

Tugas Final

Makassar, 20 Agustus 2020

## **PEMROGRAMAN**

## **JAVA LANJUT**

**“ APLIKASI BIODATA MAHASISWA ”**



Nama : Abdul Alfian Suri Tarif

Nim : 13020160015

Kelas : C2

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MUSLIM INDONESIA**

**MAKASSAR**

**2020**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Suatu institut Pendidikan mengemban tugas penting dalam pendidikan dan pengembangan keilmuan serta karakter para generasi muda di Negara ini. Dalam menjalankan tugas-tugasnya sebuah institut sudah sepatutnya melakukan pengelolaan yang baik. Tidak hanya mengelola berbagai sumber dayanya seperti keuangan, fasilitas, dan karyawannya, melainkan juga mengelola objek daripada mengapa institut Pendidikan tersebut berjalan, yakni mahasiswanya.

Sebuah institut Pendidikan perlu melakukan pengelolaan data mahasiswa dengan baik demi memudahkan mereka dalam menjalankan tugasnya. Untuk itu perlu adanya sebuah aplikasi yang berguna untuk menampung semua data-data mahasiswa yang ada, agar terus dapat dievaluasi secara berkala.

### **1.2 Rumusan Masalah**

A. Membuat Program Java GUI berbasis MVC menggunakan database

### **1.3 Tujuan dan Manfaat**

#### **1.3.1 Tujuan**

A. Membuat Aplikasi Biodata Mahasiswa menggunakan Java GUI berbasis MVC dengan menggunakan database

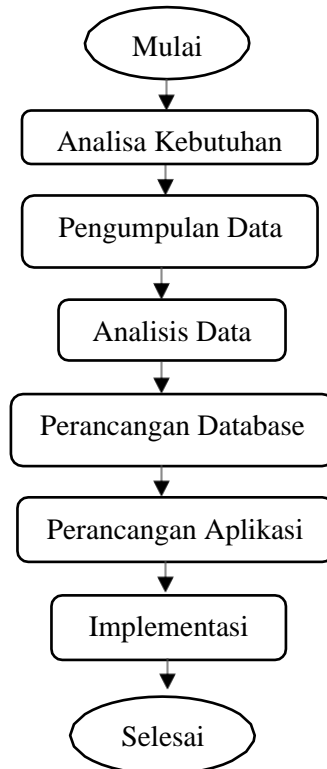
#### **1.3.2 Manfaat**

A. Memudahkan suatu institut Pendidikan dalam melakukan pengelolaan database mahasiswa

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Analisi dan Desain Prosedur**



##### **A. Analisa kebutuhan**

Kegiatan yang dilakukan adalah menganalisis kebutuhan penelitian mulai dari objek yang akan digunakan sebagai pengujian yaitu Biodata Mahasiswa

##### **B. Pengumpulan data**

Pengumpulan data dan informasi sebagai bahan dalam pembuatan database dan aplikasi

##### **C. Analisis data**

Pada tahap ini data yang telah dikumpulkan kemudian diklasifikasikan berdasarkan variabelnya, agar sesuai dengan kebutuhan aplikasi

##### **D. Pembuatan database**

Pada pembuatan database ialah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung jalannya aplikasi

