**Laporan Tugas Proyek**

**Pemrograman Berorientasi Objek**

**School**



**Disusun Oleh :**

|  |  |
| --- | --- |
| 11322001 | Gerald Renhart Aditia Sitio |
| 11322006 | Anastasya Capritiani Marpaung |
| 11322016 | Horas Sidabalok |
| 11322062 | Monica Silaban |

**Institut Teknologi Del**

**Fakultas Vokasi**

**DIII- Teknologi Informasi**

**Laguboti**

**2023**

**Daftar Isi**

[**1.** **FUNCTIONAL DESCRIPTION** 3](#_Toc153619429)

[**2.** **DESIGN CONCEPTUAL DATA MODEL** 4](#_Toc153619430)

[**3.** **DESIGN PHYSICAL DATA MODEL** 5](#_Toc153619431)

[**4.** **DESIGN TABLE STRUCTURE** 6](#_Toc153619432)

[**4.1 Tabel Siswa** 6](#_Toc153619433)

[**4.2** **Tabel Guru** 7](#_Toc153619434)

[**4.3 TABEL KELAS** 8](#_Toc153619435)

[**4.4.** **TABEL STAF** 8](#_Toc153619436)

[**5.** **DESAIN CLASS DIAGRAM** 9](#_Toc153619437)

[**6.** **OOP IMPLEMENTATION** 10](#_Toc153619438)

[**7.** **IMPLEMENTASI KESELURUHAN** 13](#_Toc153619439)

[**8.** **LINK YOUTUBE** 13](#_Toc153619440)

# **FUNCTIONAL DESCRIPTION**

Sistem yang dibangun adalah sistem informasi sekolah. Sistem ini dibangun untuk membantu siswa dalam sistem sekolah, seperti mengambil kelas. Dalam sistem sekolah terdapat fungsi – fungsi antara lain:

1. Menambah kelas

Fungsi menambah kelas digunakan oleh *user* untuk menambah kelas yang ada pada sistem sekolah.

1. Melihat kelas

Fungsi melihat kelas digunakan oleh *admin* dan *user* untuk melihat kelas yang ada pada sistem sekolah.

1. Mengedit kelas

Fungsi ini digunakan oleh *user* untuk melakukan pengeditan kelas yang sudah ditambahkan sebelumnya.

1. Menghapus kelas

Fungsi ini digunakan oleh *user* untuk menghapus kelas yang mengalami kesalahan saat melakukan penginputan atau menambah kelas.

1. Menambah data atau informasi staf

Fungsi menambah staf digunakan oleh *admin* untuk menambah data – data atau informasi staf pada sistem sekolah.

1. Mengedit data atau informasi staf

Fungsi ini digunakan oleh *admin* untuk mengedit data atau informasi staf yang telah ditambahkan sebelumnya.

1. Melihat data atau informasi staf

Fungsi melihat data dan informasi staf digunakan oleh *admin* unttuk melihat data staf yang ada pada sistem sekolah.

1. Menghapus data atau informasi staf

Fungsi ini digunakan oleh *admin* untuk melakukan penghapusan data atau informasi staf yang mengalami kesalahan.

1. Menambah data atau informasi siswa

Fungsi ini digunakan oleh *admin* untuk melakukan data siswa pada sistem sekolah.

1. Mengedit data atau informasi siswa

Fungsi mengedit data atau informasi siswa digunakan oleh *admin* untuk melakukan pengeditan terhadap data atau informasi siswa yang telah ditambahkan sebelumnya.

1. Melihat data atau informasi siswa

Fungsi melihat data atau informasi siswa digunakan oleh *admin* dan *user* untuk melihat data atau informasi siwa pada sistem sekolah.

1. Menghapus data atau informasi siswa

Fungsi ini digunkan oleh *admin* untuk menghapus data siswa yang mengalami kesalahan saat melakukan input data pada sistem sekolah.

1. Menambah data atau informasi guru

Fungsi ini digunakan oleh menambah data atau informasi data – data guru pada sistem sekolah.

1. Melihat data atau informasi guru

Fungsi ini digunakan oleh *admin* dan *user* untuk melihat data atau informasi guru pada sistem sekolah.

