

Praktikum JTable

Praktikum 1 :

Buatlah aplikasi seperti gambar 1. Pada aplikasi tersebut user dapat menambahkan, mengedit dan menghapus data. Data yang dimasukkan berupa nama, alamat, no telp dan email. Data user ditampilkan pada Jtable.

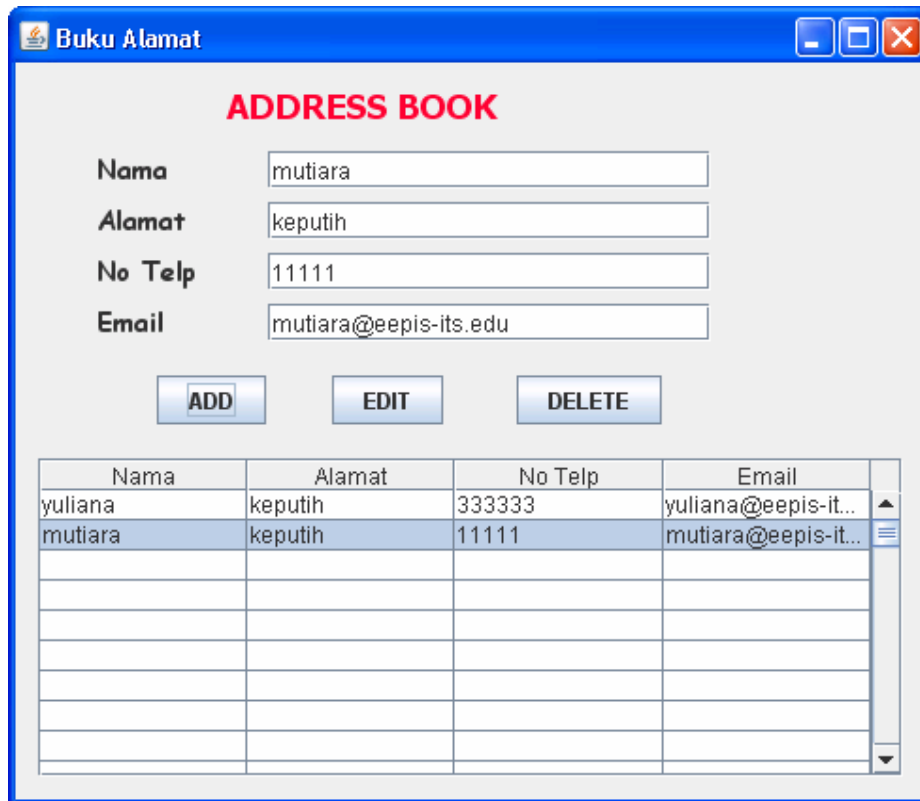


| Nama | Alamat | No Telp | Email |
|------|--------|---------|-------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Gambar 1

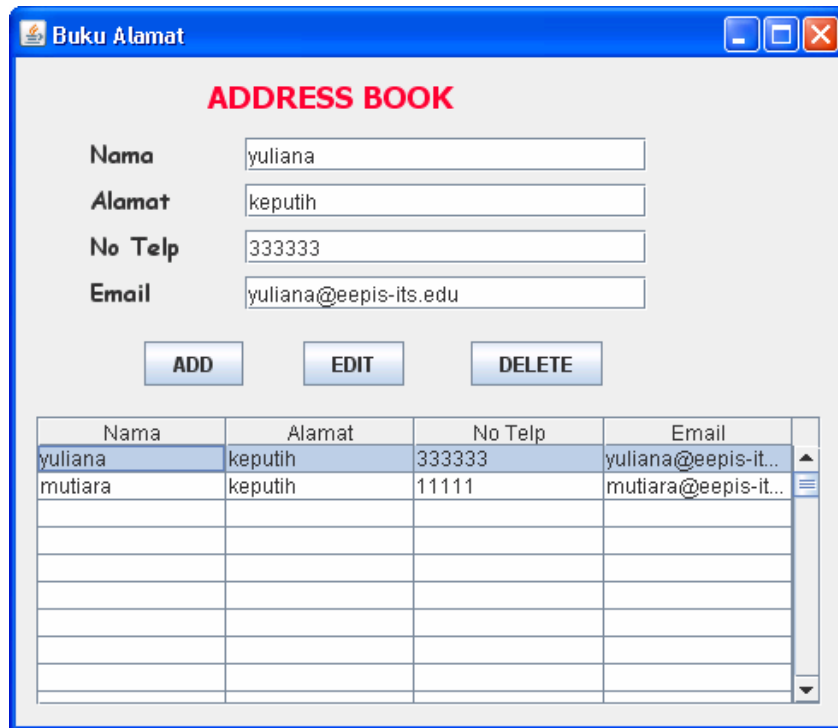
Output program :