#### E. Pembuatan aplikasi

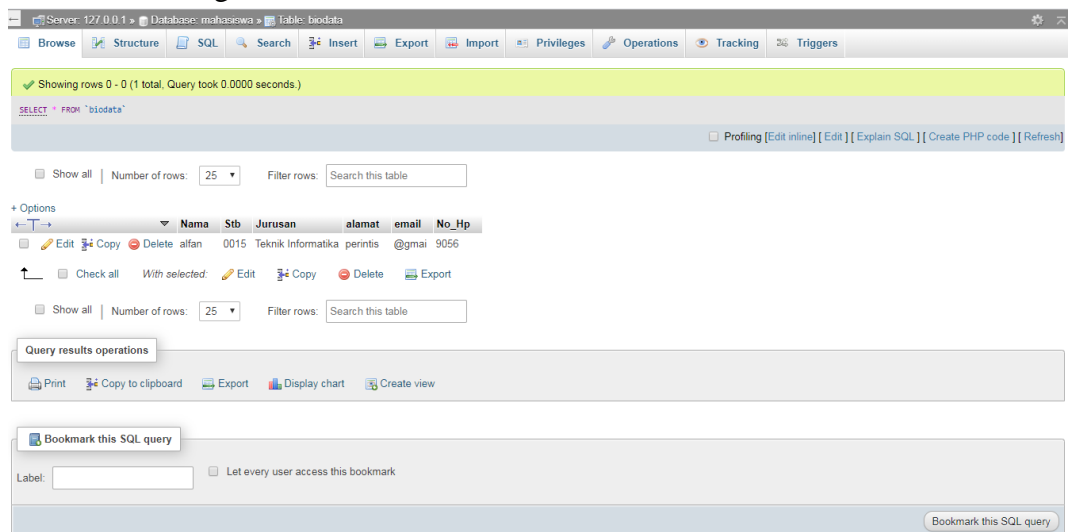
Pada pembuatan aplikasi yang dilakukan adalah mendefinisikan spesifikasi kebutuhan pengguna, kebutuhan sistem dan membuat perancangan dari sistem yang akan dibuat.

#### F. Implementasi

Setelah pembuatan aplikasi dilakukan, maka selanjutnya adalah menjalankan aplikasinya untuk memastikan apakah aplikasi tersebut bekerja sesuai yang diinginkan

### 2.2 Tampilan Database

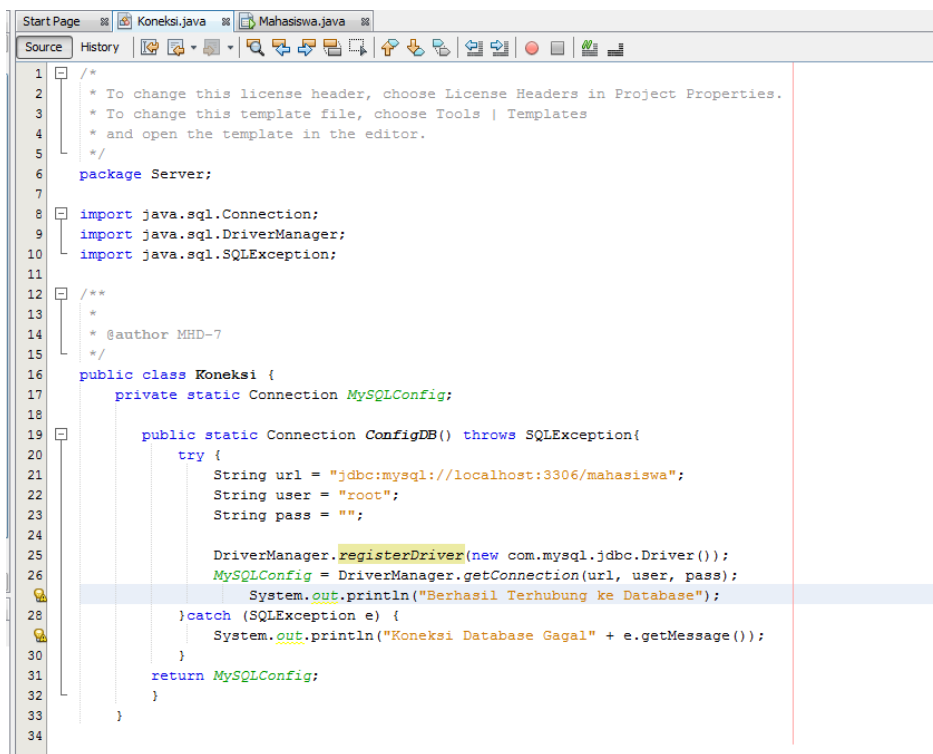
Ini adalah tampilan database sebelum pembuatan aplikasi. Pembuatan database menggunakan tools phpMyAdmin. Nama database ini adalah Mahasiswa, dengan nama tabel Biodata.



