



Riza Satria Perdana, S.T., M.T.

Teknik Informatika - STEI ITB

Interface

Contoh Interface

Pemrograman Berorientasi Objek

Contoh

```
public interface Relatable {  
    // this (object calling isLargerThan)  
    // and other must be instances of  
    // the same class returns 1, 0, -1  
    // if this is greater than,  
    // equal to, or less than other  
  
    public int isLargerThan(Relatable other);  
}
```

```
public class RectanglePlus implements Relatable {  
    public int width = 0;  
    public int height = 0;  
    public Point origin;  
  
    // four constructors  
    public RectanglePlus() {  
        origin = new Point(0, 0);  
    }  
    public RectanglePlus(Point p) {  
        origin = p;  
    }  
}
```



Contoh ...

```
public RectanglePlus(int w, int h) {  
    origin = new Point(0, 0);  
    width = w; height = h;  
}  
public RectanglePlus(Point p, int w, int h) {  
    origin = p;  
    width = w; height = h;  
}  
  
// a method for moving the rectangle  
public void move(int x, int y) {  
    origin.x = x;  
    origin.y = y;  
}  
  
// a method for computing the area of the rectangle  
public int getArea() {  
    return width * height;  
}
```

Contoh ...

// a method required to implement the Relatable interface

```
public int isLargerThan(Relatable other) {  
    RectanglePlus otherRect = (RectanglePlus)other;  
    if (this.getArea() < otherRect.getArea())  
        return -1;  
    else if (this.getArea() > otherRect.getArea())  
        return 1;  
    else  
        return 0;  
}  
}
```

Interface sebagai Type

```
public Object findLargest(Object object1, Object object2) {  
    Relatable obj1 = (Relatable)object1;  
    Relatable obj2 = (Relatable)object2;  
    if ((obj1).isLargerThan(obj2) > 0)  
        return object1;  
    else  
        return object2;  
}
```

Contoh ...

```
public Object findSmallest(Object object1, Object object2) {  
    Relatable obj1 = (Relatable)object1;  
    Relatable obj2 = (Relatable)object2;  
    if ((obj1).isLargerThan(obj2) < 0)  
        return object1;  
    else  
        return object2;  
}
```

```
public boolean isEqual(Object object1, Object object2) {  
    Relatable obj1 = (Relatable)object1;  
    Relatable obj2 = (Relatable)object2;  
    if ( (obj1).isLargerThan(obj2) == 0)  
        return true;  
    else  
        return false;  
}
```

Rewriting Interface

```
public interface DoIt {  
    void doSomething(int i, double x);  
    int doSomethingElse(String s);  
}
```

```
public interface DoItPlus extends DoIt {  
    boolean didItWork(int i, double x, String s);  
}
```

Ringkasan

- Interface mendefinisikan protokol komunikasi antara dua objek
- Deklarasi interface mengandung signature (tapi tanpa implementasi) dari sekumpulan method dan definisi konstanta

Ringkasan

- Kelas yang mengimplementasikan interface harus mengimplementasikan seluruh method yang dideklarasikan di interface
- Nama interface bisa digunakan sebagai Type

Terima Kasih