

# APLIKASI CRUD SEDERHANA JAVA GUI MENGUNAKAN DATABASE MYSQL

## A. TUJUAN

Setelah praktikum ini, praktikan diharapkan dapat memahami dan membuat program aplikasi java CRUD sederhana dengan database MySQL.

## B. PERALATAN DAN BAHAN

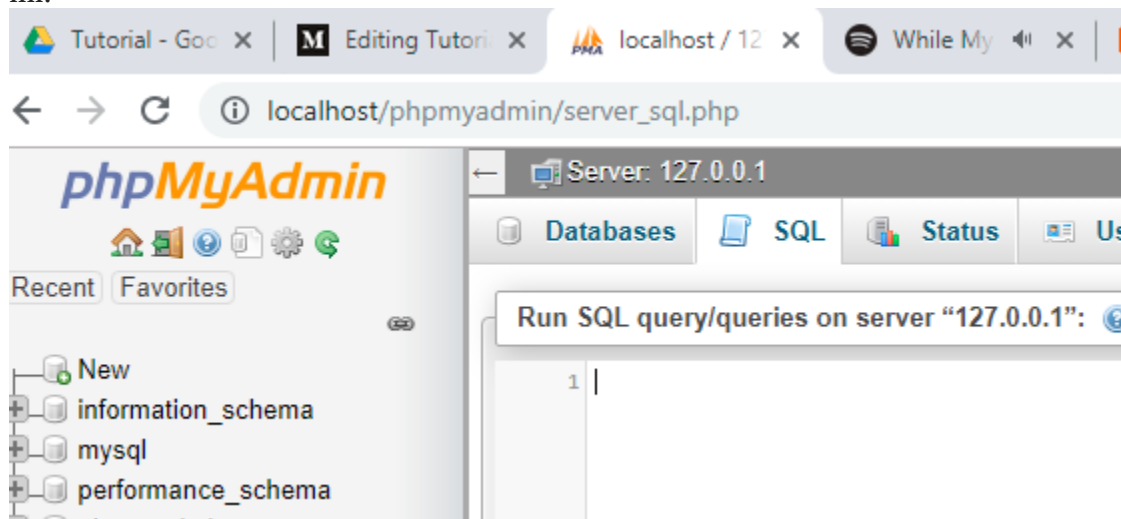
1. Personal Komputer
2. Software Netbeans

## C. TEORI

Java GUI berbasis Swing mampu membuat aplikasi java yang terkoneksi dengan database. Pada bab ini kita akan menggunakan database MySQL dan software XAMPP. Adanya koneksi antara java dengan database memungkinkan kita dapat menambah data, mengupdate data/ edit data, serta menghapus data yang ada pada database. Dalam menghubungkan coding pada java dan database MySQL kita membutuhkan library tambahan yaitu MySQL JDBC (Java Data Base Connectivity).

## D. PRAKTIKUM

Sebelum membuat database pastikan terlebih dahulu XAMPP kalian telah dibuka dan telah menjalankan server Apache dan MySql. Setelah itu kalian buka Localhost/phpmyadmin pada browser kalian seperti gambar di bawah ini:



Lalu klik pada tab **SQL**, kemudian masukkan syntaks

```
CREATE DATABASE db_siswa;
```

syntax di atas berfungsi untuk membuat database baru dengan nama tutorialjava, **nama database tersebut dapat kalian ubah sesuai kebutuhan.**

