

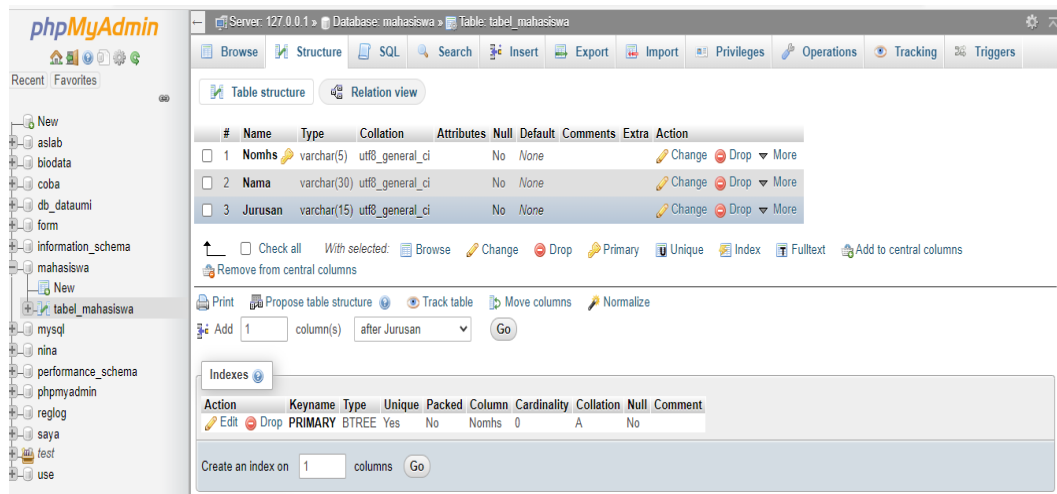
Nama : Nisrina Dwi Ambarwati

Stambuk : 13020180097

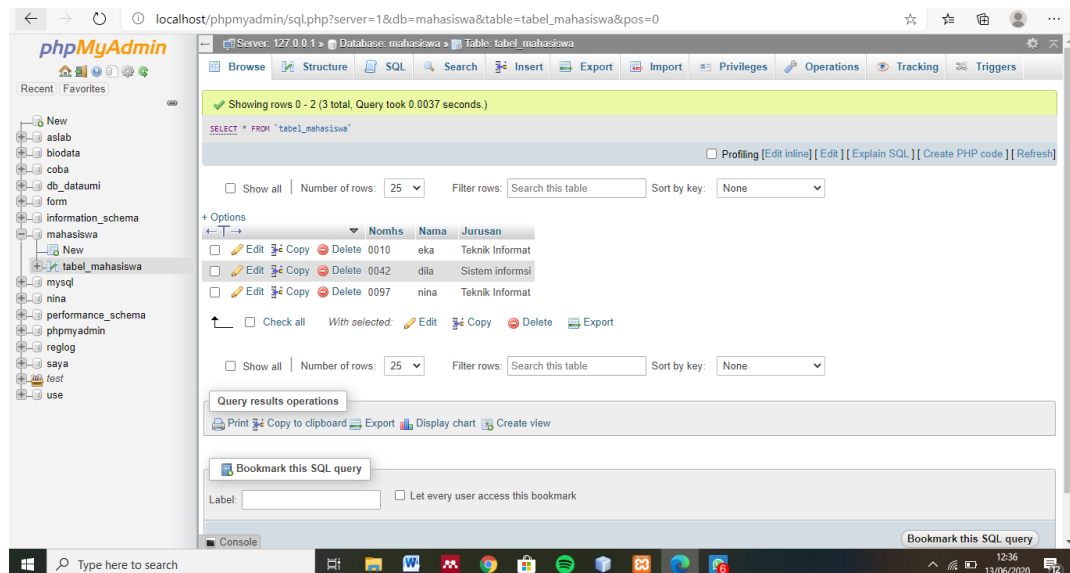
Kelas : B2

## 1. Membuat database

### a. Database mahasiswa

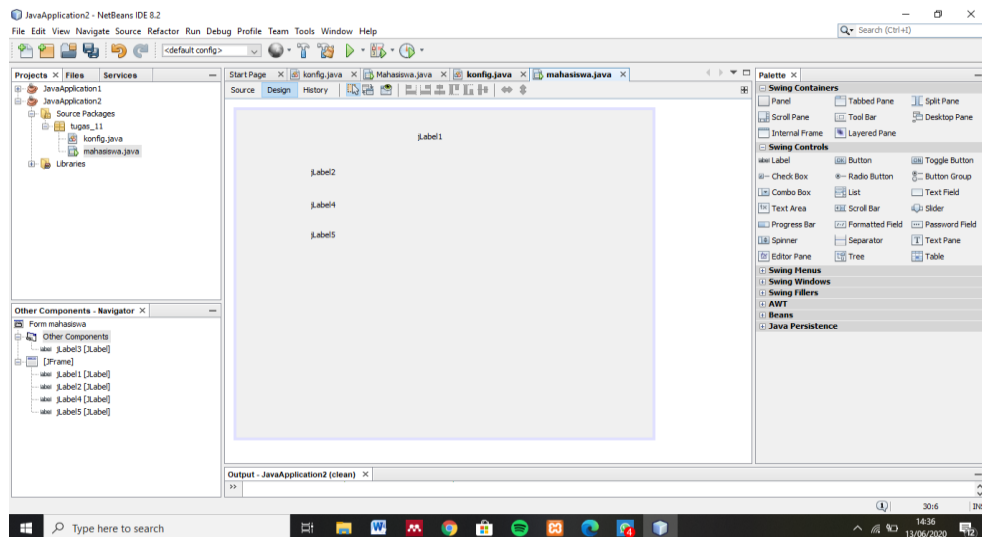


### b. Pembuatan table



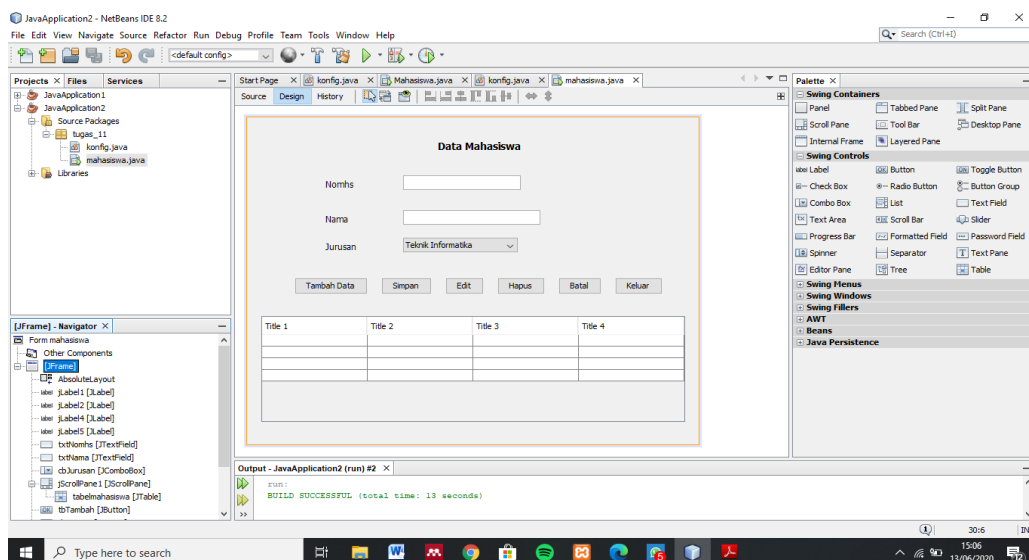