1. Jika user memasukkan data nama, alamat, no telp dan email selanjutnya tekan button Add, maka data user akan ditampilkan pada table (gambar 2).

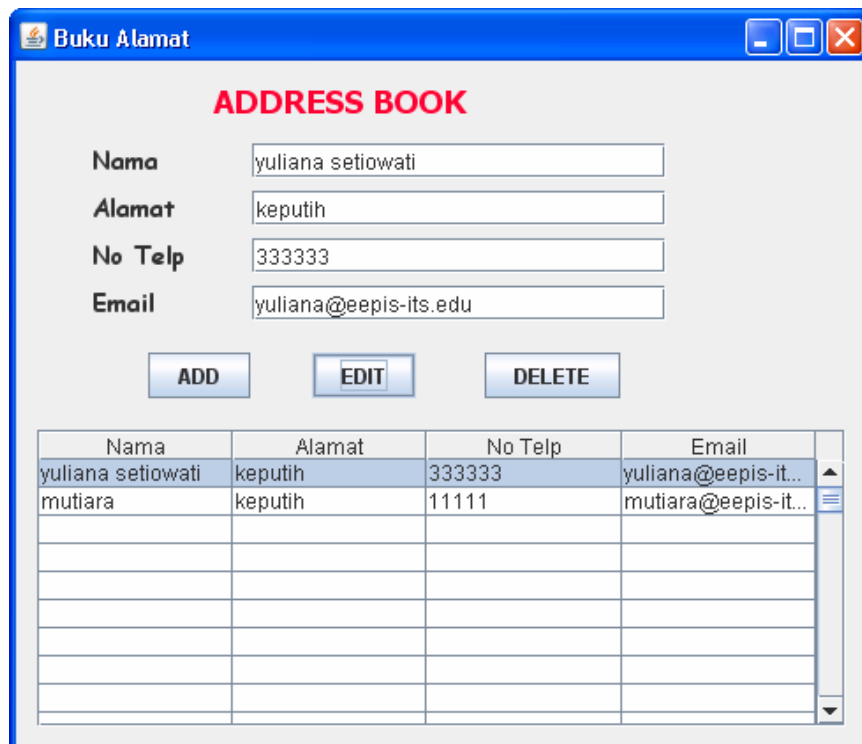


Gambar 2

2. Jika user ingin mengedit data maka arahkan tanda panah untuk menunjuk baris pada table. Maka baris yang terpilih akan ditampilkan pada textfield di atasnya (gambar 3). Selanjutnya user dapat mengedit data, untuk menyimpan tekan button Edit, maka data yang telah diedit akan tersimpan ditable (gambar 4)