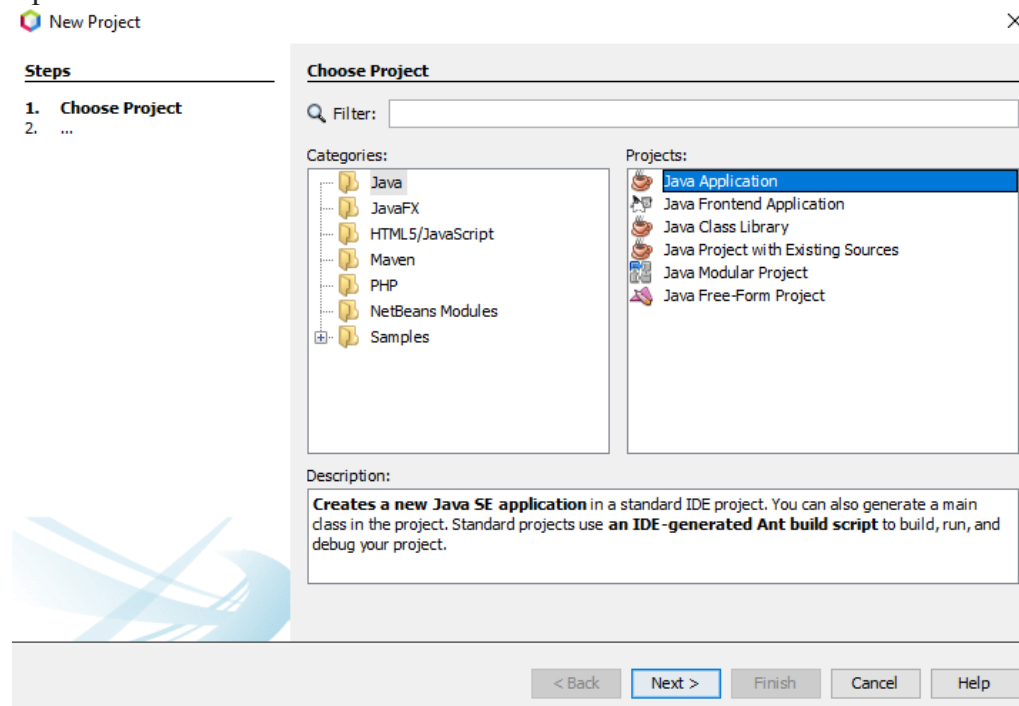
### Membuat Table

Setelah membuat database saatnya membuat table, pertama kalian harus select database yg telah kalian buat terlebih dahulu. setelah itu kalian tuliskan syntaxnya (caranya sama seperti membuat database) hanya saja syntaxnya yang berbeda, disini kita akan membuat sebuah table dengan nama **students** dengan isinya **nis, nama, jurusan, jk, alamat** seperti dibawah ini:

```
CREATE TABLE students(  
  nis int PRIMARY KEY,  
  nama varchar(100),  
  jurusan varchar(100),  
  jk varchar(16),  
  alamat text  
);
```

### Membuat Project di NetBeans

Pertama kalian harus membuka aplikasi Netbeans nya, lalu klik file -> new project. Setelah itu pilih kaegori Java dan pilih tipe project Java-nya Java Application lalu klik next:



The image shows the 'Name and Location' dialog box in NetBeans. It has a title bar 'Name and Location'. Below the title bar, there are three input fields: 'Project Name' with the text 'Java CRUD', 'Project Location' with the text 'C:\Users\ariqh\Documents\NetBeansProjects', and 'Project Folder' with the text 'C:\Users\ariqh\Documents\NetBeansProjects\Java CRUD'. To the right of the 'Project Location' field is a 'Browse...' button. Below these fields is a section with a checkbox 'Use Dedicated Folder for Storing Libraries' which is unchecked. Below this checkbox is a 'Libraries Folder' input field and another 'Browse...' button. Below that is a text label: 'Different users and projects can share the same compilation libraries (see Help for details)'. At the bottom of this section is another checkbox 'Create Main Class' which is unchecked, followed by an input field containing the text 'java.crud.JavaCRUD'. At the very bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish' (which is highlighted with a blue border), 'Cancel', and 'Help'.

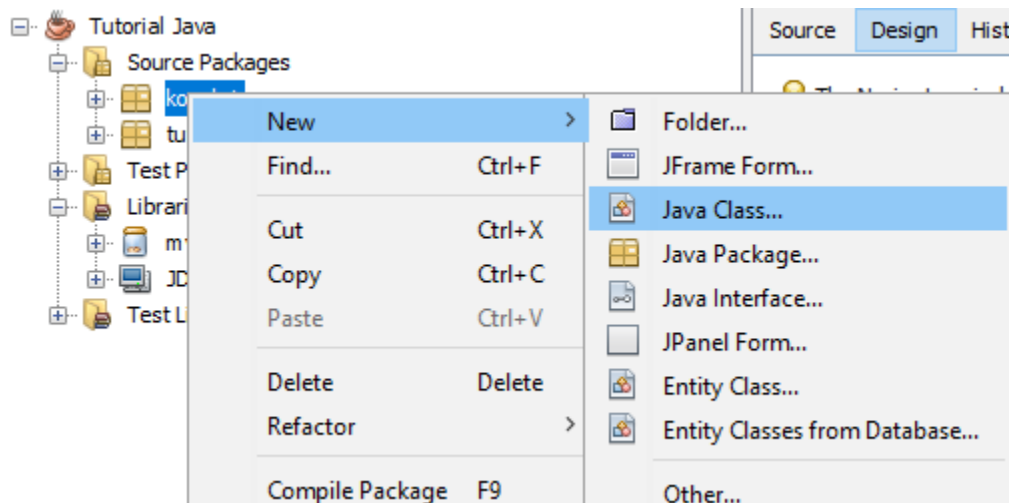
Beri nama dan **jangan ceklis pada Create Main Class**. Karena kita belum memerlukan sebuah Main Class. Setelah itu klik tombol finish, sekarang kita telah membuat project Java menggunakan Netbeans IDE.