## 2. Tampilan awal form

### a. Pembuatan judul untuk form data mahasiswa

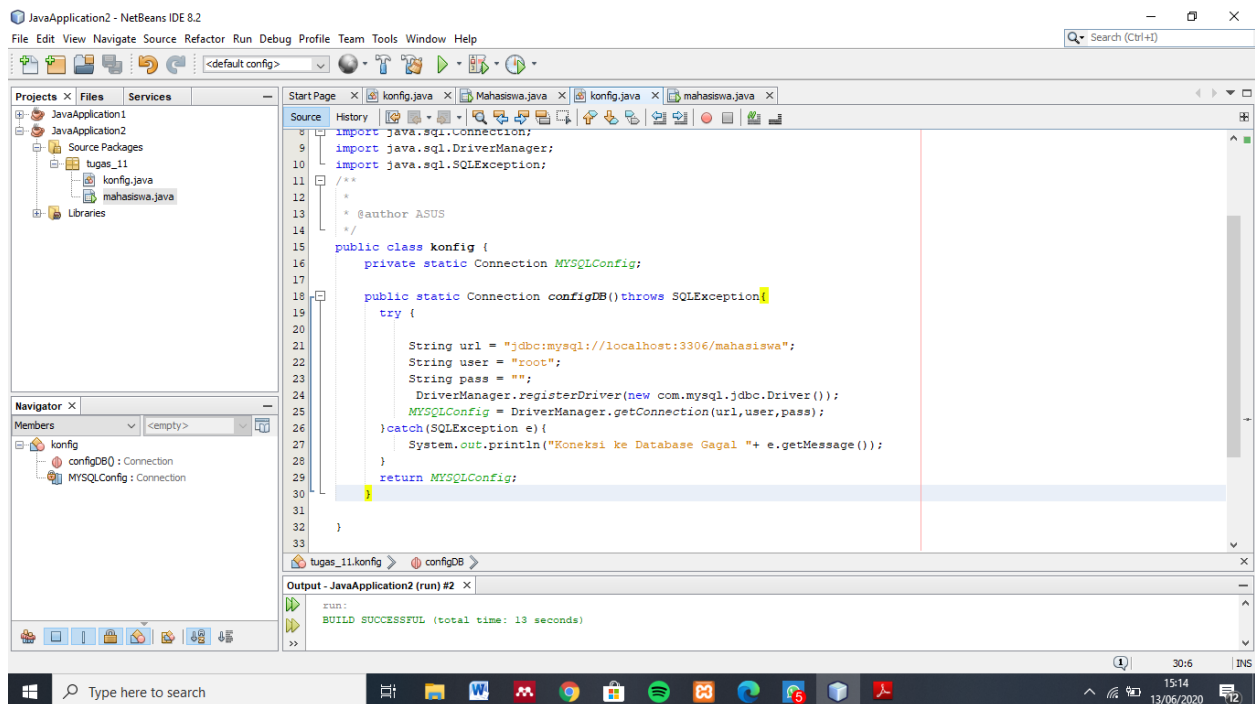


### b. Ganti nama sesuai data dari database yg dibuat dan jangan lupa ubah untuk memberi nama, di kotak kiri bawah agar lebih mudah ketika di panggil.