## 2.3 Kode Program dan Implementasi Aplikasi

### A. Class Koneksi

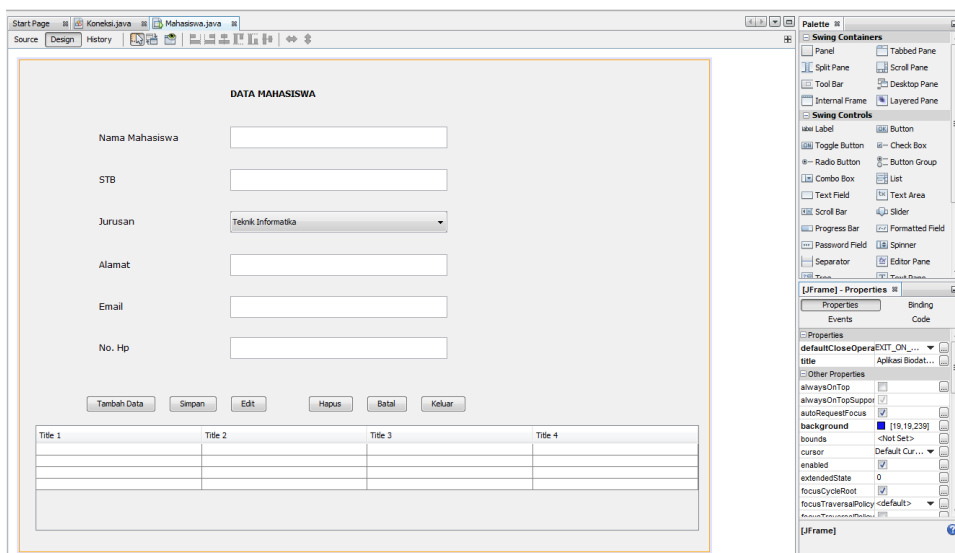
1. Kelas ini berfungsi untuk menghubungkan program ke database yang telah dibuat



