

MODUL PRAKTIKUM

Graphical User Interface (GUI) dan Collection

a. Tujuan

- Mengenal dan menerapkan GUI sebagai antarmuka grafis dalam pemrograman berorientasi Objek
- Mengimplementasikan penggunaan Komponen JList
- Mengimplementasikan penggunaan Collection (List, Set, Map)

b. Landasan Teori

How to Use Lists: <https://docs.oracle.com/javase/tutorial/uiswing/components/list.html>

Java JList - Add, Update, Delete, Clear: <https://youtu.be/w7xiY0fc6cs>

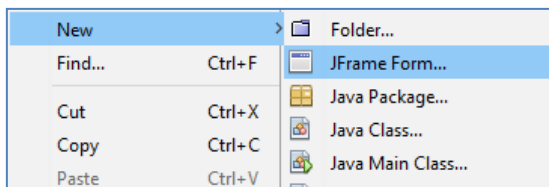
Membuat file JAR: <https://youtu.be/6U7xkdj6RJs>

Dan berbagai sumber referensi dari perkuliahan dan Internet

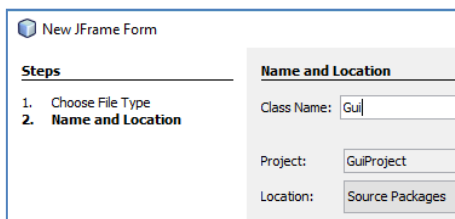
c. Langkah Praktikum

Proyek 0. Memulai proyek melalui Netbeans Gui Builder

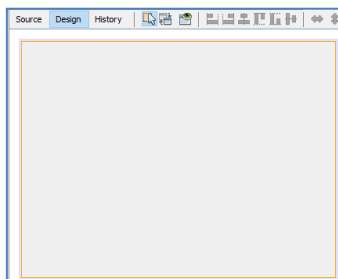
1. Pilih JFrame Form:



2. Beri nama kelas:

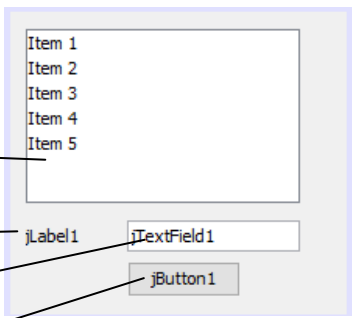
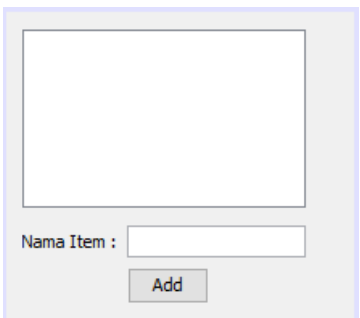


3. Selesai



Proyek 1. Menambahkan item ke jList

1. Merancang tampilan dan variabel menggunakan Netbeans GUI Builder

Atur posisi Swing control dari menu Palletes		Atur nama tampilan
		
Atur nama variabel:		
<pre>116 // Variables declaration - do not modify 117 private javax.swing.JButton jButtonAdd; 118 private javax.swing.JLabel jLabelNamaItem; 119 private javax.swing.JList jListItem; 120 private javax.swing.JScrollPane jScrollPane1; 121 private javax.swing.JTextField jTextFieldNamaItem; 122 // End of variables declaration</pre>		

2. Menyiapkan model menggunakan DefaultListModel untuk jList

```
public class GuiSimple extends javax.swing.JFrame {

    DefaultListModel<String> dlm;

    /**
     * Creates new form GuiSimple
     */
    public GuiSimple() {
        initComponents();
        this.dlm = new DefaultListModel<>();
        this.jListItem.setModel(dlm);
    }
}
```

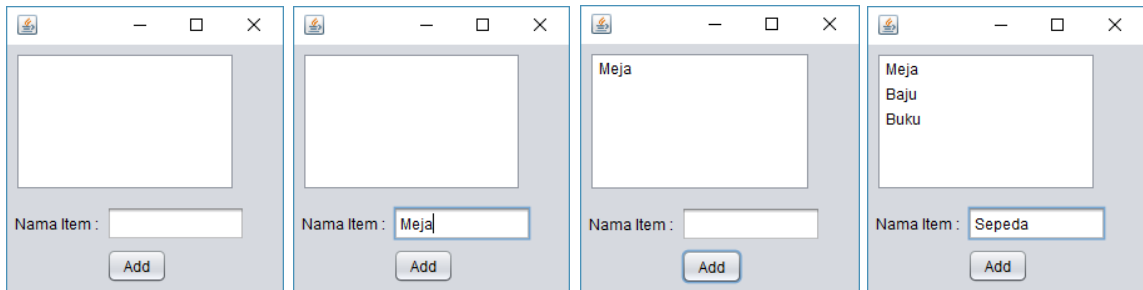
3. Membuat method addItem()

```
private void addItem(String namaItem) {
    dlm.addElement(namaItem);
}
```