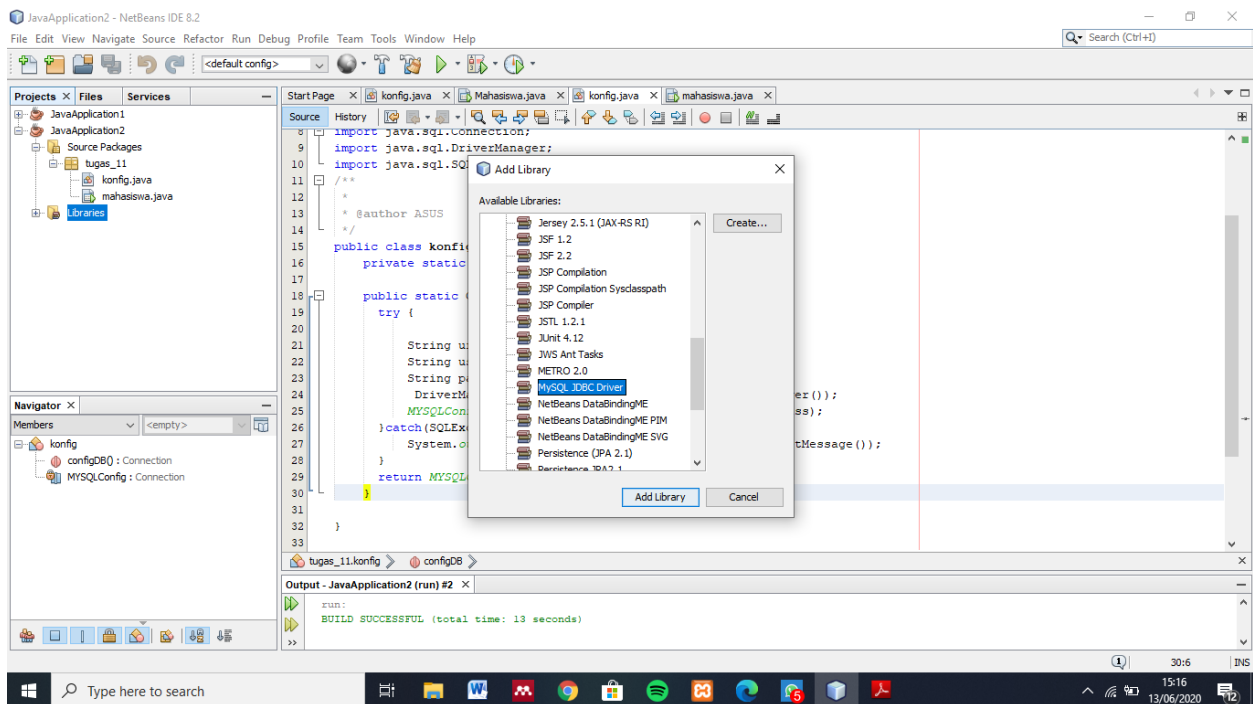
- Kemudian pada bagian jurusan saya membuat 2 jurusan yang dapat dipilih yaitu Teknik Informatika dan Sistem Inofrmasi.
- Pembuatan Tombol untuk Form
- Pembuatan tabel untuk menampilkan data yang telah diinput di form



### 3. Membuat File konfigurasi agar bisa terhubung dengan database



- Tambahkan bagian mysql JDBC driver agar dapat terhubung dengan database.



#### 4. Pembuatan tambahan agar desain yang telah dibuat dapat berfungsi

- a. Bagian Tombol Tambah data, berfungsi untuk mengosongkan form, dan kemudian di isi dan di simpan

```
private void tabel_mahasiswaMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    int baris = tabel_mahasiswa.rowAtPoint(evt.getPoint());  
    String nomhs = tabel_mahasiswa.getValueAt(baris, 1).toString();  
    txtNomhs.setText(nomhs);  
  
    String nama = tabel_mahasiswa.getValueAt(baris, 2).toString();  
    txtNama.setText(nama);  
  
    String jur = tabel_mahasiswa.getValueAt(baris, 3).toString();  
    cbJurusan.setSelectedItem(jur);  
}
```

- b. Kemudian menambahkan beberapa procedure untuk mengosongkan form nya(method) berfungsi untuk mengosongkan tekstil tekstil yang telah di buat .

```
public class mahasiswa extends javax.swing.JFrame {  
    private void kosongkan_form() {  
        txtNomhs.setEditable(true);  
        txtNomhs.setText(null);  
        txtNama.setText(null);  
        cbJurusan.setSelectedItem(this);  
    }  
}
```