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package Server;
7
8  import java.sql.Connection;
9  import java.sql.DriverManager;
10 import java.sql.SQLException;
11
12 /**
13 *
14 * @author MHD-7
15 */
16 public class Koneksi {
17     private static Connection MySQLConfig;
18
19     public static Connection ConfigDB() throws SQLException{
20         try {
21             String url = "jdbc:mysql://localhost:3306/mahasiswa";
22             String user = "root";
23             String pass = "";
24
25             DriverManager.registerDriver(new com.mysql.jdbc.Driver());
26             MySQLConfig = DriverManager.getConnection(url, user, pass);
27             System.out.println("Berhasil Terhubung ke Database");
28         } catch (SQLException e) {
29             System.out.println("Koneksi Database Gagal" + e.getMessage());
30         }
31         return MySQLConfig;
32     }
33 }
34
```

### B. Tampilan Form awal

Inilah desain awal untuk aplikasi nantinya.



DATA MAHASISWA

Nama Mahasiswa

STB

Jurusan

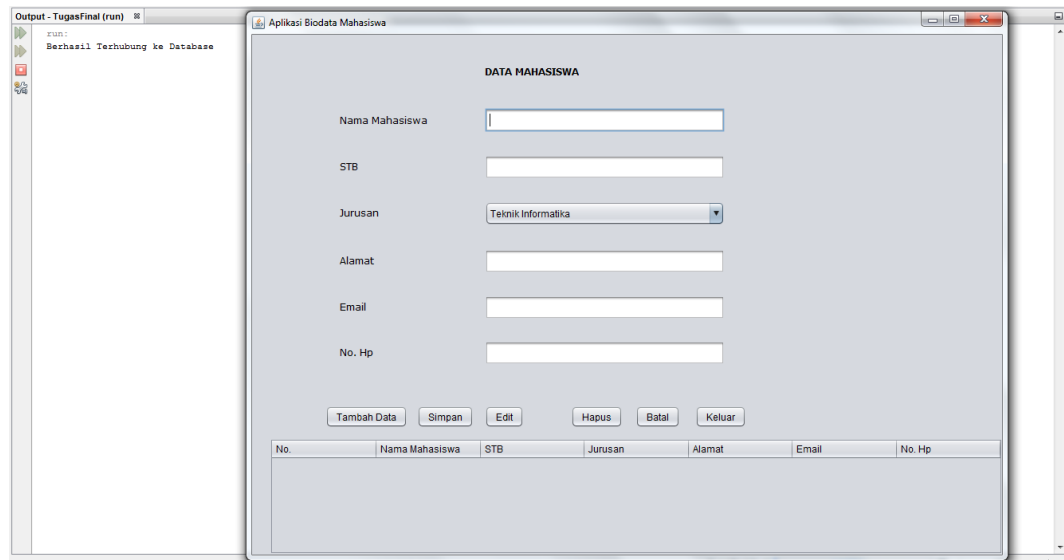
Alamat

Email

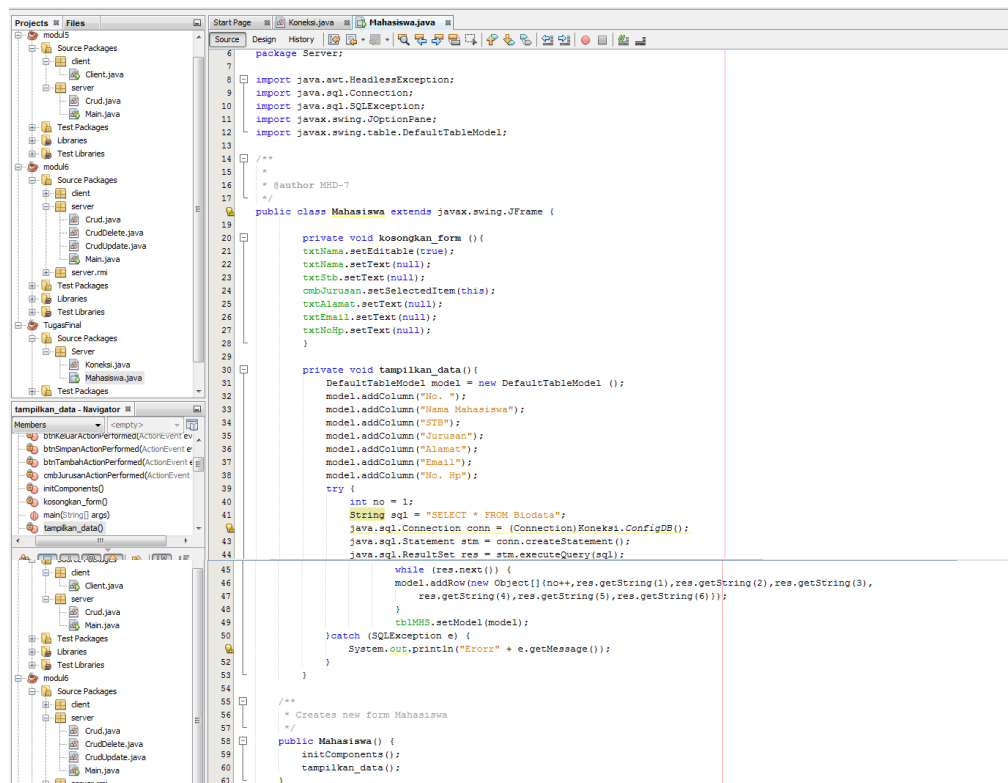
No. Hp

Title 1	Title 2	Title 3	Title 4

1. Tampilan setelah aplikasi telah berjalan dan telah terhubung ke database  
Pada tahap ini aplikasi telah terhubung ke database. Namun belum dapat dipakai karena tombol pada aplikasi belum diberi codingan. Ini hanyalah contoh jika desain awal aplikasi telah siap dan terhubung ke database. Selanjutnya akan diberikan codingan pada setiap tombol pada aplikasi.



## 2. Codingan dari Program



## C. Codingan dari tombol “Simpan”

1. Ini adalah source code dari menu simpan.

