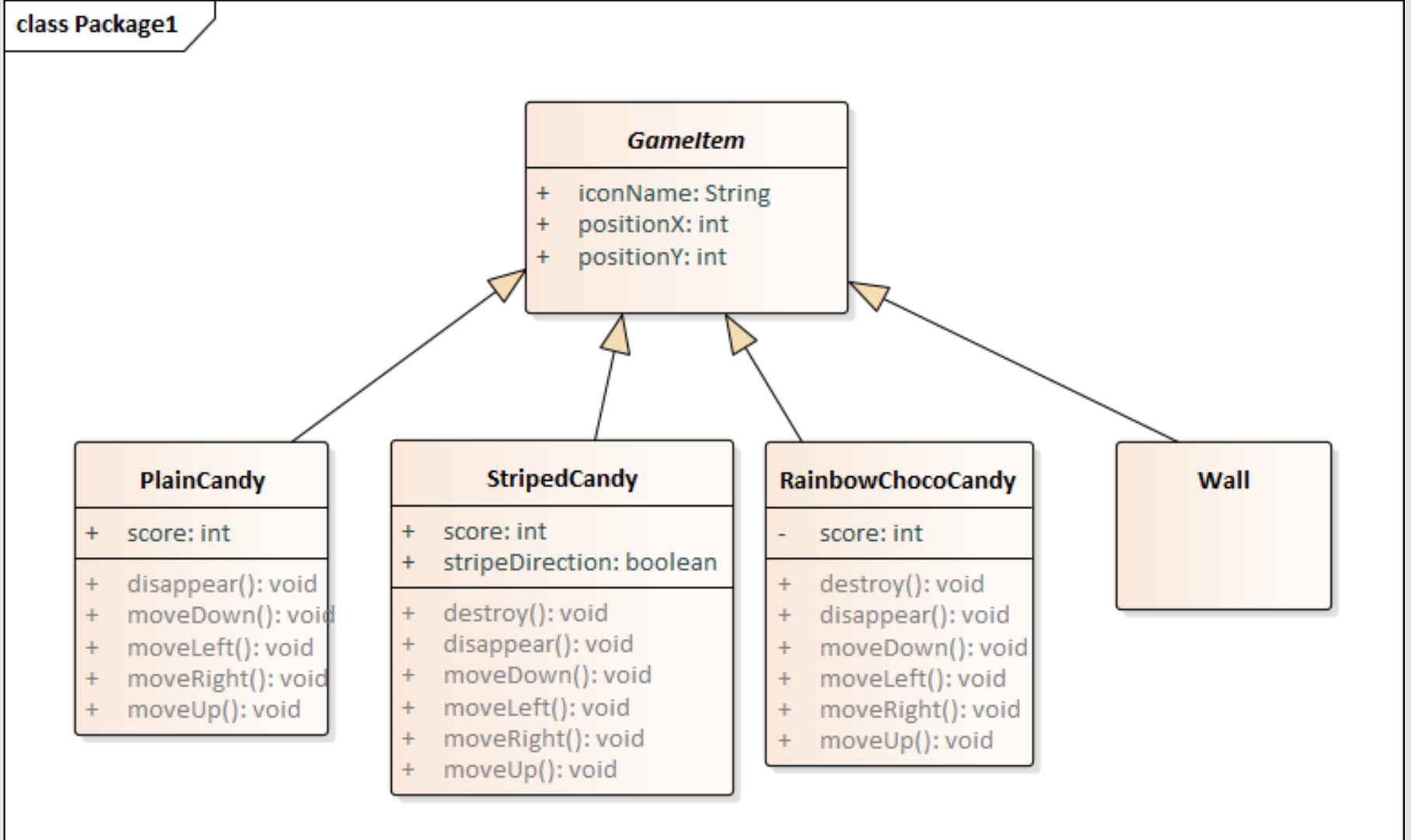


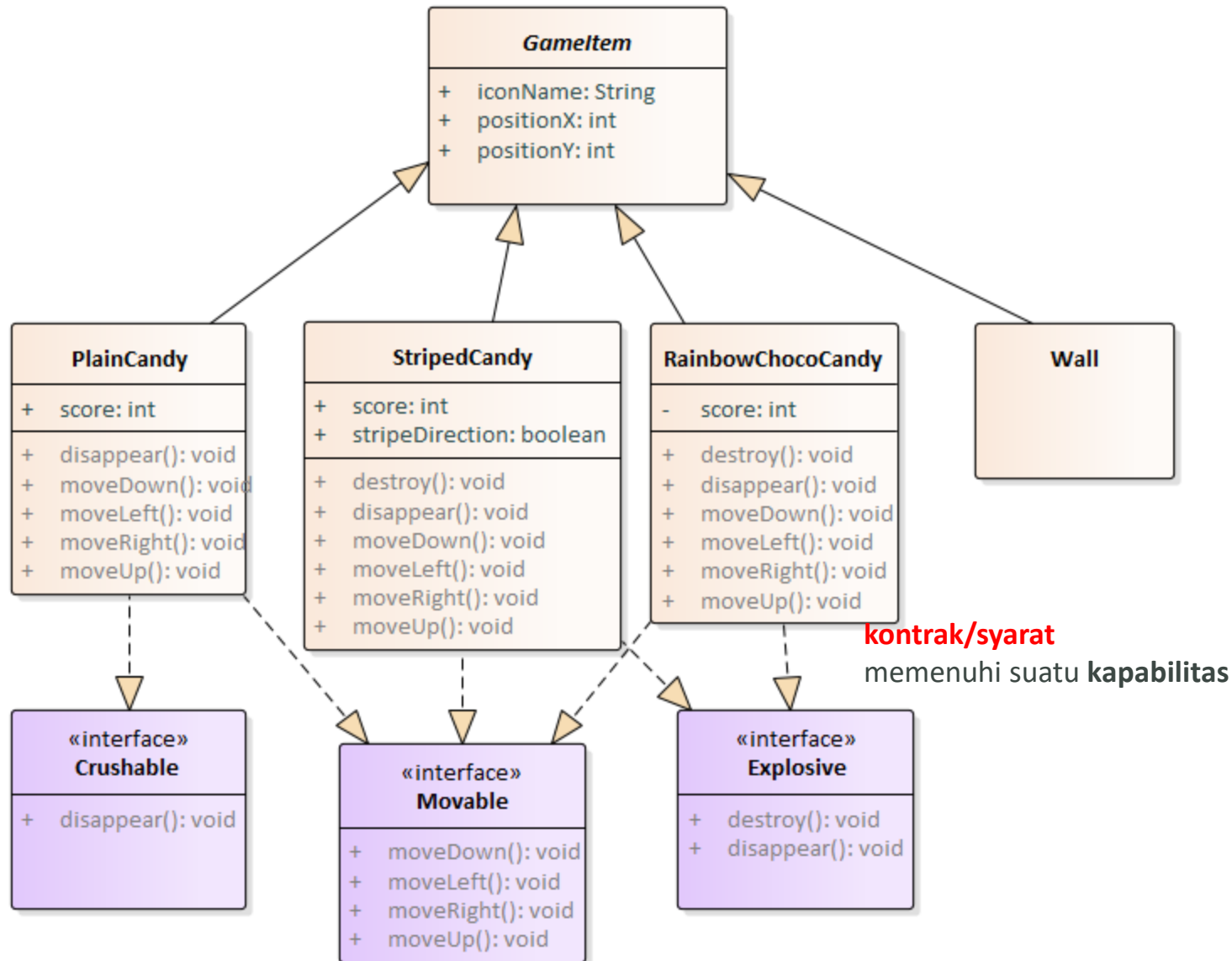
The background is a dark, textured surface resembling a chalkboard. It features various faint, light-colored sketches. In the upper left, there's a large 'V' and a globe. Below the globe is a detailed drawing of a microscope. In the lower left, there's a stack of books. In the lower right, there are mathematical symbols including a percentage sign, a plus sign, and a less-than sign. The overall theme is academic and scientific.