- c. Kemudian menambahkan source untuk menampilkan data di dalam table

```
private void tampilkan_data() {  
  
    DefaultTableModel model = new DefaultTableModel();  
    model.addColumn("No.");  
    model.addColumn("Nomhs");  
    model.addColumn("Nama");  
    model.addColumn("Jurusan");  
}
```

- d. Membuat data untuk di tampilkan dari database ke dalam table

```
    } catch (SQLException e) {  
        System.out.println("Error: " + e.getMessage());  
    }  
}
```

```

private void tampilkan_data(){

    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("No.");
    model.addColumn("Nomhs");
    model.addColumn("Nama");
    model.addColumn("Jurusan");

    try{
        int no = 1;
        String sql = "SELECT * FROM mahasiswa";
        java.sql.Connection conn = (Connection)konfig.configDB();
        java.sql.Statement stm = conn.createStatement();
        java.sql.ResultSet res = stm.executeQuery(sql);

        while(res.next()){
            while(res.next()){
                model.addRow(new Object[]{no++,res.getString(1),res.getString(2),res.getString(3)});
            }
        }
        tabelmahasiswa.setModel(model);
    }catch (SQLException e){
        System.out.println("Error: " + e.getMessage());
    }
}

```

e. Untuk menampilkan data

```

    */
    public mahasiswa() {
        initComponents();
        tampilkan_data();
    }

```

f. Bagian Tombol Simpan, berfungsi untuk menyimpan data yang kita input dan mengarahkannya ke tabel yang telah kita siapkan. Tombol simpan variabelnya bsimpan.

```

ate void tbSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "INSERT INTO tabel_mahasiswa VALUES ('"+txtNomhs.getText()+"','"+txtNama.getText()+"','"+cbJurusan.getS
        java.sql.Connection conn = (Connection)konfig.configDB();
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(null, "Proses Simpan Data Berhasil..");
        tampilkan_data();
        kosongkan_form();
    }catch (HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

```

- g. Bagian Tombol Edit, berfungsi mengedit data yang telah diinputkan sebelumnya. Tombol edit variabelnya bedit.

```
private void tbEditActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{  
        String sql = "UPDATE tabel_mahasiswa SET nomhs='"+txtNomhs.getText()+"',nama='"+txtNama.getText()+"',"  
            + "'"+cbJurusan.getSelectedItem()+"' WHERE nomhs = '"+txtNomhs.getText()+"'";  
        java.sql.Connection conn = (Connection)konfig.configDB();  
        java.sql.PreparedStatement pstm = conn.prepareStatement(sql);  
        pstm.execute();  
        JOptionPane.showMessageDialog(null, "Edit Data Berhasil...");  
    }catch (HeadlessException | SQLException e){  
        JOptionPane.showMessageDialog(this, e.getMessage());  
    }  
    tampilkan_data();  
    kosongkan_form();  
}
```

- h. Bagian Tombol Hapus, berfungsi untuk menghapus data yang telah diinputkan sebelumnya. Variablenya bhapus.

```
private void tbHapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{  
        String sql = "DELETE FROM tabel_mahasiswa WHERE nomhs='"+txtNomhs.getText()+"'";  
        java.sql.Connection conn = (Connection)konfig.configDB();  
        java.sql.PreparedStatement pstm = conn.prepareStatement(sql);  
        pstm.execute();  
        JOptionPane.showMessageDialog(null, "Hapus Data Berhasil...");  
    }catch (HeadlessException | SQLException e){  
        JOptionPane.showMessageDialog(this, e.getMessage());  
    }  
    tampilkan_data();  
    kosongkan_form();  
}
```

- i. Mengubungkan agar ketika mengklik data yang telah diinputkan pada tabel maka akan terhubung pada bagian form.

```
private void tabel_mahasiswaMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    int baris = tabel_mahasiswa.rowAtPoint(evt.getPoint());  
    String nomhs = tabel_mahasiswa.getValueAt(baris, 1).toString();  
    txtNomhs.setText(nomhs);  
  
    String nama = tabel_mahasiswa.getValueAt(baris, 2).toString();  
    txtNama.setText(nama);  
  
    String jur = tabel_mahasiswa.getValueAt(baris, 3).toString();  
    cbJurusan.setSelectedItem(jur);  
}
```

- j. Bagian Tombol Keluar, berfungsi untuk keluar dari program. Variabelnya bkeluar

```
+ Generated Code
- private void tbKeluarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}
```