### Koneksi ke Database

Setelah membuat project, saatnya membuat koneksi ke database db\_crud yang telah kita buat sebelumnya. Jadi yang pertama kali kita lakukan adalah menambah driver MySql ke dalam project yang kita buat. Caranya klik kanan bagian **Libraries -> Add Library, pilih MySql JDBC Driver lalu klik add library**.

Kemudian sekarang saatnya membuat sebuah Class yang digunakan untuk melakukan koneksi ke database MySql. Tapi sebelum kita membuat sebuah Class, pastikan kita membuat Package dulu, caranya klik kanan bagian Source Project yang telah kita buat lalu pilih **New -> Java Package dan beri nama "koneksi"**.

Setelah itu buat sebuah Class dengan mengklik kanan package yg telah kita buat tadi:



Beri nama **Koneksi**

**Koneksi.java**

```

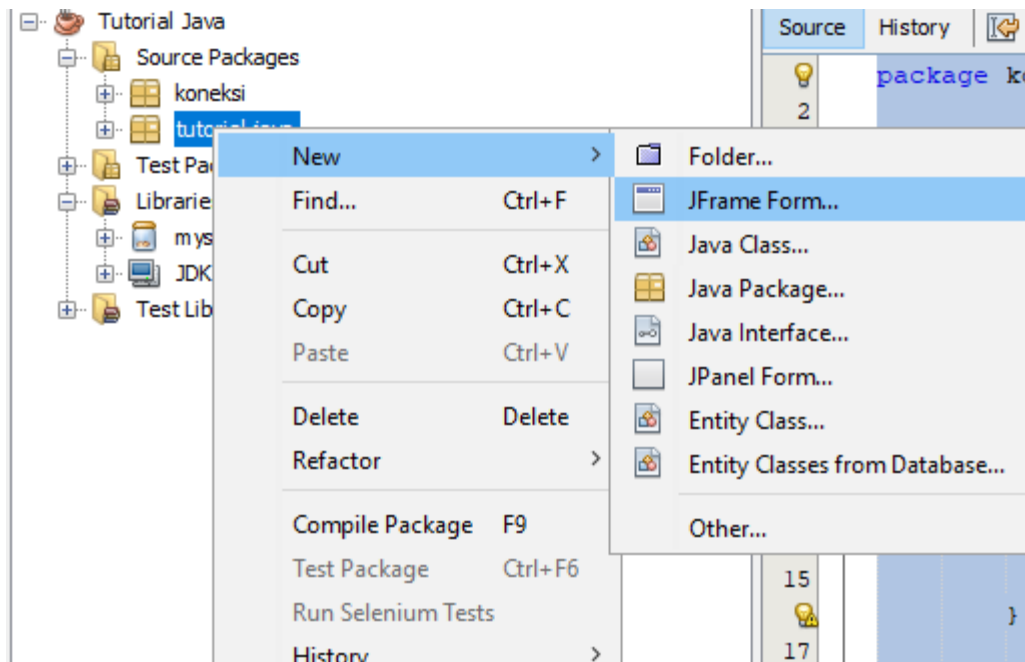
1  package koneksi;
2
3  import java.sql.Connection;
4  import java.sql.DriverManager;
5  import javax.swing.JOptionPane;
6
7  public class Koneksi {
8      Connection koneksi;
9
10     public static Connection Koneksi() {
11         try {
12             Class.forName("com.mysql.jdbc.Driver");
13             Connection koneksi = DriverManager.getConnection("jdbc:mysql:"
14                 + "://localhost/db_siswa", "root", "");
15
16             return koneksi;
17         } catch (Exception e) {
18             JOptionPane.showMessageDialog(null, e);
19
20             return null;
21         }
22     }
23 }

```

Pada line 13 -14 **db\_siswa** adalah nama database yang tadi kita buat, sesuaikan dengan nama database yang kalian buat.

