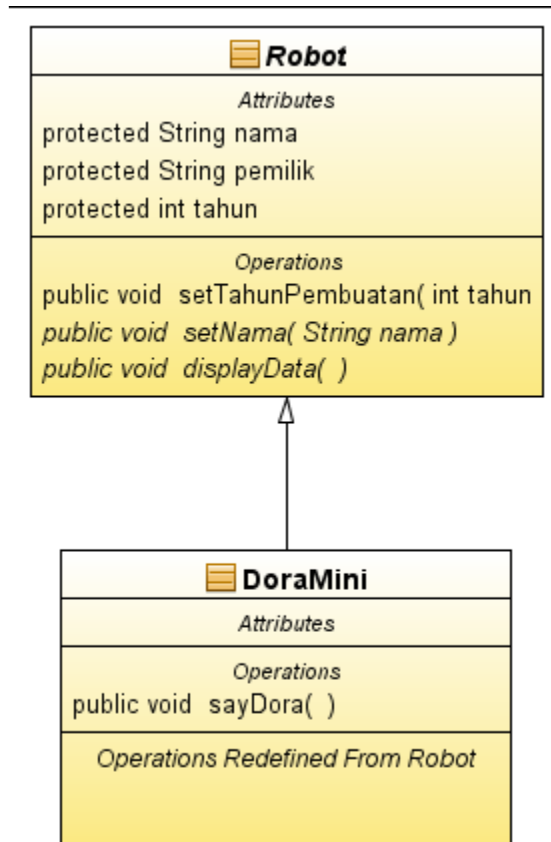


Nama : Annisa Rusydina Sabila  
NIM : 190513097004  
Kelas : D4 Manajemen Informatika 2019 A

### Latihan Praktikum “Abstract Class”



### 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 setName dan displayData adalah method abstract

Jawaban :

```
3  abstract class robotdora
4  {
5      String nama;
6      String pemilik;
7      int tahun;
8
9      void setTahunpembuatan(int tahun)
10     {
11         System.out.println("Saya dibuat tahun " + tahun);
12     }
13
14     abstract void setNama(String nama);
15
16     abstract void displayData();
17 }
18
19 class DoraMini extends robotdora
20 {
21     void sayDora()
22     {
23         System.out.println("Hi, saya Dora Mini");
24     }
25
26     @Override
27     void setNama(String nama)
28     {
29         System.out.println("Nama panggilan saya " + nama);
30     }
31 }
```

```
32     @Override
33     void displayData()
34     {
35         sayDora();
36         setNama("Dora");
37         setTahunpembuatan(2020);
38         System.out.print("\n");
39     }
40
41     Run | Debug
42     public static void main(String[] args) {
43         DoraMini doraMini = new DoraMini();
44         doraMini.displayData();
45     }
```

2. Seorang analyst membuat aplikasi simulasi permainan. Analyst membuat class diagram dengan

Abstract class sebagai berikut:

<b><i>Permainan</i></b>
-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:

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

Tugas:

- Buatlah kode berdasarkan abstract class di atas.
- Buatlah 2 classes lain yang menggunakan abstract class di atas dengan deskripsi:
  - "PermainanArcade" dengan aturan hitung skor: jumlah hit x 3 – jumlah miss x 1;
  - "PermainanStrategy" dengan aturan hitung skor: jumlah hit x 5;Perhatikan bahwa algoritma hitungSkor ditentukan oleh subclass, bukan superclass.
- Buktikan bahwa abstract method memastikan bahwa method tersebut di-override oleh subclass.
- Buktikan bahwa objek dari abstract class tidak dapat dibentuk.

Jawaban :

```
import java.util.*;

abstract class Permainan
{
    Scanner inputUser = new Scanner(System.in);
    String namaPemain;
    int levelPemain;

    void setNamaPemain(String namaPemain)
    {
        System.out.print("Masukkan nama anda : ");
        namaPemain = inputUser.nextLine();
    }
}
```

```
void setLevelPemain(int levelPemain)
{
    System.out.print("Pilih level anda : ");
    levelPemain = inputUser.nextInt();
    if (levelPemain >= 1 && levelPemain <= 20 )
    {
        System.out.println("Level normal");
    }
    else if (levelPemain >= 21 && levelPemain <= 80)
    {
        System.out.println("Level medium");
    }
    else if (levelPemain >= 81 && levelPemain <= 100)
    {
        System.out.println("Level hard");
    }
    else
    {
        System.out.println("Level tidak tersedia!");
        System.out.println("Level terendah : 1");
        System.out.println("Level tertinggi : 100");
    }
}
```

```

class PermainanArcade extends Permainan
{
    @Override
    int hitungSkor(int hit, int miss)
    {
        int total = (hit*3) - (miss*1);
        System.out.println("Maka, skor perolehan : " + total);
        System.out.println("\n");
        return total;
    }
}

class PermainanStrategy extends Permainan
{
    @Override
    int hitungSkor(int hit, int miss)
    {
        int total = hit*5;
        System.out.println("Maka, skor perolehan : " + total);
        System.out.print("\n");
        return total;
    }
}

```

```

public class mainpermainan {
    Run | Debug
    public static void main(String[] args) {
        Scanner inputUser = new Scanner(System.in);
        Permainan Arcade = new PermainanArcade();
        Permainan Strategy = new PermainanStrategy();

        System.out.println("--MODE PERMAINAN--");
        System.out.println("1. Permainan Arcade");
        System.out.println("2. Permainan Strategy");
        System.out.print("Masukkan pilihan anda : ");
        int pilmode = inputUser.nextInt();
        if (pilmode == 1) {
            Arcade.jalankan();
            System.out.println("\n--PERMAINAN ARCADE--");
            Arcade.hitungSkor(5, 3);
        }
        else if (pilmode == 2)
        {
            Strategy.jalankan();
            System.out.println("\n--PERMAINAN STRATEGY--");
            Strategy.hitungSkor(10, 5);
        }
        else
        {
            System.out.println("\nMode tidak tersedia!");
        }
    }
}