## Output :

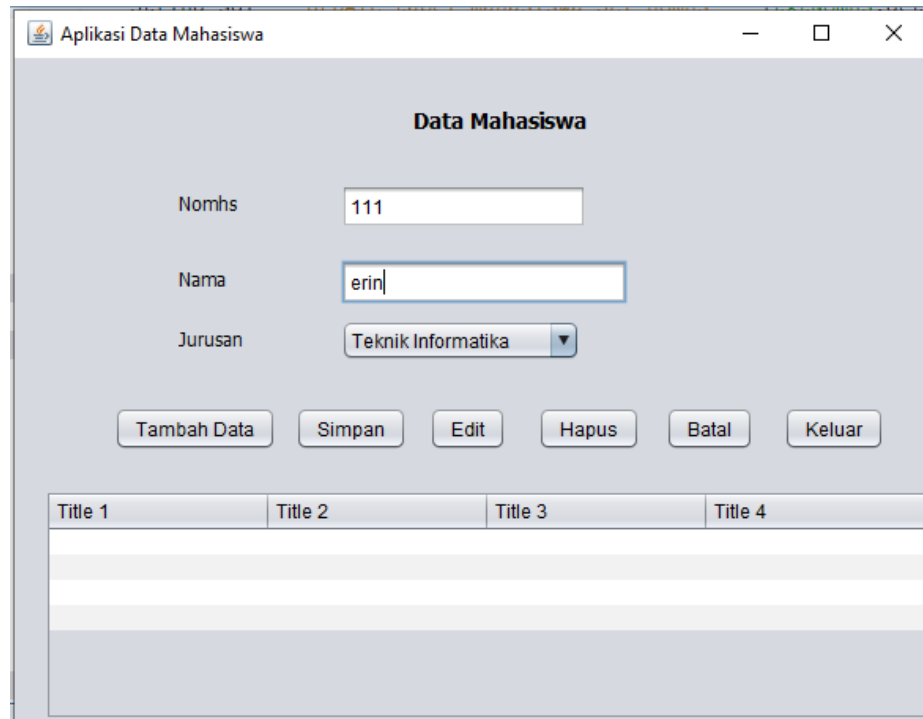
Tampilan Awal

The screenshot shows a Java Swing window titled "Aplikasi Data Mahasiswa". The window has a title bar with standard minimize, maximize, and close buttons. The main content area is titled "Data Mahasiswa" and contains a form with three input fields: "Nomhs" (a single-line text field), "Nama" (a multi-line text area), and "Jurusan" (a dropdown menu currently showing "Teknik Informatika"). Below the form are six buttons: "Tambah Data", "Simpan", "Edit", "Hapus", "Batal", and "Keluar". At the bottom of the window is a table with four columns labeled "Title 1", "Title 2", "Title 3", and "Title 4". The table has three empty rows below the header.

Title 1	Title 2	Title 3	Title 4

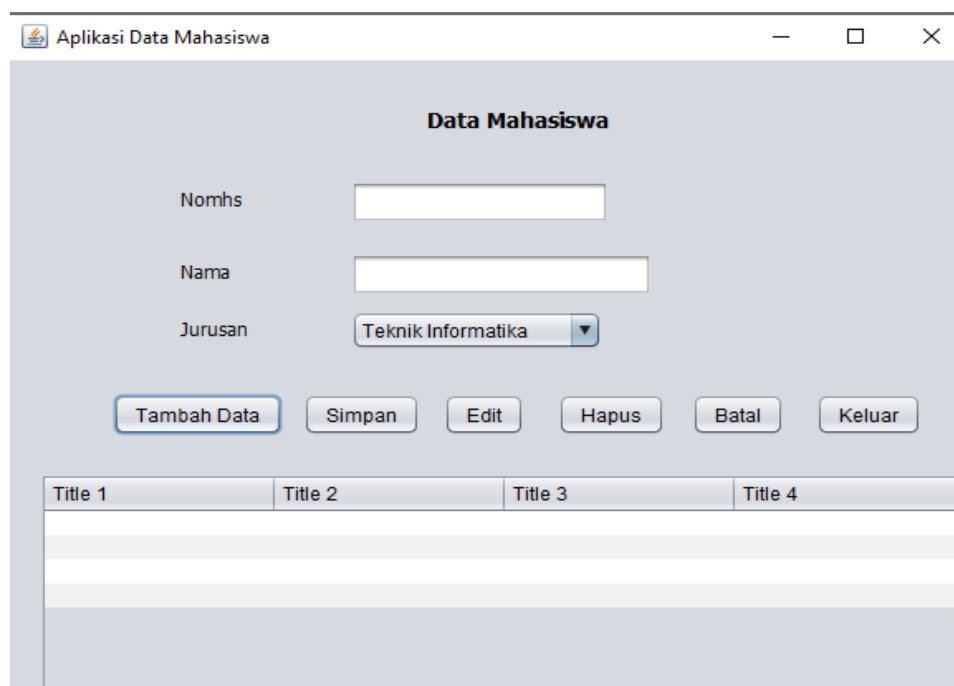
### a. Bagian Tombol Tambah Data

Di tombol Tambah ini kita masukan data.



The screenshot shows a window titled "Aplikasi Data Mahasiswa". Inside, there's a section titled "Data Mahasiswa". It contains three input fields: "Nomhs" with the value "111", "Nama" with the value "erin", and "Jurusan" with a dropdown menu showing "Teknik Informatika". Below these fields are six buttons: "Tambah Data", "Simpan", "Edit", "Hapus", "Batal", and "Keluar". At the bottom, there's a table with four columns labeled "Title 1", "Title 2", "Title 3", and "Title 4". The table has three empty rows below the header.