Gambar 3

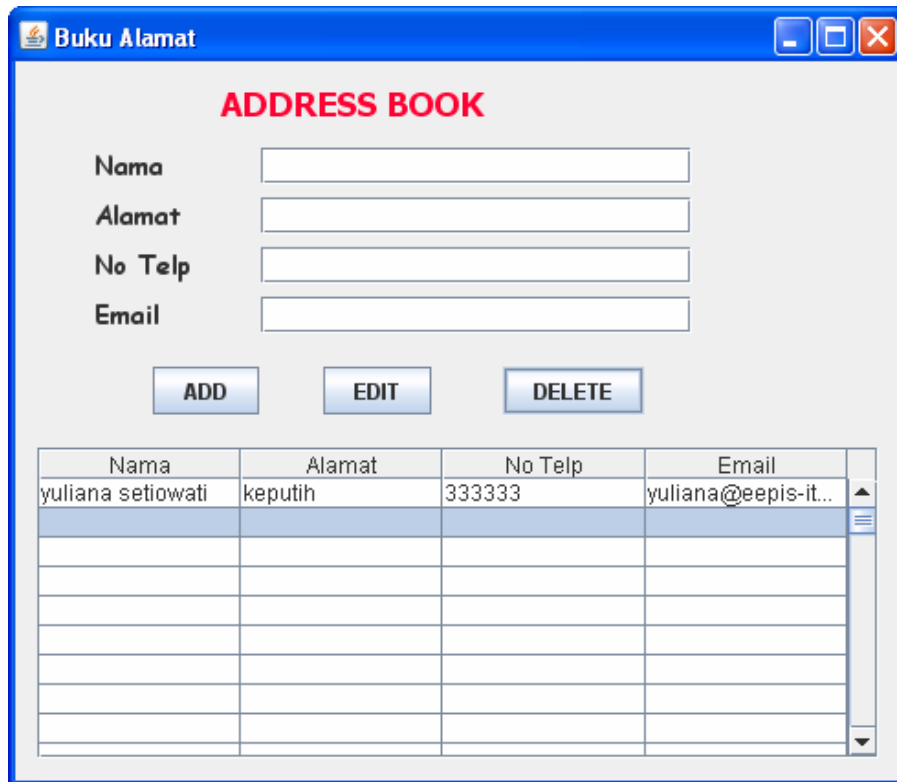


Gambar 4

3. User melakukan delete pada data, arahkan tanda kursor pada baris yang dipilih pada table (gambar 5). Selanjutnya tekan button Delete maka data pada table akan dihapus (gambar 6).

| Nama | Alamat | No Telp | Email |
|-------------------|---------|---------|---------------------|
| yuliana setiowati | keputih | 333333 | yuliana@eepis-it... |
| mutiara | keputih | 11111 | mutiara@eepis-it... |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Gambar 5



Gambar 6

Untuk menampilkan data pada object `JTable` gunakan fungsi `public void setModel(TableModel dataModel)`. Parameter dari fungsi adalah object dari class `TableModel`. Sehingga buat class `TableModel` (class `MyTableModel` adalah class yang extends `AbstractTableModel`)

```
import javax.swing.table.AbstractTableModel;
public class MyTableModel extends AbstractTableModel {
    private String[] columnNames = {"Nama", "Alamat", "No
Telp", "Email"};
    private Object[][] data = new Object[30][4];

    public int getColumnCount() {
        return columnNames.length;
    }

    public int getRowCount() {
        return data.length;
    }

    public String getColumnName(int col) {
        return columnNames[col];
    }

    public Object getValueAt(int row, int col) {
```

```

        return data[row][col];
    }

    public void setValueAt(Object value, int row, int col) {
        data[row][col] = value;
        this.fireTableCellUpdated(row, col);
    }
}

```

Pada object JTable terdapat 4 kolom yaitu Nama, Alamat, No Telp dan Email.

```

private String[] columnNames = {"Nama", "Alamat", "No
Telp", "Email"};

```

Data disimpan dalam array dimensi dua (variabel : data)

```

private Object[][] data = new Object[30][4];

```

Fungsi untuk mendapatkan jumlah kolom tabel

```

public int getColumnCount() {
    return columnNames.length;
}

```

Fungsi untuk mendapatkan jumlah baris tabel

```

public int getRowCount() {
    return data.length;
}

```

Fungsi untuk mendapatkan nama kolom tertentu

```

public String getColumnName(int col) {
    return columnNames[col];
}

```

Fungsi untuk mendapatkan data (bertipe Object) pada baris dan kolom tertentu

```

public Object getValueAt(int row, int col) {
    return data[row][col];
}

```

Fungsi untuk mengisi data pada baris dan kolom tertentu. `fireTableCellUpdated(row, col)` untuk memberitahukan semua listener bahwa data pada baris dan kolom tertentu sudah di update.

```

public void setValueAt(Object value, int row, int col) {
    data[row][col] = value;
    this.fireTableCellUpdated(row, col);
}

```

```
}
```

Praktikum 2 :

Kembangkan dari aplikasi yang sebelumnya, dimana user dapat melakukan pencarian data user berdasarkan nama dan alamat.

Output program :

1. Pertama kali masukkan data user seperti gambar 7. Lakukan pencarian berdasarkan nama, masukkan nama yang akan dicari pada text field, maka pada table akan muncul data-data user yang mengandung nama tersebut.