Interface

Tim Ajar PBO – JTI Polinema



class Package1



Definisi Interface

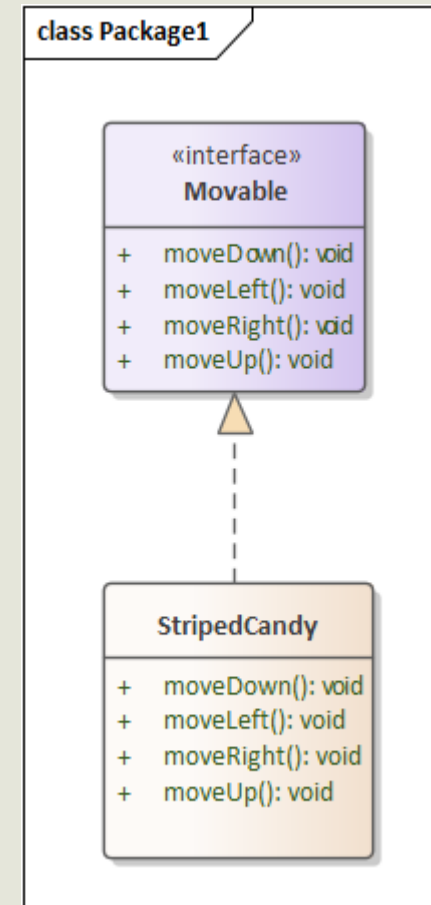
- Interface adalah sekumpulan method tanpa *body* yang saling berkaitan
- Karakteristik:
 - Umumnya hanya terdiri dari *abstract methods*
 - **Selalu** dideklarasikan dengan menggunakan *keyword* '**interface**'
 - Diimplementasikan dengan menggunakan *keyword* '**implements**'
 - Tidak memiliki constructor dan tidak dapat diinstansiasi. Yang dapat diinstansiasi adalah class yang meng-***implement*** interface tersebut
 - Dapat memiliki atribut tetapi hanya bersifat **public static final**

Kegunaan Interface

- Bertindak sebagai **kontrak/syarat** yang berisi sekumpulan *behavior/method* yang saling **terkait** untuk memenuhi suatu **kapabilitas**
- Dengan kata lain, interface memberikan panduan mengenai method apa saja yang perlu diimplementasikan untuk memenuhi kapabilitas tertentu

Notasi Class Diagram

- Nama interface **tidak** dicetak miring
- Keterangan <<interface>> di atas nama interface
- Nama method dicetak miring atau tidak
- **Implements** dilambangkan dengan garis panah putus-putus



Aturan Penulisan Interface

- Secara struktur hampir sama dengan class.
- Beberapa aturan:
 - **Tidak** memiliki concrete method (method biasa yang bukan abstract)
 - **Tidak** memiliki constructor
 - Dapat memiliki atribut, tapi hanya dapat bersifat **public, static, dan final**

Sintaks Interface

- Untuk mendeklarasikan suatu interface:
 - **public interface** <NamaInterface>
- Untuk mengimplementasikan interface:
 - **public class** <NamaClass> **implements** <NamaInterface>
- Nama interface sebaiknya dalam bentuk **adjective/kata sifat** jika merepresentasikan **kapabilitas**. Dapat juga menggunakan **kata benda**
- Contoh:

```
public interface Movable {  
    void moveLeft();  
    void moveRight();  
    void moveUp();  
    void moveDown();  
}
```

```
public class PlainCandy extends GameItem implements Movable{  
    @Override  
    public void moveLeft() {}  
    @Override  
    public void moveRight() {}  
    @Override  
    public void moveUp() {}  
    @Override  
    public void moveDown() {}  
}
```


Implementasi Interface

- Bila sebuah class mengimplementasikan suatu interface:
 - **Seluruh variable** dari interface akan dapat diakses oleh class tersebut
 - **Seluruh method** pada interface **harus diimplementasikan**
 - Bila class yang meng-implement interface **tidak mengimplementasikan semua method**, maka class tersebut harus dideklarasikan sebagai **abstract class**