### Membuat Form Siswa

yang pertama kalian lakukan adalah membuat package baru dengan nama view, dan klik kanan pada package tersebut new -> JFrame Form dan beri nama **FormSiswa**:



Sekarang saatnya membuat tampilan dari form nya, kita akan menambahkan sebuah **label**, **textfield**, **textarea**, **button**, **comboBox** dan **table**:

## Form Siswa

NIS

Nama

Jurusan Rekayasa Perangkat Lunak ▼

JK Laki-laki ▼

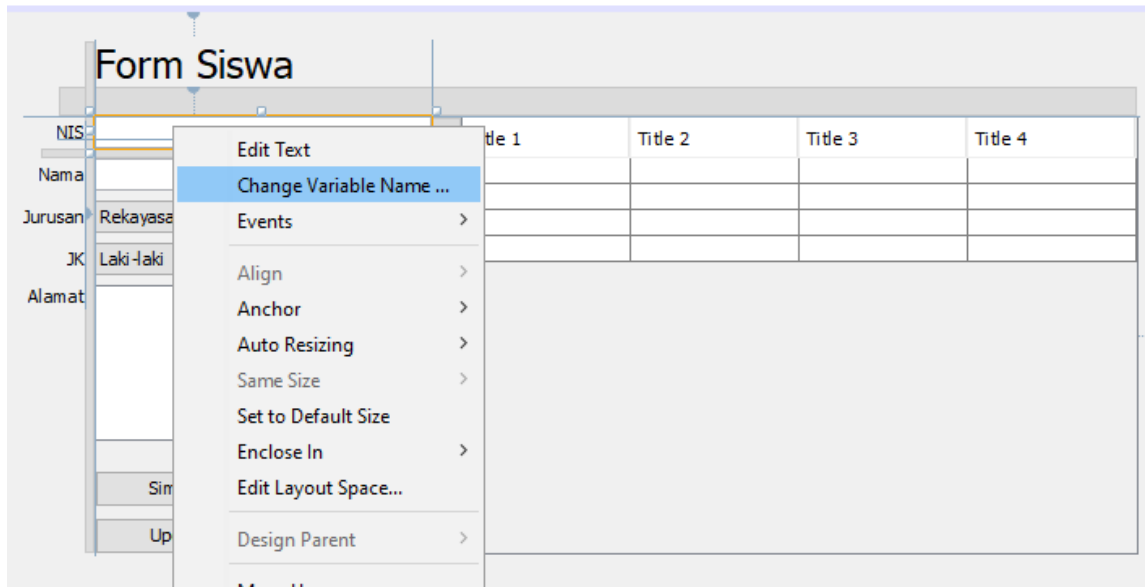
Alamat

Simpan
Hapus

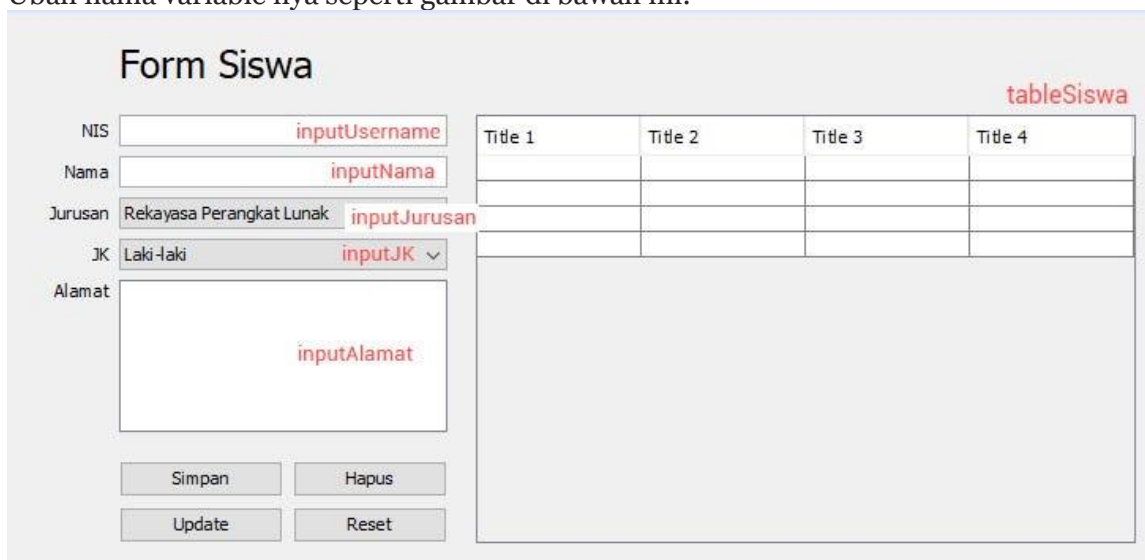
Update
Reset

Title 1	Title 2	Title 3	Title 4

Dan setelah itu kalian ganti nama varilbelnya dengan cara klik kanan lalu **change variable name**:



Ubah nama variable nya seperti gambar di bawah ini:



Kemudian klik kanan pada **inputJurusan** dan **inputJK** -> klik properties -> klik pada tombol dengan “titik tiga” disebelah kanan model kemudian masukkan jurusan yang akan di-input user.

### FormSiswa.java

Klik pada tab **source** dibawah tab file FormSiswa.java masukkan syntax berikut dibawah **package tutorial.java;**

```
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

Kemudian deklarasikan variable seperti dibawah ini

```
public class FormSiswa extends javax.swing.JFrame {  
    public Statement st;  