Cara pengerjaan :

Letakkan Jtable yang baru (nama : tableSearch). Pada saat button Search di tekan maka buatlah objek dari class MyTableModel.

```
tableSearch.setModel(new MyTableModel());
```

untuk melakukan pengisian pada tableSearch, tambahkan variabel brsSearch (deklarasikan global) untuk menentukan data yang telah ditambahkan pada tableSearch. Pertama kali lakukan inisialisasi brsSearch = 0.

Untuk melakukan perbandingan antara keyword yang dimasukkan dengan data pada tableSearch gunakan fungsi pada String yaitu contains().

```
//cbSearch : nama combo box ;
//keyword : inputan dari text field
if (cbSearch.getSelectedIndex() == 0){
    for(int i=0;i<brs;i++){
        {
            String nm = (String) table.getValueAt(i, 0) ;
            if (nm.contains(keyword))
            {
                String alamat = (String) table.getValueAt(i, 1);
                String noTelp = (String) table.getValueAt(i, 2);
                String email = (String) table.getValueAt(i, 3);

                tableSearch.setValueAt(nm, brsSearch, 0);
                tableSearch.setValueAt(alamat, brsSearch, 1);
                tableSearch.setValueAt(noTelp, brsSearch, 2);
                tableSearch.setValueAt(email, brsSearch, 3);

                brsSearch++;
            }
        }
    }
}
```

Buku Alamat

BUKU ALAMAT

Nama
Alamat
No Telp
Email

| Nama | Alamat | No Telp | Email |
|---------|-------------------|---------|-----------------------|
| yuliana | Keputih Indah 1 | 333333 | yuliana@eepis-its.edu |
| yuliana | Keputih Kejawan 2 | 333333 | yuliana@hotmail.com |
| Intan | Keputih Kejawan 3 | 111111 | intan@hotmail.com |
| Intan | Nginden | 111111 | intan2@hotmail.com |
| Mutiara | Nginden 23 | 111111 | mutia@hotmail.com |
| | | | |
| | | | |

SEARCH **NAMA**

| Nama | Alamat | No Telp | Email |
|---------|-------------------|---------|---------------------|
| yuliana | Keputih Indah 1 | 333333 | yuliana@eepis-it... |
| yuliana | Keputih Kejawan 2 | 333333 | yuliana@hotmail... |
| | | | |
| | | | |

Gambar 7

2. Pada gambar 8 user dapat melakukan pencarian berdasarkan alamat. Masukkan alamat yang akan di cari, maka data user akan ditampilkan pada tabel.

BUKU ALAMAT

Nama

Alamat

No Telp

Email

| Nama | Alamat | No Telp | Email |
|---------|-------------------|---------|---------------------|
| yuliana | Keputih Indah | 333333 | yuliana@eepis-it... |
| yuliana | Keputih Kejawan 2 | 333333 | yuliana@hotmail... |
| Intan | Keputih Kejawan 3 | 111111 | intan@hotmail.com |
| Intan | Nginden | 111111 | intan2@hotmail.c... |
| Mutiara | Nginden 23 | 111111 | mutia@hotmail.c... |
| | | | |
| | | | |

SEARCH

| Nama | Alamat | No Telp | Email |
|---------|-------------------|---------|---------------------|
| yuliana | Keputih Indah | 333333 | yuliana@eepis-it... |
| yuliana | Keputih Kejawan 2 | 333333 | yuliana@hotmail... |
| Intan | Keputih Kejawan 3 | 111111 | intan@hotmail.com |
| | | | |
| | | | |

Gambar 8

Praktikum 3 :

Kembangkan dari aplikasi yang sebelumnya, pada aplikasi ini form penambahan, dan pencarian merupakan form yang berbeda dari form aplikasi utama.

Output program :

Pada gambar 10 merupakan halaman utama dari aplikasi yang akan dibuat. Terdapat 4 button pada aplikasi tersebut yaitu button add (untuk menambahkan data user), button edit (untuk mengedit data user), button delete (untuk menghapus data user) dan button search untuk melakukan pencarian user berdasarkan nama dan alamat. Jika kita tekan button add maka akan muncul form seperti gambar 10. Pada form tersebut masukkan nama, alamat, no telp dan email dari data user selanjutnya tekan button add, maka hasilnya akan tampak pada gambar 11.