Kemudian apa bila kita menekan tombol 'tambah data' otomatis form nya akan kosong



The screenshot shows the same window as before, but the "Tambah Data" button is now highlighted with a blue border. The input fields for "Nomhs", "Nama", and "Jurusan" are now empty. The table at the bottom remains the same.



kemudian masukan data lalu klik simpan,

**Aplikasi Data Mahasiswa**

**Data Mahasiswa**

Nomhs: 0123

Nama: dita

Jurusan: Teknik Informatika

Tambah Data Simpan Edit Hapus Batal Keluar

Title 1	Title 2	Title 3	Title 4

kemudian data yang di masukan itu tersimpan di phpMyAdmin

localhost/phpmyadmin/sql.php?db=mahasiswa&table=tabel\_mahasiswa&pos=0

Server: 127.0.0.1 x Database: mahasiswa x Table: tabel\_mahasiswa

Showing rows 0 - 2 (3 total, Query took 0.0036 seconds)

```
SELECT * FROM 'tabel_mahasiswa'
```

Options: Nomhs Nama Jurusan

<input type="checkbox"/>	Edit	Copy	Delete	001	eka	Sistem Informasi
<input type="checkbox"/>	Edit	Copy	Delete	0010	dila	Teknik Informatika
<input type="checkbox"/>	Edit	Copy	Delete	0123	dita	Teknik Informatika

Query results operations: Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label:  ☐ Let every user access this bookmark

## b. Bagian Tombol Simpan

Masukan data, lalu klik tombol Simpan.

**Aplikasi Data Mahasiswa**

**Data Mahasiswa**

Nomhs: 0123

Nama: dita

Jurusan: Teknik Informatika

Tambah Data Simpan Edit Hapus Batal Keluar

Title 1	Title 2	Title 3	Title 4

Maka data yang di masukan, akan tersimpan di phpmyadmin

localhost/phpmyadmin/sql.php?db=mahasiswa&table=tabel\_mahasiswa&pos=0

Server: 127.0.0.1 Database: mahasiswa Table: tabel\_mahasiswa

Showing rows 0 - 2 (3 total. Query took 0.0036 seconds.)

```
SELECT * FROM `tabel_mahasiswa`
```

Options: Nomhs Nama Jurusan

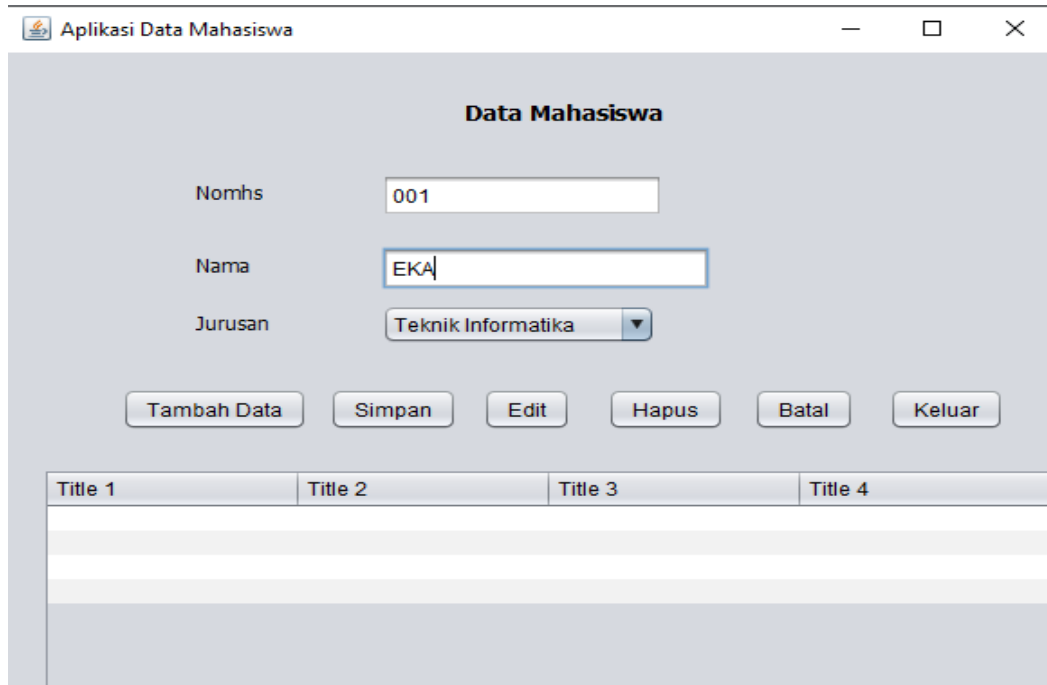
	Nomhs	Nama	Jurusan
<input type="checkbox"/>	001	eka	Sistem Informasi
<input type="checkbox"/>	0010	dila	Teknik Informatika
<input type="checkbox"/>	0123	dita	Teknik Informatika

Query results operations: Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query: Label:  Let every user access this bookmark

### c. Bagian Tombol Edit

Di tombol edit ini kita masukan data '001, EKA, Teknik Informatika'



