

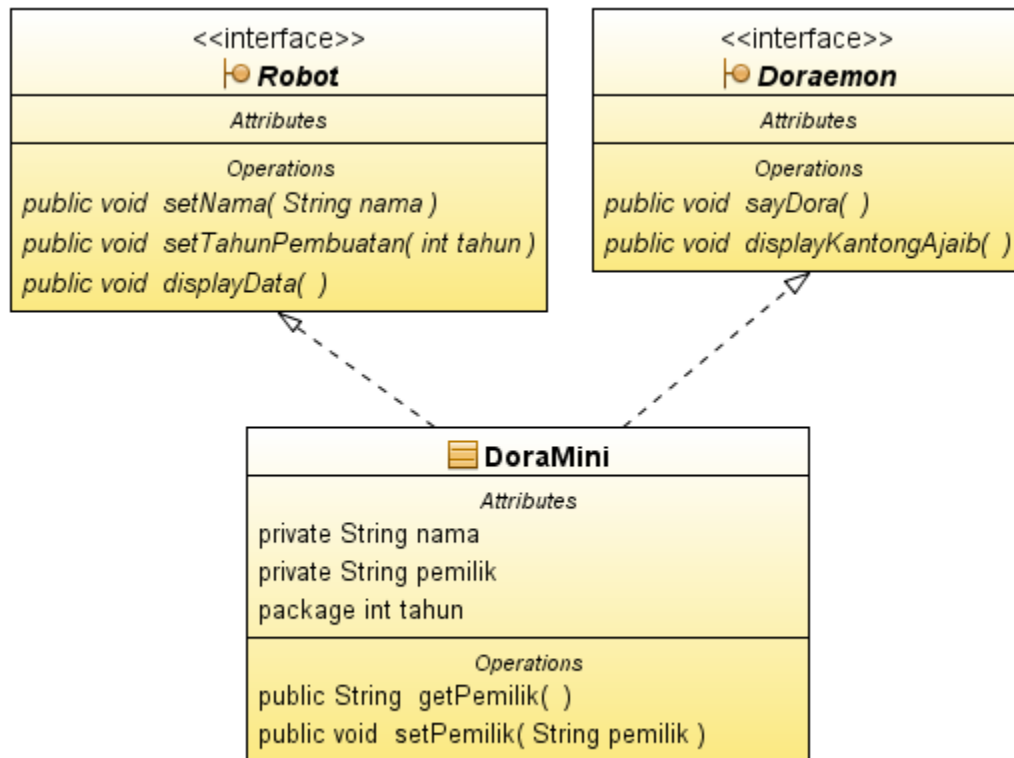
Interface

Interface adalah prototype kelas yang berisi definisi konstanta dan deklarasi method (hanya nama method tanpa definisi kode programnya). Terdapat 2 fungsi utama dari interface, yaitu:

1. Memungkinkan 2 class yang tidak sama memiliki sebuah behaviour yang spesifik. Contoh, tombol (button) dan combo box merupakan 2 komponen yang berbeda, tapi ketika ditekan atau dimodifikasi memiliki behaviour yang sama, yaitu membutuhkan suatu “aksi”. Aksi tersebut bisa didapat dari interface.
2. Memungkinkan terjadinya multiple inheritance, yaitu kondisi sebuah class memiliki lebih dari 2 parent.

Penggambaran interface pada UML menggunakan “keyword” <<interface>> untuk menyatakan interface, dan simbol realization (panah putus-putus dengan akhir segitiga) untuk menyatakan hubungan “implements”.

Contoh:



Buatlah 3 kelas berdasarkan diagram kelas di atas (beserta kelas Main). Keterangan tambahan:

- 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

```
public interface RobotInt{
    public void setNama(String nama);
    public void setTahunPembuatan(int tahun);
    public void displayData();
}
```

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

```
public class DoraMini implements RobotInt, Doraemon{
    private String nama;
    private String pemilik;
    int tahun;
    public String getPemilik(){
        return this.pemilik;
    }
    public void setPemilik(String pemilik){
        this.pemilik = pemilik;
    }
    @Override
    public void setNama(String nama){
        this.nama = nama;
    }
    @Override
    public void setTahunPembuatan(int tahun){
        this.tahun = tahun;
    }
    @Override
    public void displayData(){
        System.out.println("Nama:"+this.nama);
        System.out.println("Pemilik:"+this.pemilik);
        System.out.println("Tahun Produksi:"+this.tahun);
    }
    @Override
    public void sayDora(){
        System.out.println("Say Dora");
    }
    @Override
    public void displayKantongAjaib(){
        System.out.println("Kantong Ajaib");
    }
}
```

```
public class MainInt{
    public static void main(String args[]){
        DoraMini dm = new DoraMini();
        dm.setPemilik("Nobita");
        dm.setNama("Dora Mini Satu");
        dm.setTahunPembuatan(2014);
        dm.displayData();
        dm.sayDora();
        dm.displayKantongAjaib();
    }
}
```

Hasil:

```
[rbudiawan@eja Java]$ java MainInt
Nama:Dora Mini Satu
Pemilik:Nobita
Tahun Produksi:2014
Say Dora
Kantong Ajaib
[rbudiawan@eja Java]$ █
```

Perhatikan bahwa:

1. Tidak ada body method pada interface
2. DoraMini harus meng-override semua method dari interface yang diimplements olehnya.
3. Annotasi `@Override` dapat ditambahkan di atas method yang di-override. Hal ini berlaku seperti pada kode inheritance/pewarisan.