Gambar 9



Gambar 10



Gambar 11

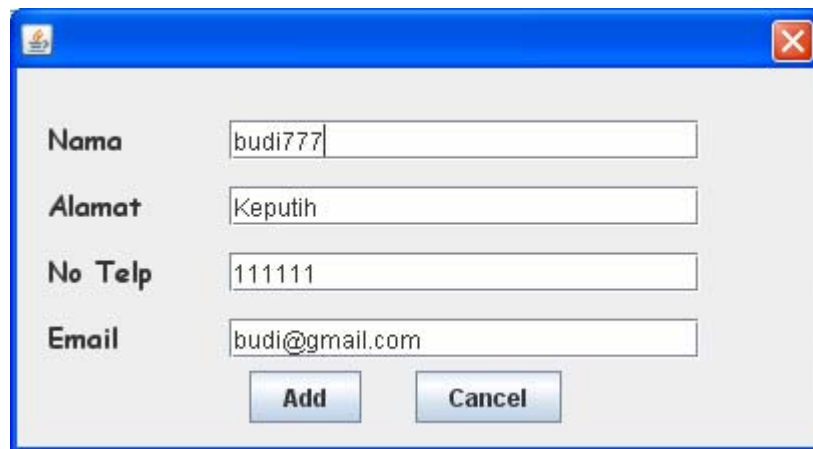
Pada gambar 12 ini merupakan contoh data user sebanyak 5 yang kita masukkan melalui form penambahan user. Pilihlah data dengan nama budi alamat Keputih (baris 2 – indeks dimulai baris 0) (gambar 13), lalu tekan button edit maka akan muncul form yang sama dengan form penambahan data (gambar 14), tapi juga menampilkan data user yang tadi kita pilih. Selanjutnya lakukan pengeditan terhadap data tersebut, tekan button edit. Hasil tampak pada table pada form utama (gambar 15)



Gambar 12



Gambar 13



Gambar 14



Gambar 15

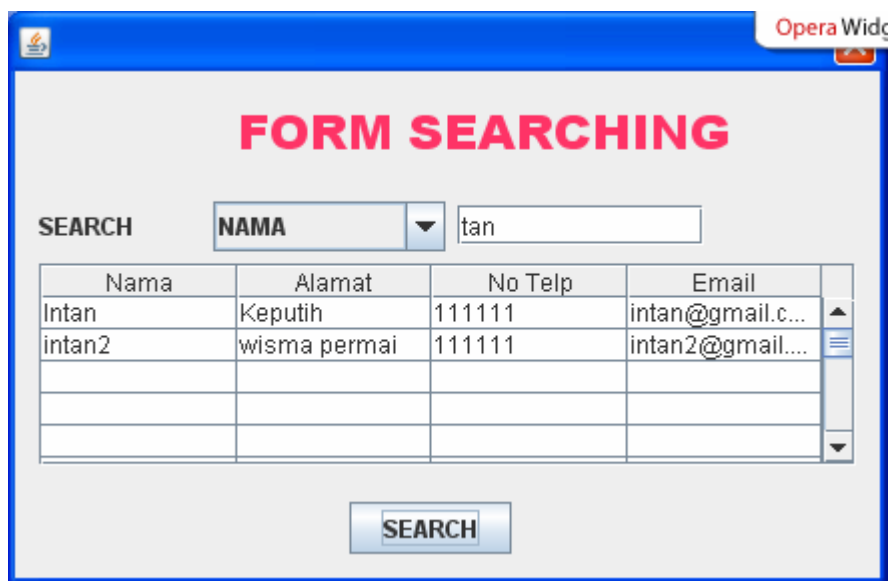
Lakukan pemilihan data pada baris 2 (gambar 15), selanjutnya tekan button delete, maka hasil akan tampak seperti gambar 16. Misalkan data terakhir seperti gambar 17, selanjutnya tekan button search, lakukan pencarian berdasarkan nama , masukkan kata kunci tan, maka table akan menampilkan data user dengan nama yang mengandung kata "tan" (gambar 18).



