

UJIAN AKHIR SEMESTER



PEMROGRAMAN BERORIENTASI OBJEK

YUSUF HAAZIMULHIKAM

A11.2020.12639

UNIVERSITAS DIAN NUSWANTORO SEMARANG

4 JULI 2022

1. Definisi dan implementasi kotlin :

a. **Nested Class**

Nested class adalah class yang telah dibuat di dalam class lain.

Implementasi dalam kotlin :

```
fun main(args: Array<String>) {  
    val test = Outer.Nested().pesan() // memanggil metode kelas bersarang  
    print(test)  
}  
class Outer {  
    class Nested {  
        fun pesan() = "Selamat Mengerjakan!"  
    }  
}
```

B. **Singleton**

Singleton adalah salah satu dari klasifikasi creational design pattern. Patten ini memastikan bahwa kamu hanya menggunakan satu instance, yang menyediakan akses secara global ke instance.

Implementasi dalam kotlin :

```
public class ListInstacesFrame extends javax.swing.JFrame {  
    /**  
     * Creates new form FirstFrame  
     */  
    public ListInstacesFrame() {  
        initComponents();  
        createList();  
    }  
  
    private void createList(){  
        // instantiating Singleton class with variable a  
        SingletonClassExample a = SingletonClassExample.getInstance();  
        // instantiating Singleton class with variable b  
        SingletonClassExample b = SingletonClassExample.getInstance();  
        // instantiating Singleton class with variable c  
        SingletonClassExample c = SingletonClassExample.getInstance();  
        DefaultListModel<String> StringList = new DefaultListModel<>();  
        StringList.addElement("String dari a is " + x.myText);  
        StringList.addElement("String dari b is " + y.myText);  
        StringList.addElement("String dari c is " + z.myText);  
        jList1.setModel(StringList);  
    }  
  
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {GEN-  
FIRST:event_jButton1ActionPerformed  
        // TODO add your handling code here:  
        new ManipulateStringFrame().setVisible(true);  
    }
```

```

        dispose();
    } //GEN-LAST:event_jButton1ActionPerformed
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ListInstacesFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify //GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JList<String> jList1;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration //GEN-END:variables
}

```

c. Interface

Antarmuka di kotlin mirip seperti java. Antarmuka dapat memiliki deklarasi metode abstrak. Yang membedakannya dengan abstract class adalah interface tidak dapat menyimpan status . Interface dapat memiliki properti tetapi harus abstrak.

Implementasi dalam kotlin :

```

Interface A {
    Fun foo() { print("A") }
    Fun bar()
}
Interface B {
    Fun foo() { print("B") }
    Fun bar() { print("bar") }
}
Class C: A {
    Override fun bar() { print("bar") }
}
Class D: A, B{
    Override fun foo() {
        Super<A>.foo()
        Super<B>.foo()
    }

    Override fun bar() {
        Super<B>.bar()
    }
}

```

2. Persamaan dan perbedaan bahasa pemrograman kotlin dan java

Persamaan :

- Sama-sama dapat bekerja pada apa yang dapat dikerjakan Java. Ini karena pada Kotlin, kita juga dapat menggunakan Java jadi Kotlin, yang dapat beroperasi pada apapun yang menggunakan Java.
- Sama-sama Bahasa yang digunakan untuk pemrograman berbasis android. Java dan Kotlin adalah 2 bahasa yang sama-sama dapat beroperasi pada platform android.

Perbedaan :

- Memiliki jenis platform yang berbeda. Jika Java hanya bisa perintah untuk merancang aplikasi berbasis android saja, Kotlin didukung system yang memungkinkan untuk membuat program aplikasi yang berbasis android dan iOS.
- Perbedaan kompilasi. Kecepatan kompilasi koding antara koding Java dan Kotlin berbeda. Kompilasi koding pada Java lebih cepat namun sering terjadi debug. Beda halnya dengan Kotlin, walau kecepatan kompilasinya lebih lambat namun kemungkinan Null pada aplikasi yang dirancang dapat anda hindari.