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

Teknik Informatika - STEI ITB

Nested Class

Pemrograman Berorientasi Objek

Nested Class

- Pada Java dimungkinkan untuk mendefinisikan kelas di dalam suatu kelas
- Disebut sebagai: **Nested Class**
- Nested class dibagi menjadi 2 kategori: static dan non static
- Nested class yang static disebut Static Nested Class

Contoh Nested Class

```
public class OuterClass {
    ...
    class NestedClass {
        ...
    }
}

public class OuterClass {
    ...
    static class StaticNestedClass {
        ...
    }
    class InnerClass {
        ...
    }
}
```

Nested Class ...

- Inner class non static mempunyai akses ke method atau atribut dari outer class-nya meskipun private
- Static nested class tidak memiliki akses tsb.

Mengapa Nested Class?

- Salah satu cara untuk mengelompokkan kelas yang hanya digunakan di satu tempat
- Meningkatkan enkapsulasi
- Dapat meningkatkan readability dan maintainability kode

Static Nested Class

- Seperti halnya method dan atribut yang lain, static nested class diasosiasikan dengan outer class-nya
- Tidak bisa mengakses langsung instans atribut atau method dari outer class-nya

Contoh

```
public class OuterShape {
    private String shapeName = “Outer Shape”;
    private static String className = “OuterShape”;
    private void printHello() {
        System.out.println(“hello from outer class”);
    }

    // Static nested class StaticPoint
    public static class StaticPoint {
        private double x;
        private double y;

        public StaticPoint(double x, double y) {
            this.x = x;
            this.y = y;
        }
        public void displayInfo() {
            System.out.println("Class Name: " + className); // can we do this?
            System.out.println("Shape Name: " + shapeName); // what about this?
            printHello(); // ??
        }
```

Contoh

```
// Non-static inner class - can access all members of outer class
public class InnerRectangle{
    private double width;
    private double height;

    public InnerRectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    public void displayInfo() {
        System.out.println("Shape Name: " + shapeName);  // Access private field
        printHello();  // Access private method
        System.out.println("Width: " + width);
        System.out.println("Height: " + height);
    }

    public void updateShapeName(String name) {
        shapeName = name;
    }
}
```

Contoh

```
    public static void main(String[] args) {
        // Creating instance of Rectangle
        OuterShape outer = new OuterShape();
        InnerRectangle rectangle = outer.new InnerRectangle(5, 3);

        rectangle.displayInfo();
        rectangle.updateShapeName("My Rectangle");
        rectangle.displayInfo();

        // Creating instance of StaticPoint (doesn't need OuterShape instance)
        Point point = new Point(10, 20);
        point.displayInfo();
    }
}
```

Terima Kasih