Gambar 16



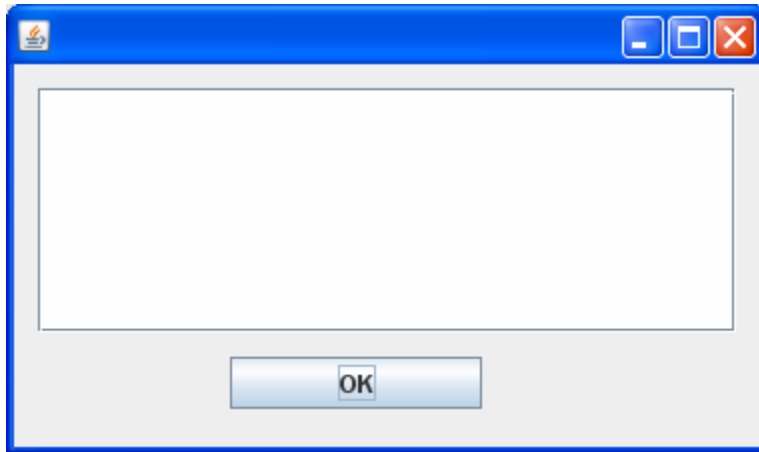
Gambar 17



Gambar 18

Persiapan Praktikum :

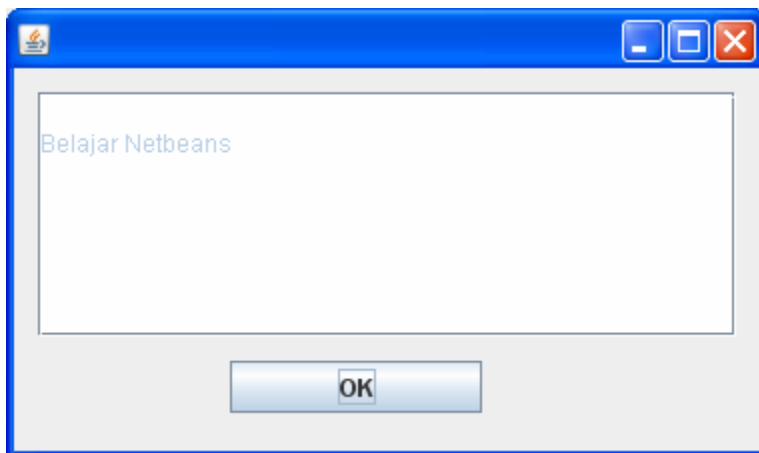
Buatlah aplikasi dimana terdapat form utama yang terdiri dari text area dan sebuah button(gambar 19), beri nama classnya dengan Frame1.java. Jika button ok ditekan maka akan muncul form seperti gambar 20 yang terdiri sebuah lable, sebuah textfield dan sebuah button (beri nama classnya Dialog1.java). Kita masukkan string dan tekan button ok (gambar 20), hasil akan tampak pada gambar 21, ulangi lagi dengan langkah yang sama maka hasil akan tampak pada gambar 21.