    public ResultSet rs;  
    public DefaultTableModel tabModel;  
    Connection cn = koneksi.Koneksi.Koneksi();  
....
```

## Menampilkan Data ke Dalam Tabel

```
public void judul() {  
    Object[] judul = {  
        "NIS", "Nama", "Jurusan", "JK", "Alamat"  
    };  
    tabModel = new DefaultTableModel(null, judul);  
    tableSiswa.setModel(tabModel);  
}  
  
public void tampilData(String where) {  
    try {  
        st = cn.createStatement();  
        tabModel.getDataVector().removeAllElements();  
        tabModel.fireTableDataChanged();  
        rs = st.executeQuery("SELECT * FROM students " + where);  
  
        while (rs.next()) {  
            Object[] data = {  
                rs.getString("nis"),  
                rs.getString("nama"),  
                rs.getString("jurusan"),  
                rs.getString("jk"),  
                rs.getString("alamat"),  
            };  
            tabModel.addRow(data);  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Deklarasikan class **tampilData()** & **judul()** ke dalam class **FormSiswa**, contohnya seperti ini:

```
public FormSiswa() {  
    initComponents();  
    judul();  
    tampilData("");  
    buttonHapus.setEnabled(false);  
    buttonUpdate.setEnabled(false);  
}
```