```
private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String sql = "INSERT INTO biodata VALUES ('"+txtNama.getText()+"','"+txtStb.getText()+"  
        "','"+cmbJurusan.getSelectedText()+"','"+txtAlamat.getText()+"','"+txtEmail.getText()+"','"+txtNoHp.getText()+"')";  
        java.sql.Connection conn = (Connection)Koneksi.ConfigDB();  
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
        pstmt.executeUpdate();  
        JOptionPane.showMessageDialog(null, "Proses Simpan Data Berhasil");  
        tampilkan_data();  
        kosongkan_form();  
    } catch (HeadlessException | SQLException e) {  
        JOptionPane.showMessageDialog(this, e.getMessage());  
    }  
}
```

2. Ini adalah tampilan ketika data mahasiswa telah berhasil diinput.

The screenshot shows the 'Aplikasi Biodata Mahasiswa' window. The form fields are filled with: Nama Mahasiswa: alfan, STB: 0015, Jurusan: Teknik Informatika, Alamat: Jl. Perintis Kemerdekaan, Email: alfanaast@gmail.com, No. Hp: 082144159056. A message dialog box titled 'Message' with an information icon displays the text 'Proses Simpan Data Berhasil' and an 'OK' button. Below the form are buttons: 'Tambah Data', 'Simpan', 'Edit', 'Hapus', 'Batal', and 'Keluar'. At the bottom is a table with columns: No., Nama Mahasiswa, STB, Jurusan, Alamat, Email, and No. Hp.

The screenshot shows the 'Aplikasi Biodata Mahasiswa' window after saving. The form fields are empty. The 'Jurusan' dropdown menu is set to 'Teknik Informatika'. The buttons are the same as in the previous screenshot. The table at the bottom now contains one row of data:

No.	Nama Mahasiswa	STB	Jurusan	Alamat	Email	No. Hp
1	alfan	0015	Teknik Informatika	Jln. Perintis Kemerde...	alfanaast@gmail.c...	082144159056

## D. Codingan tombol “Edit”

### 1. Ini adalah source code dari menu “Edit”

```
private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String sql = "UPDATE biodata SET nama='"+txtNama.getText()+"', stb='"+txtStb.getText()+"', jurusan='"+cmbJurusan.getSelectedItem()+  
            "', alamat='"+txtAlamat.getText()+"', email='"+txtEmail.getText()+"', no_hp='"+txtNoHp.getText()+"' WHERE nama='"+txtNama.getText()+"'";  
        java.sql.Connection conn = (Connection)Koneksi.ConfigDB();  
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
        pstmt.executeUpdate();  
        JOptionPane.showMessageDialog(null, "Edit Data Berhasil...");  
    } catch (HeadlessException | SQLException e) {  
        JOptionPane.showMessageDialog(this, e.getMessage());  
    }  
    tampilkan_data();  
    kosongkan_form();  
}
```

### 2. Ini adalah tampilan ketika data mahasiswa telah berhasil diedit

#### a. Sebelum diedit

DATA MAHASISWA

Nama Mahasiswa: alfan

STB: 0015

Jurusan: Teknik Informatika

Alamat: Jln. Perintis Kemerdekaan

Email: alfhanaast@gmail.com

No. Hp: 082144159056

Tambah Data Simpan Edit Hapus Batal Keluar

No.	Nama Mahasiswa	STB	Jurusan	Alamat	Email	No. Hp
1	alfan	0015	Teknik Informatika	Jln. Perintis Kemerdekaan	alfhanaast@gmail.com	082144159056

#### b. Setelah diedit

DATA MAHASISWA

Nama Mahasiswa: alfan

STB: 13020160015

Jurusan: Teknik Informatika

Alamat: Jln. Perintis Kemerdekaan

Email: alfhanaast@gmail.com

No. Hp: 082144159056

Tambah Data Simpan Edit Hapus Batal Keluar

No.	Nama Mahasiswa	STB	Jurusan	Alamat	Email	No. Hp
1	alfan	0015	Teknik Informatika	Jln. Perintis Kemerdekaan	alfhanaast@gmail.com	082144159056



Aplikasi Biodata Mahasiswa

### DATA MAHASISWA

Nama Mahasiswa:

STB:

Jurusan:

Alamat:

Email:

No. Hp:

No.	Nama Mahasiswa	STB	Jurusan	Alamat	Email	No. Hp
1	alfan	13020160015	Teknik Informatika	Jln. Perintis Kemerdekaan	alfhanaast@gmail.com	082144159056

## E. Codigan tombol “Hapus”

### 1. Ini adalah source code dari menu “Hapus”