Gambar 19



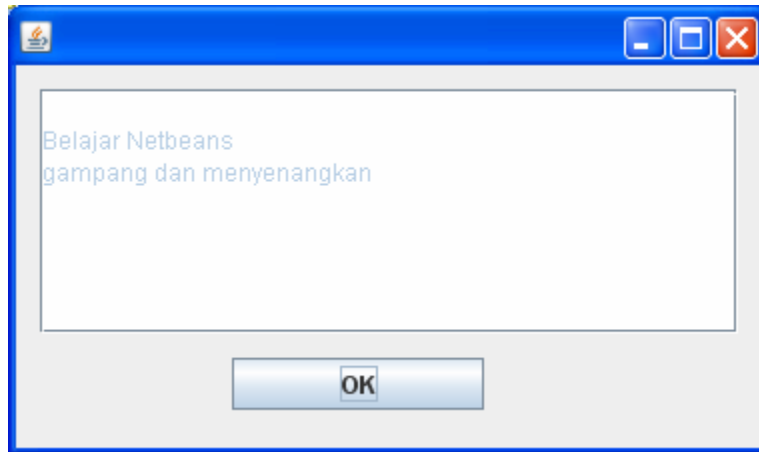
Gambar 20



Gambar 21

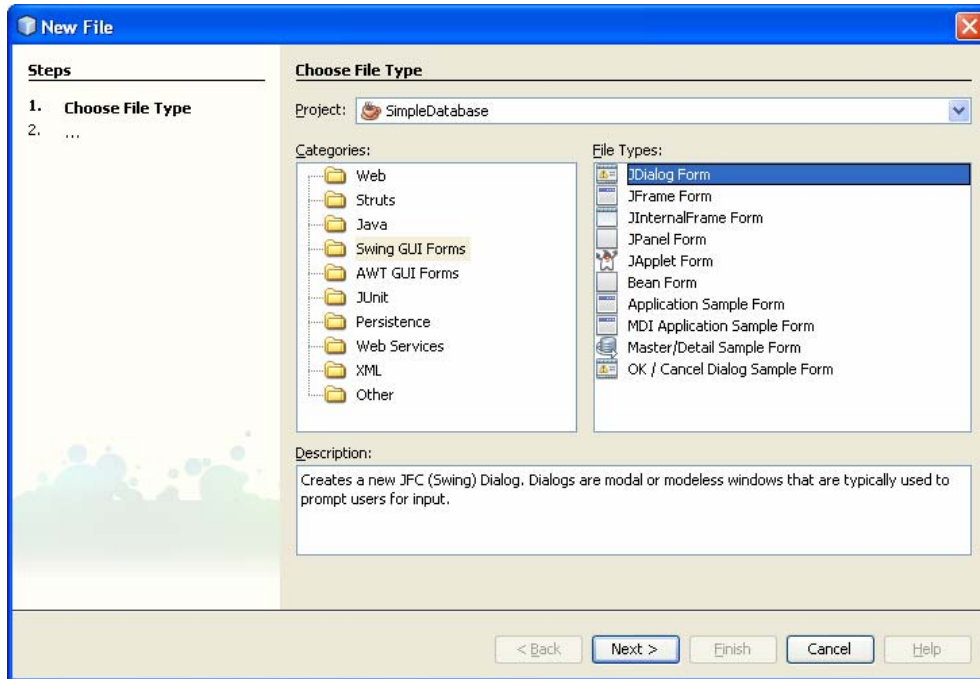


Gambar 22



Gambar 23

Untuk membuat form seperti gambar 20, buatlah file baru pilih kategori: Swing GUI Forms dengan type file JDialog Form (gambar 24). Class yang dibuat merupakan class yang extends class JDialog. Masukkan sebuah lable, sebuah textfield dan sebuah button, form seperti gambar 20.



Gambar 24

Pada class Dialog1, buat variabel f1 dari class Frame1, jangan lupa deklarasikan private (memenuhi aturan enkapsulasi). Pada konstruktor class Dialog1 terdapat 2 parameter yaitu Frame dan boolean, tambahkan :

```
f1 = (Frame1) parent;
```

Kemudian pada saat button OK ditekan pada form Dialog1, isi dari text field akan ditulis ke text area pada form Frame1. Buatlah fungsi dengan parameter String pada form Frame1 untuk menuliskan ke textArea. Sehingga fungsi ini dapat dipanggil dari form Dialog1 (deklarasikan public) cara memanggil `f1.isiText(tf.getText());` Untuk memanggil class Dialog1 `new Dialog1(this,true).setVisible(true);`

```
//Form Utama
public class Frame1 extends javax.swing.JFrame {

    /** Creates new form Frame1 */
    public Frame1() {
        initComponents();
    }

    public void isiText(String str){
        ta.append("\n"+str);
    }

    private void bOKActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Dialog1(this,true).setVisible(true);
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
```

```

        new Framel().setVisible(true);
    });
}
}

public class Dialog1 extends javax.swing.JDialog {
    private Framel f1 ;

    /** Creates new form Dialog1 */
    public Dialog1(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        f1 = (Framel) parent ;
        initComponents();
    }

    private void bOKActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        f1.isiText(tf.getText());
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                Dialog1 dialog = new Dialog1(new javax.swing.JFrame(), true);
                dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                    public void windowClosing(java.awt.event.WindowEvent e) {
                        System.exit(0);
                    }
                });
                dialog.setVisible(true);
            }
        });
    }
}

```

*****Selamat Mengerjakan*****