4. Mengatur event action pada tombol jButtonAdd

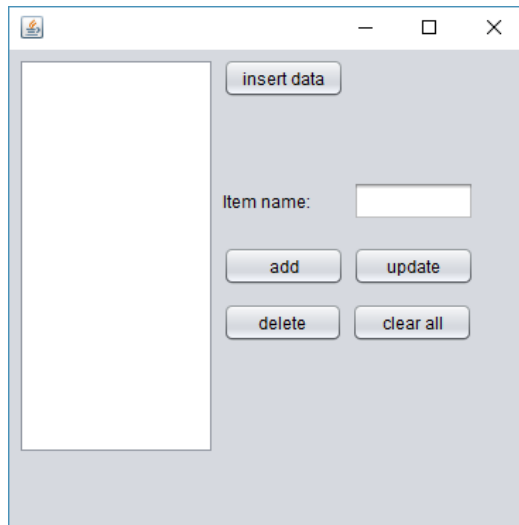
```
private void jButtonAddActionPerformed(java.awt.event.ActionEvent evt) {  
    addItem(jTextFieldNamaItem.getText());  
  
    jTextFieldNamaItem.setText("");  
}
```

5. Selesai



Proyek 2. Add, update, delete item pada jList

1. Rancanglah agar penataan tampilan eksekusi sebagai berikut



2. Beri even aksi pada Tombol **add**

Kerjakan sama seperti proyek 1

3. Beri even aksi pada tombol **delete**

```
int index = jListItem.getSelectedIndex();  
dml.removeElementAt(index);  
  
jTextFieldItem.setText("");
```

3. Beri even aksi pada tombol **update**

```
int index = jListItem.getSelectedIndex();  
String selected = jTextFieldItem.getText();  
dml.setElementAt(selected, index);  
  
jTextFieldItem.setText("");
```

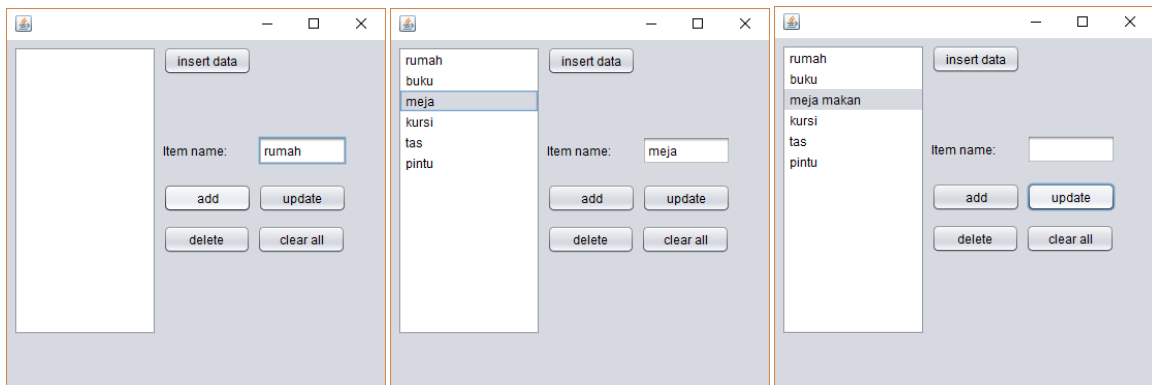
4. Beri even aksi pada tombol **clear all**

```
dml.clear();
```

5. Beri even aksi pada tombol **insert data**

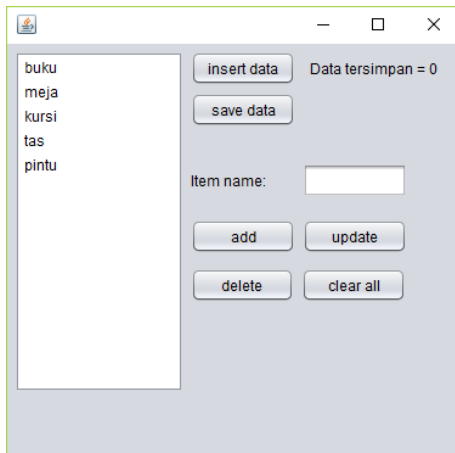
```
String[] items = {"buku", "meja", "kursi", "tas", "pintu"};  
for (String item : items) {  
    dml.addElement(item);  
}
```

5. Selesai



Proyek 3. Save dari jList ke List

Menambahkan fitur **save** data ke **list**, dan data tersebut dapat di tambahkan kembali (**insert data**) ke **jList**:



Misalkan dengan hasil pengaturan variabel sebagai berikut:

```
// Variables declaration - do not modify
private javax.swing.JButton jButtonAddItem;
private javax.swing.JButton jButtonClearAll;
private javax.swing.JButton jButtonDelete;
private javax.swing.JButton jButtonInsertData;
private javax.swing.JButton jButtonUpdate;
private javax.swing.JLabel jLabelItem;
private javax.swing.JList jListItem;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextField jTextFieldItem;
// End of variables declaration
```