Multiple Interface

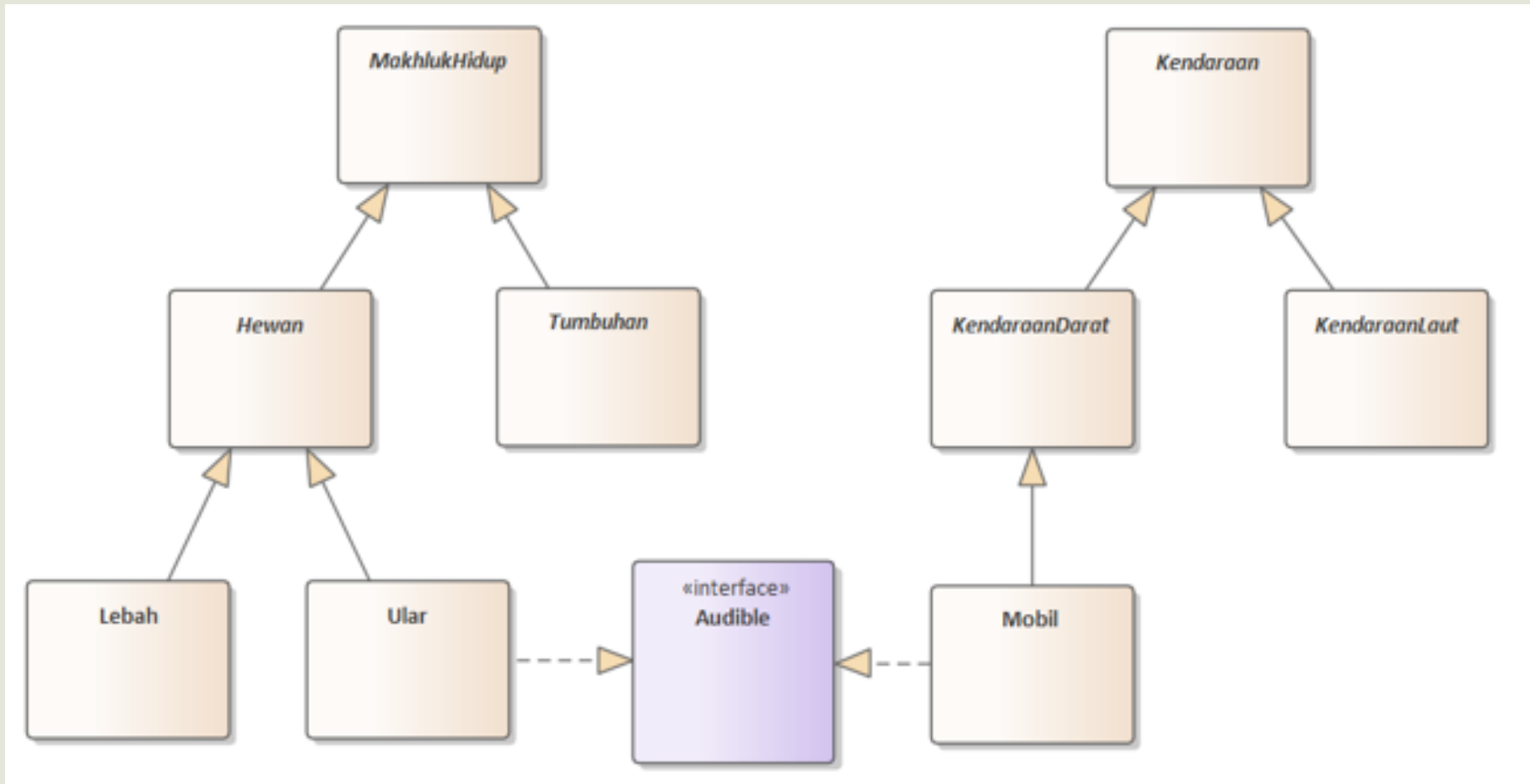
- Suatu class dapat meng-implement multiple interface
- Bila suatu class merupakan subclass dan meng-implement interface, maka keyword **extends** mendahului **implements**
- Contoh:

```
public class PlainCandy extends GameItem implements Crushable, Movable
```

Abstract Class vs Interface

Abstract Class	Interface
Dapat memiliki concrete method atau abstract method	Hanya dapat memiliki abstract method
Level modifier atribut dan method: public, protected, no-modifier, private	Level modifier variable dan method hanya public (boleh tidak dituliskan)
Dapat memiliki static/non-static, final/non final variable	Hanya dapat memiliki static dan final variable
Method boleh bersifat static/non-static dan final/non final	Method tidak boleh bersifat static dan final
Abstract class harus terdapat dalam hirarki yang sama dengan class yang meng-extend	Interface tidak terkait pada suatu hirarki

