jalankan();
98             System.out.println("\n--PERMAINAN STRATEGY--");
99             Strategy.hitungSkor(10, 5);
100        }
101        else
102        {
103            System.out.println("\nMode tidak tersedia!");
104        }
105    }
106

```

3. Tulislah dan jelaskan perbedaan antara abstract class dan interface!

Jawaban :

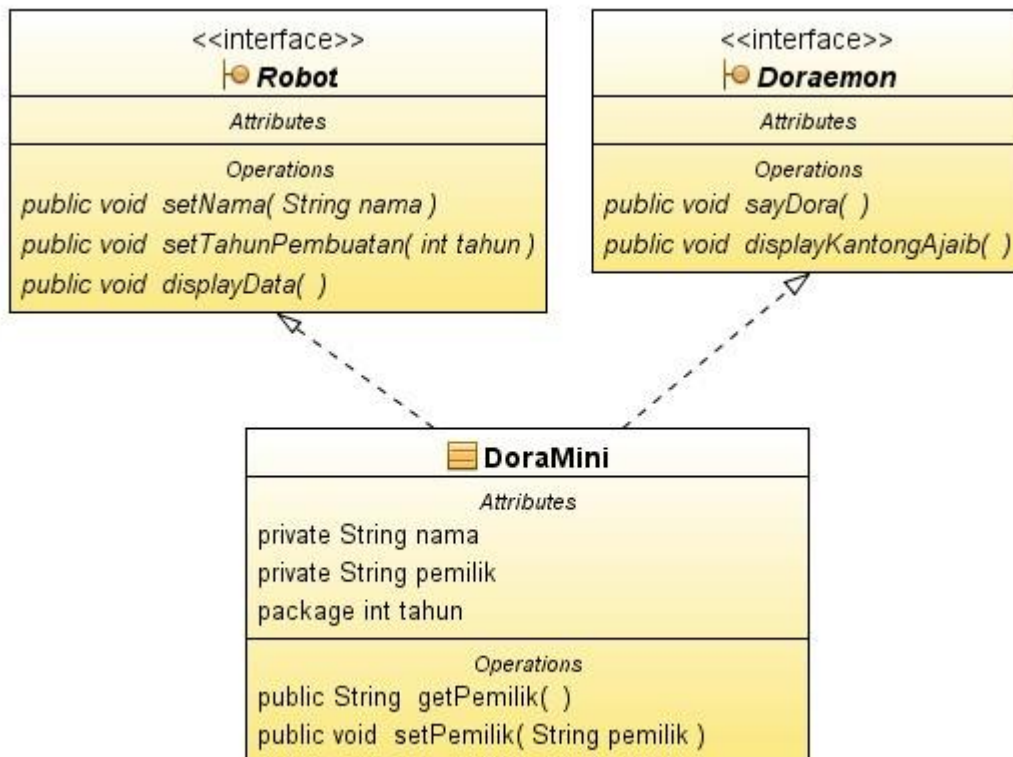
- **Abstract Class**

Abstract class adalah sebuah class dalam OOP yang tidak dapat diinstansiasi atau dibuat object-nya. Abstract class biasanya berisi fitur-fitur dari sebuah class yang belum implementasikan. Abstract class harus diimplementasikan melalui proses pewarisan, maka dalam abstract class berlaku aturan-aturan yang ada pada konsep pewarisan. Di dalam sebuah abstract class kita dapat membuat property dan method yang nantinya dapat digunakan oleh child class. Tentu saja property dan method yang dapat digunakan oleh child class adalah property dan method yang memiliki visibilitas protected dan public.

- **Interface**

Interface adalah sebuah class yang semua method-nya adalah abstract method. Karena semua method-nya adalah abstract method maka interface pun harus diimplementasikan oleh child class seperti halnya pada abstract class. Hanya saja bila kita sebelumnya menggunakan keyword extends untuk mengimplementasikan sebuah abstract class, maka pada interface kita menggunakan keyword implements untuk mengimplementasikan sebuah interface.

4.



Keterangan:

- Method “sayDora” akan menampilkan “Halo, Saya Dora Mini” di layar
- Method “dispKantongAjaib” akan menampilkan “Saya juga seperti Doraemon yang memiliki kantung ajaib”
- Method displayData akan menampilkan setiap nilai dari atribut yang dimiliki ke layar

Tugas: Buatlah 3 kelas berdasarkan diagram kelas di atas (dan juga class Main untuk membentuk objeknya).

Jawaban :

```
1  interface Robot
2  {
3      void setNama(String nama);
4      void setTahun(int tahun);
5      void displayData();
6  }
7
8  interface Doraemon
9  {
10     void sayDora();
11     void displayKantongAjaib();
12 }
13 class DoraMini implements Robot , Doraemon
14 {
15     String pemilik;
16     String nama;
17     int tahun;
18
19     String getPemilik()
20     {
21         return pemilik;
22     }
23
24     void setPemilik(String pemilik)
25     {
26         System.out.println("Pemilik Dora Mini : " + pemilik);
27     }
28 }
```



```

29     @Override
30     public void setName(String nama)
31     {
32         System.out.println("Nama : " + nama);
33     }
34
35     @Override
36     public void setTahun(int tahun)
37     {
38         System.out.println("Dibuat tahun : " + tahun);
39     }
40
41     @Override
42     public void displayData()
43     {
44         setName("DoraMini");
45         setTahun(2020);
46         setPemilik("Ni Made Wideasanti.\n");
47         System.out.println("--DoraMini Berkata--");
48         sayDora();
49         displayKantongAjaib();
50     }
51
52     @Override
53     public void sayDora()
54     {
55         System.out.println("Halo, Saya Dora Mini");
56     }
57
58     @Override
59     public void displayKantongAjaib()
60     {
61         System.out.println("Saya juga seperti Doraemon yang memiliki kantung ajaib\n");
62     }
63
64     Run | Debug
65     public static void main(String[] args)
66     {
67         DoraMini doraMini = new DoraMini();
68         doraMini.displayData();
69     }

```