**Aplikasi Data Mahasiswa**

**Data Mahasiswa**

Nomhs: 001

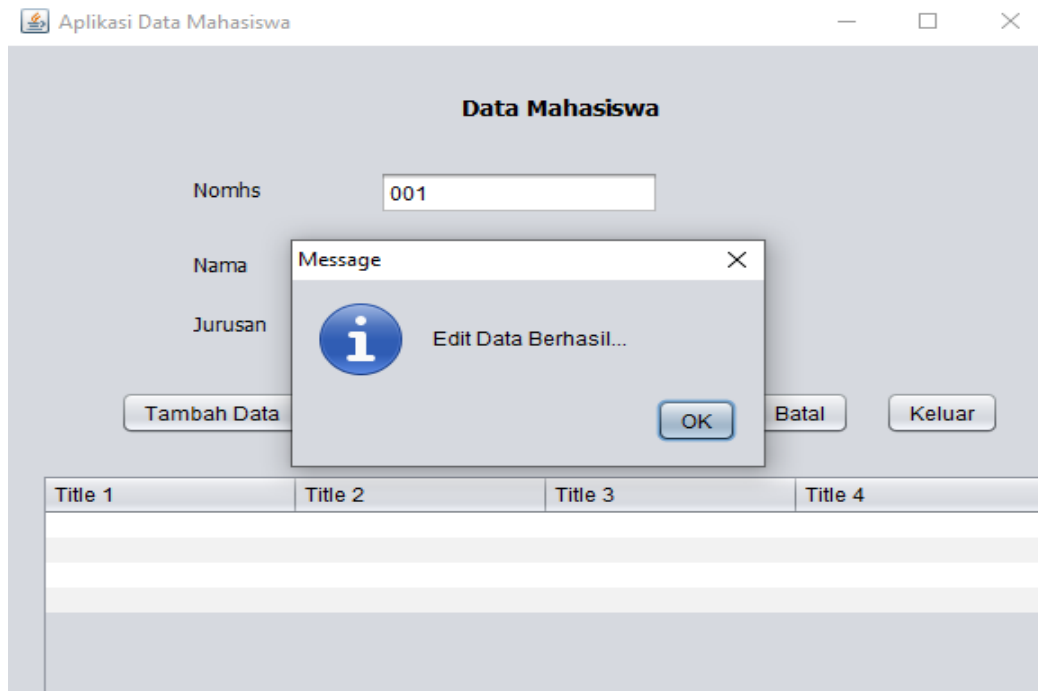
Nama: EKA

Jurusan: Teknik Informatika

Tambah Data Simpan Edit Hapus Batal Keluar

Title 1	Title 2	Title 3	Title 4

Kemudian klik tombol edit maka akan muncul 'edit data berhasil' lalu Oke



**Aplikasi Data Mahasiswa**

**Data Mahasiswa**

Nomhs: 001

Nama: EKA

Jurusan: Teknik Informatika

Tambah Data Simpan Edit Hapus Batal Keluar

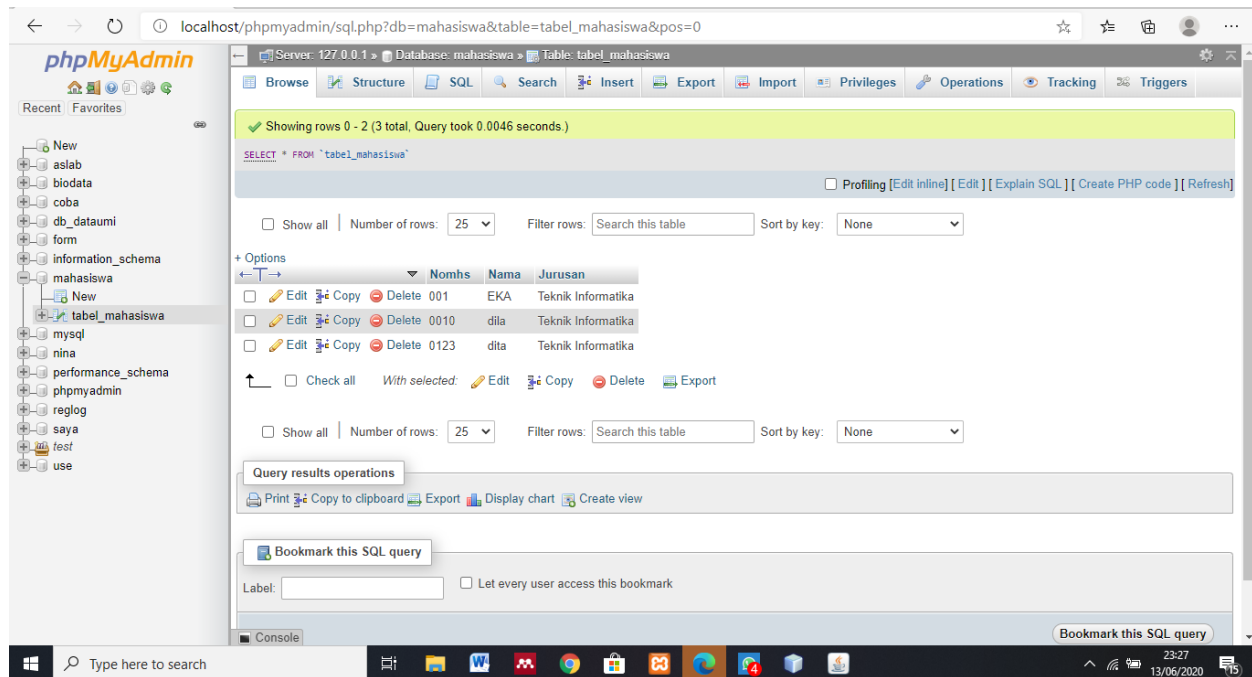
**Message**

**i** Edit Data Berhasil...

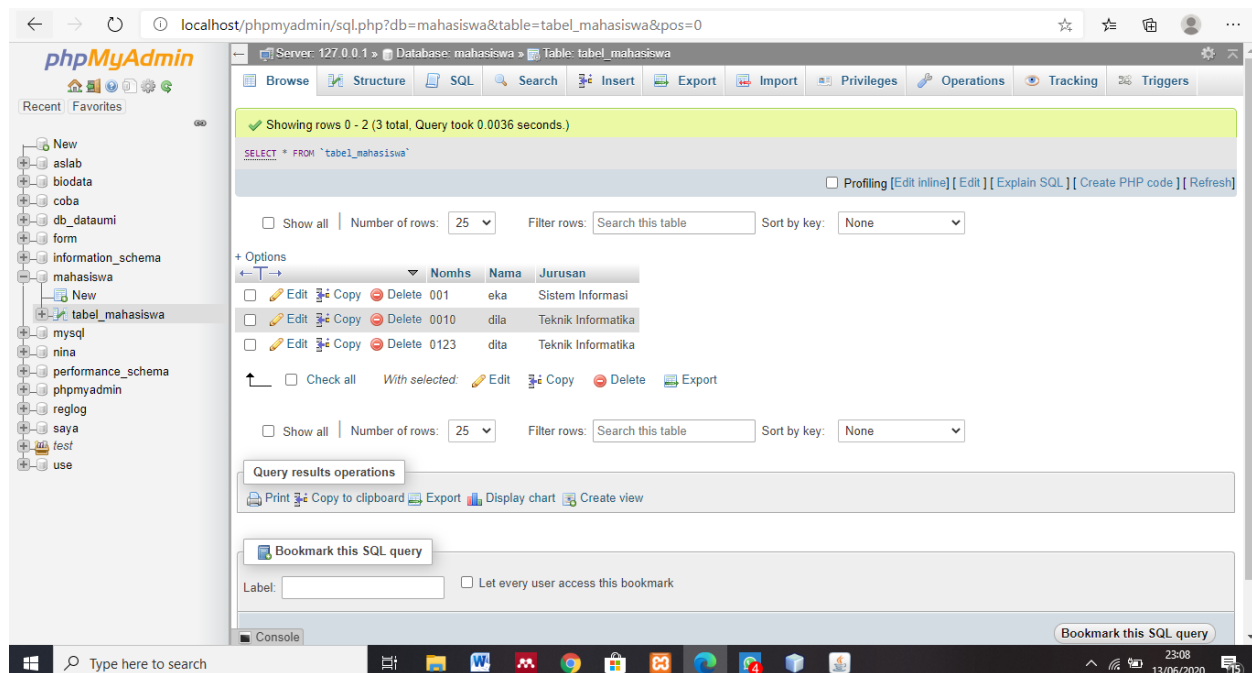
OK Batal Keluar

Title 1	Title 2	Title 3	Title 4

Maka data yang sudah kita masukkan tadi akan berubah di phpMyAdmin menjadi '001, EKA, Teknik informatika'



Yang tadi nya data sebelum di edit '001, eka, Sistem informasi'



d. Bagian Tombol Hapus

The screenshot shows a window titled "Aplikasi Data Mahasiswa". Inside, there's a section titled "Data Mahasiswa" with three input fields: "Nomhs" (empty), "Nama" (empty), and "Jurusan" (set to "Teknik Informatika"). Below these are six buttons: "Tambah Data", "Simpan", "Edit", "Hapus" (highlighted with a blue border), "Batal", and "Keluar". At the bottom is a table with four columns labeled "Title 1", "Title 2", "Title 3", and "Title 4". The table has three empty rows below the header.

e. Bagian Tombol Batal

Pertama masukan data

This screenshot shows the same "Aplikasi Data Mahasiswa" window, but now with data entered. The "Nomhs" field contains "3111", the "Nama" field contains "Dwi", and the "Jurusan" dropdown is still set to "Teknik Informatika". The "Hapus" button is no longer highlighted. The table at the bottom remains empty with three rows below the header.

Kemudian apabila kita menekan tombol batal maka formnya akan kosong.

**Aplikasi Data Mahasiswa**

**Data Mahasiswa**

Nomhs

Nama

Jurusan

Title 1	Title 2	Title 3	Title 4

#### f. Bagian Tombol Keluar

Apa bila kita menekan tombol Keluar, maka kita akan keluar dari form.

**Aplikasi Data Mahasiswa**

**Data Mahasiswa**

Nomhs

Nama

Jurusan

Title 1	Title 2	Title 3	Title 4