## Membuat Fungsi Simpan

Pertama double click pada button **buttonSimpan**, lalu masukan kode seperti bawah ini:

```
private void buttonSimpanActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        st = cn.createStatement();  
        st.executeUpdate("INSERT INTO students VALUES(' + inputNis.getText() + "','"  
            + inputNama.getText() + "','"  
            + inputJurusan.getSelectedItem() + "','"  
            + inputJK.getSelectedItem() + "','"  
            + inputAlamat.getText() + "')");  
        tampilData("");  
        JOptionPane.showMessageDialog(null, "Simpan Berhasil");  
        inputNis.setText("");  
        inputNama.setText("");  
        inputAlamat.setText("");  
        inputJurusan.setSelectedItem("");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Silahkan run project kalian untuk mencoba fungsi simpan **Shift + F6**. Jika berhasil akan muncul window message "Simpan Berhasil".

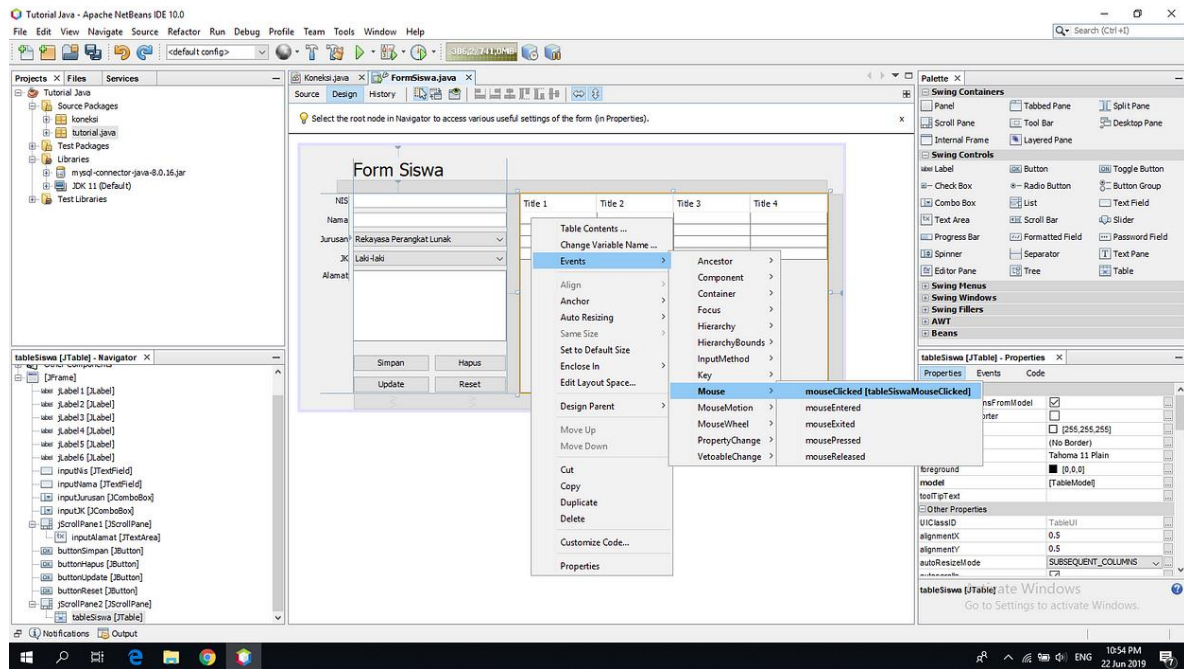
## Persiapan Untuk Membuat Fungsi Update

Pada bagian ini kita akan membuat aplikasi kita pada saat klik datanya, maka data tersebut akan berpindah ke bagian textfield

NIS	Nama	Jurusan	JK	Alamat
11705978	Ariq Heritsa	rangkat Lunak	Laki-laki	Jl. Jalan Ke...

klik kanan pada table -> Events -> Mouse -> MouseClicked





Masukkan kode berikut:

```
private void tableSiswaMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    inputNis.setText(tableSiswa.getValueAt(tableSiswa.getSelectedRow(), 0).toString());
    inputNama.setText(tableSiswa.getValueAt(tableSiswa.getSelectedRow(), 1).toString());
    inputJurusan.setSelectedItem(tableSiswa.getValueAt(tableSiswa.getSelectedRow(), 2).toString());
    inputJK.setSelectedItem(tableSiswa.getValueAt(tableSiswa.getSelectedRow(), 3).toString());
    inputAlamat.setText(tableSiswa.getValueAt(tableSiswa.getSelectedRow(), 4).toString());
    buttonSimpan.setEnabled(false);
    buttonUpdate.setEnabled(true);
    buttonHapus.setEnabled(true);
}
```

## Membuat Fungsi Reset

Fungsi ini adalah untuk mengosongkan semua textfield

```
public void reset () {
    inputNis.setText("");
    inputNama.setText("");
    inputJurusan.setSelectedItem("");
    inputJK.setSelectedItem("");
    inputAlamat.setText("");
}
```

Panggil fungsi tersebut pada **buttonReset**

```
private void buttonResetActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    reset();  
}
```

## Membuat Fungsi Update

```
private void buttonUpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        st = cn.createStatement();  
        st.executeUpdate("UPDATE students set "  
            + "nis='"      + inputNis.getText() + "', "  
            + "nama='"    + inputNama.getText() + "', "  
            + "jurusan='" + inputJurusan.getSelectedItem() + "', "  
            + "jk='"      + inputJK.getSelectedItem() + "', "  
            + "alamat='"  + inputAlamat.getText() + "'");  
        tampilData("");  
        JOptionPane.showMessageDialog(null, "Update Berhasil");  
        reset();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

## Membuat Fungsi Delete

```
private void buttonHapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        int jawab;  
  
        if ((jawab = JOptionPane.showConfirmDialog(null, "Ingin menghapus "  
            + "data?", "konfirmasi", JOptionPane.YES_NO_OPTION)) == 0)  
        {  
            st = cn.createStatement();  
            st.executeUpdate("DELETE FROM students WHERE nis='"  
                + tabModel.getValueAt(tableSiswa.getSelectedRow(), 0) + "'");  
            tampilData("");  
            reset();  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```