1. Mengatur nilai awal

```
public class GuiCollection extends javax.swing.JFrame {

    DefaultListModel<String> dlm;
    List<String> items = new ArrayList<>();
    String[] datas = {"buku", "meja", "kursi", "tas", "pintu"};

    /**
     * Creates new form Gui
     */
    public GuiCollection() {
        initComponents();
        dlm = new DefaultListModel<>();
        jListItem.setModel(dlm);

        for (String data : datas) {
            dlm.addElement(data);
            updateJumDataTersimpan();
        }
    }
}
```

```
private void updateJumDataTersimpan() {
    jLabelJumlahData.setText(
        "Data tersimpan = " + items.size());
}
```

2. Beri even aksi pada tombol **save data**

```
if (!items.isEmpty()) {
    items.clear();
}

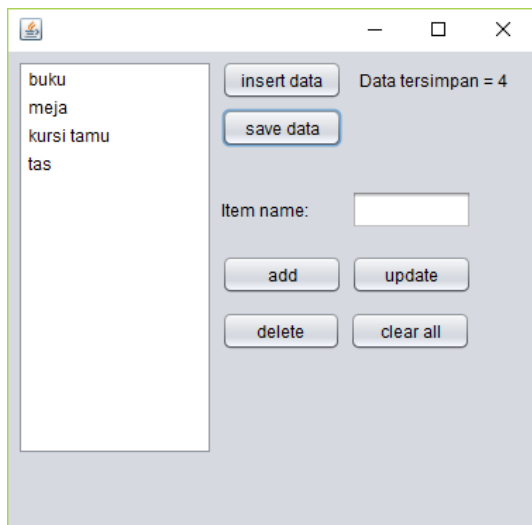
for (int i = 0; i < dlm.getSize(); i++) {
    items.add(dlm.getElementAt(i));
}

updateJumDataTersimpan();
```

3. Beri even aksi pada tombol **insert data**

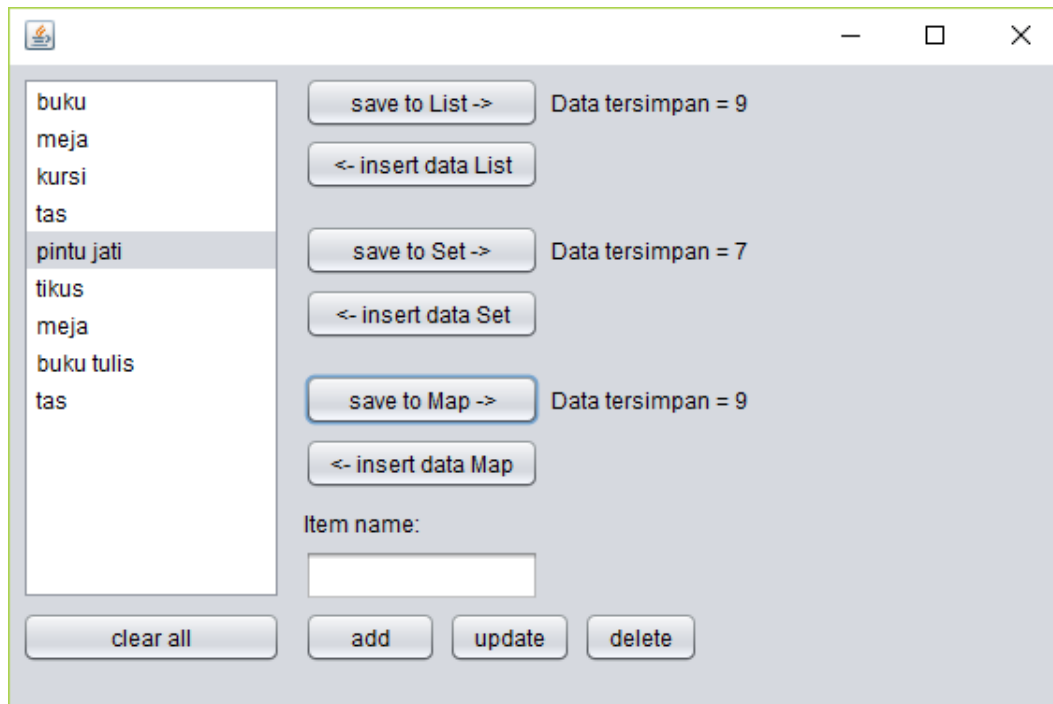
```
for (String item : items) {
    dlm.addElement(item);
}
```

4. Selesai



Proyek 4. Berlatih menyimpan ke dan mengambil dari: List, Set, dan Map

Berdasarkan pengalaman sebelumnya, buat aplikasi seperti contoh berikut:



Pastikan setelah menyelesaikan aplikasi ini, anda telah menguasai operasi menyimpan dan mengambil data dari List, Set, dan Map, serta memahami karakteristik masing-masing termasuk kelebihan dan kekurangannya.