```

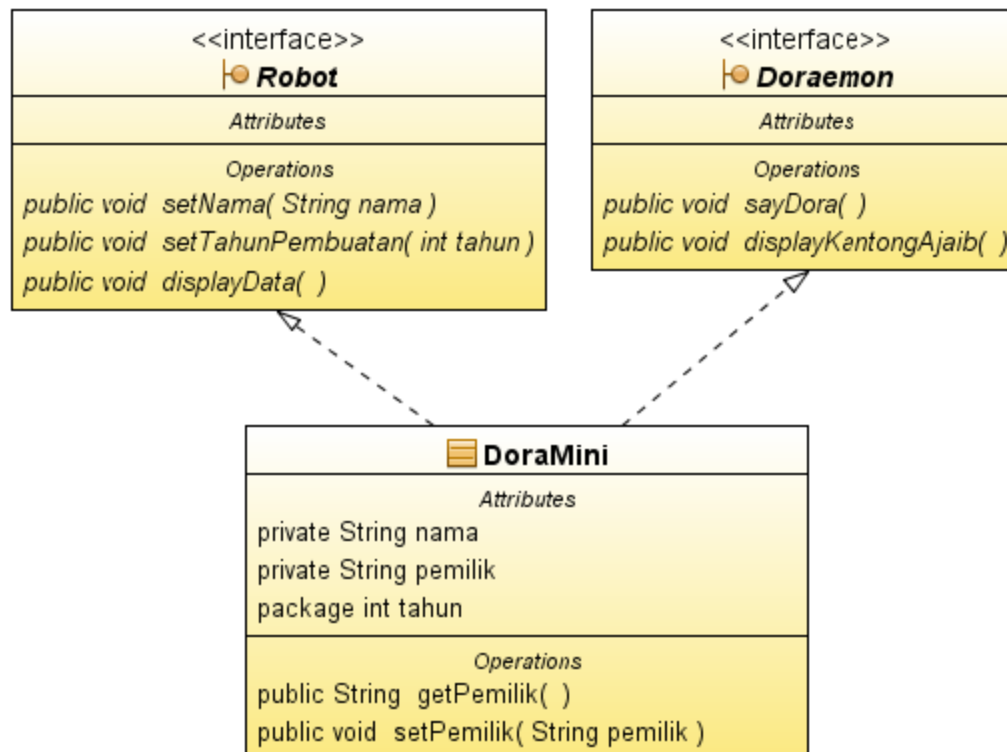
3. Tulislah dan jelaskan perbedaan antara abstract class dan interface

**Jawaban :**

Perbedaan antara abstract class dan interface. Saat deklarasi abstract class dan interface digunakan modifier public. Hal ini bertujuan agar abstract class dan interface tersebut dapat digunakan secara bebas oleh class lain yang membutuhkannya. Sekilas interface tampak mirip dengan abstract class karena abstract class juga menentukan bentuk method untuk subclass, tetapi ada beberapa perbedaan antara interface dan abstract class, yaitu:

- a. Abstract class dapat mengandung abstract method maupun method tidak abstrak. Sedangkan interface hanya boleh mengandung abstract method.
- b. Abstract class dapat mendeklarasikan variabel instant. Sedangkan interface hanya dapat mendeklarasikan konstanta.
- c. Abstract class digunakan oleh class lain melalui pewarisan dengan kata kunci extends. Sedangkan sebuah interface diimplementasikan ke dalam suatu class dengan menggunakan kata kunci implements.

4.



Keterangan:

- a) Method "sayDora" akan menampilkan "Halo, Saya Dora Mini" di layar
- b) Method "dispKantongAjaib" akan menampilkan "Saya juga seperti Doraemon yang memiliki kantung ajaib"
- c) Method displayData akan menampilkan setiap nilai dari atribut yang dimiliki ke layar

Jawaban :

```
package abstractdaninterface;

interface robot
{
    void setNama(String nama);
    void setTahun(int tahun);
    void displayData();
}

interface Doraemon
{
    void sayDora();
    void displayKantongAjaib();
}
```

```
class DoraMini implements robot , Doraemon
{
    String pemilik;
    String nama;
    int tahun;

    String getPemilik()
    {
        return pemilik;
    }

    void setPemilik(String pemilik)
    {
        System.out.println("Pemilik Dora Mini : " + pemilik);
    }

    @Override
    public void setNama(String nama)
    {
        System.out.println("Nama : " + nama);
    }

    @Override
    public void setTahun(int tahun)
    {
        System.out.println("ahun Pembuatan : " + tahun);
    }
}
```

```

@Override
public void setName(String nama)
{
    System.out.println("Nama : " + nama);
}

@Override
public void setTahun(int tahun)
{
    System.out.println("ahun Pembuatan : " + tahun);
}

@Override
public void displayData()
{
    setName("DoraMini");
    setTahun(2020);
    setPemilik("Annisa Rusydina Sabila.\n");
    System.out.println("--DoraMini Berkata--");
    sayDora();
    displayKantongAjaib();
}

@Override
public void sayDora()
{
    System.out.println("Halo, Saya Dora Mini");
}

```

```

@Override
public void sayDora()
{
    System.out.println("Halo, Saya Dora Mini");
}

@Override
public void displayKantongAjaib()
{
    System.out.println("Saya juga seperti Doraemon yang memiliki kantung ajaib\n");
}

Run | Debug
public static void main(String[] args)
{

```