Interface tidak terikat pada hirarki

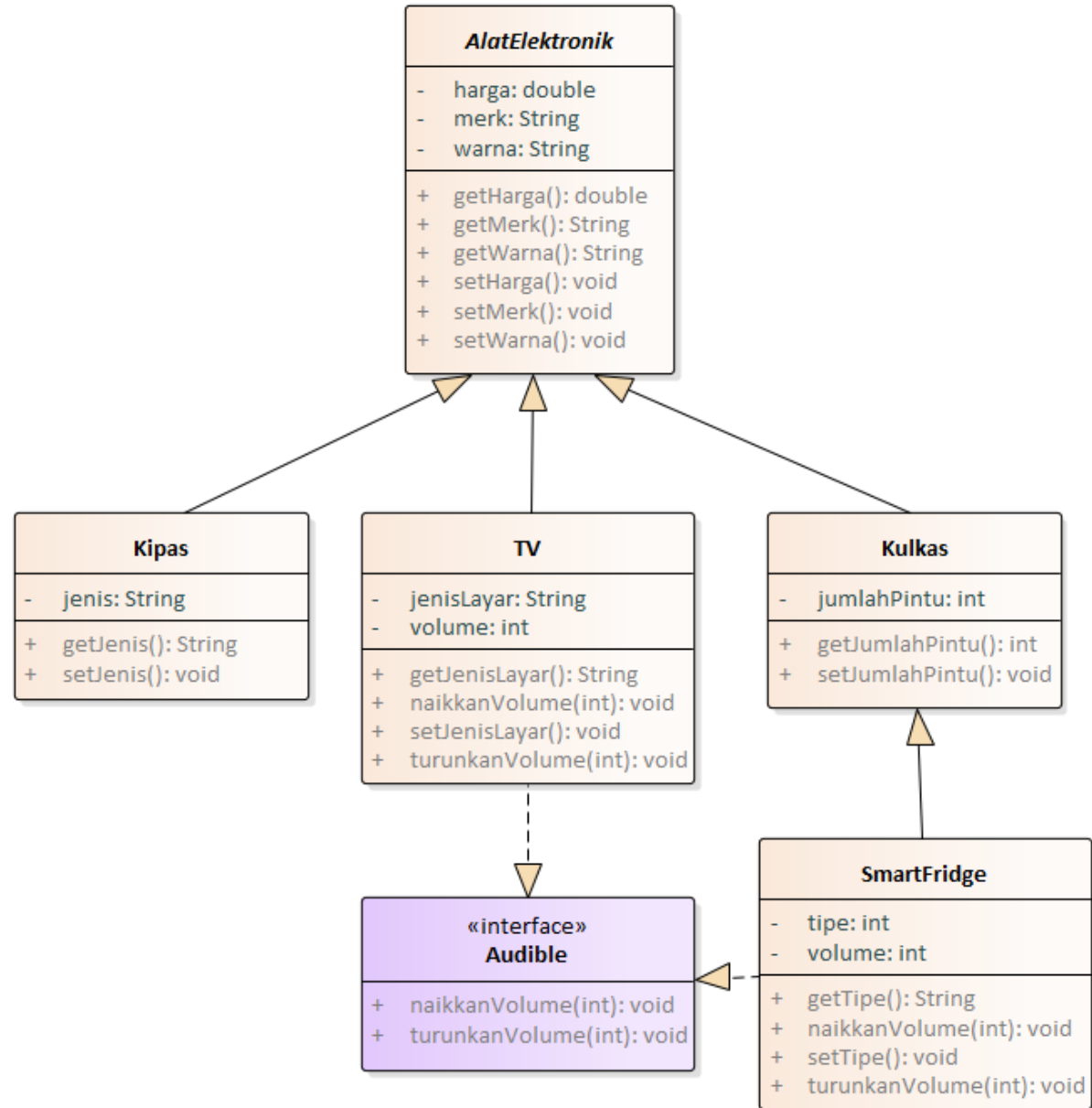


Penggunaan Abstract Class vs Interface

- Common properties (and methods) → abstract class
- Common methods → interface

Kesimpulan

- Interface tidak dapat dibuat objek/*instance*-nya
- Interface merepresentasikan **kontrak/syarat** yang berisi sekumpulan *behavior* yang saling **terkait** untuk memenuhi suatu **kapabilitas**



Latihan kelompok

- Cari 1 contoh dari interface berupa gambar UML class diagram dan identifikasi
- identifikasi project PBL ada tidak terkait interface