```
private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String sql = "DELETE FROM biodata WHERE nama='"+txtNama.getText()+"'";
        java.sql.Connection conn = (Connection)Koneksi.ConfigDB();
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(null, "Hapus Data Berhasil...");

    } catch (HeadlessException | SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    tampilkan_data();
    kosongkan_form();
}
}
```

### 2. Ini adalah tampilan ketika data mahasiswa telah berhasil di hapus

Aplikasi Biodata Mahasiswa

### DATA MAHASISWA

Nama Mahasiswa:

STB:

Jurusan:

Alamat:

Email:

No. Hp:

**Message**

Hapus Data Berhasil...

No.	Nama Mahasiswa	STB	Jurusan	Alamat	Email	No. Hp
1	alfan	13020160015	Teknik Informatika	Jln. Perintis Kemerdekaan	alfhanaast@gmail.com	082144159056

#### F. Codengan dari tabel yang memuat data mahasiswa

Codengan ini dimaksudkan agar setiap baris atau kolomnya dapat diklik sehingga dapat edit maupun dihapus datanya

```
private void tblMHSMouseClicked(java.awt.event.MouseEvent evt) {
    int baris = tblMHS.rowAtPoint(evt.getPoint());
    String nama = tblMHS.getValueAt(baris, 1).toString();
    txtNama.setText(nama);

    String stb = tblMHS.getValueAt(baris, 2).toString();
    txtStb.setText(stb);

    String jurusan = tblMHS.getValueAt(baris, 3).toString();
    cmbJurusan.setSelectedItem(jurusan);

    String alamat = tblMHS.getValueAt(baris, 4).toString();
    txtAlamat.setText(alamat);

    String email = tblMHS.getValueAt(baris, 5).toString();
    txtEmail.setText(email);

    String hp = tblMHS.getValueAt(baris, 6).toString();
    txtNoHp.setText(hp);
}
```

#### G. Codengan dari tombol “Tambah Data”

```
private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {
    kosongkan_form();
}
```

## H. Codengan dari tombol “Keluar”

```
private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

## I. Codengan fungsi main dari JFrom

```
339 public static void main(String args[]) {  
340     /* Set the Nimbus look and feel */  
341     Look and feel setting code (optional)  
362  
363     /* Create and display the form */  
364     java.awt.EventQueue.invokeLater(new Runnable() {  
365         public void run() {  
366             new Mahasiswa().setVisible(true);  
367         }  
368     });  
369 }  
370
```

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Sudah dapat terlihat bagaimana kode program dan implementasi dari aplikasi ini. Aplikasi ini memang dirancang dengan sangat sederhana untuk memudahkan user dalam penggunaannya. Aplikasi ini hanyalah sampel dan sebagai bukti bahwa, untuk serta penggunaan aplikasi Java GUI berbasis MVC dengan menggunakan database bukan sesuatu yang sulit bagi mahasiswa yang bergelut di bidang IT maupun institusi Pendidikan itu sendiri.

#### **3.2 Saran**

Penulis menyadari betapa banyak kekurangan dari laporan ini. Untuk itu penulis memohon maaf kepada para pembaca jika terdapat kekeliruan dalam penulisan laporan serta pembuatan aplikasi ini. Penulis mengharapkan saran maupun kritik yang bijak dan bersifat membangun demi pengembangan pengetahuan serta aplikasi ini.