1. Mengedit data atau informasi guru

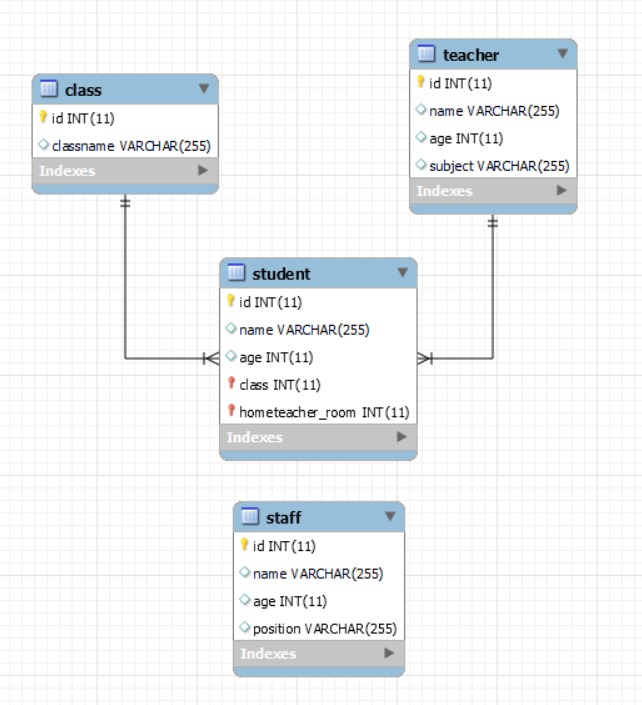
Fungsi ini digunakan oleh *admin* untuk melakukan pengeditan pada data atau informasi guru yang telah ditambahkan sebelumnya pada sistem sekolah.

1. Menghapus data atau informasi guru

Fungsi ini digunakan oleh *admin* untuk melakukan penghapusan data atau informasi guru pada sistem sekolah.

# **DESIGN CONCEPTUAL DATA MODEL**

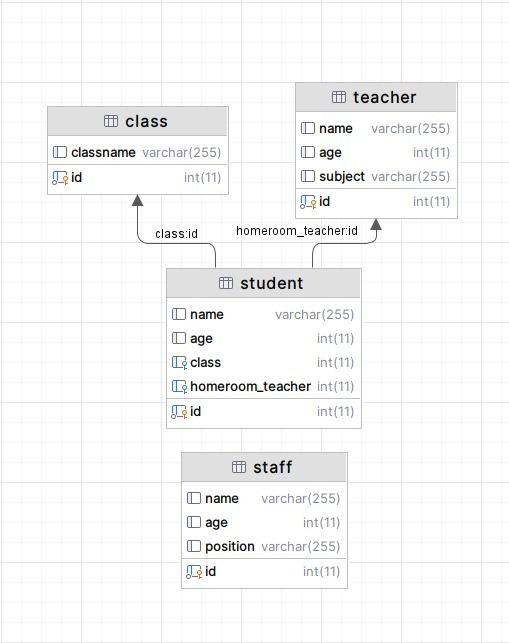
Berikut merupakan Conceptual Data Model dari sistem sekolah yang dapat dilihat pada gambar 1.



**Gambar 1 Conceptual Data Model Sistem Sekolah**

# **DESIGN PHYSICAL DATA MODEL**

Berikut merupakan Physical Data Model pada sistem sekolah yang dapat dilihat pada gambar 2.



**Gambar 2 Physical Data Model Sistem Sekolah**

# **DESIGN TABLE STRUCTURE**

## **4.1 Tabel Siswa**

Identifikasi/Nama : Siswa

Deskripsi Isi : Tabel ini berisi data tentang informasi siswa

Jenis : Tabel master

Volum : 5 *field*

Primary Key : ID

**Tabel 1 Tabel Siswa**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Deskripsi** | **Type & Lenght** | **Boleh NULL** | **Default** | **Keterangan** |
| Id | Berisi id siswa | INT(11) | NO | None | *Primary Key* |
| Nama | Berisi nama siswa | VARCHAR(255) | NO | None | *Attribute non key* |
| Umur | Berisi umur siswa | INT(11) | NO | None | *Attribute non key* |
| Kelas | Berisi kelas yang dimasuki oleh siswa | INT(11) | NO | None | *Foreign Key* |
| guru | Berisikan guru yang mengajar pada siswa | INT(11) | NO | none | *Foreign Key* |

## **4.2** **Tabel Guru**

Identifikasi/Nama : Guru

Deskripsi Isi : Tabel ini berisi data tentang informasi guru

Jenis : Tabel master

Volum : 4 *field*

Primary Key : ID

**Tabel 2 Tabel Guru**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Deskripsi** | **Type & Lenght** | **Boleh NULL** | **Default** | **Keterangan** |
| Id | Berisi id guru | INT(11) | NO | None | *Primary Key* |
| Nama | Berisi nama guru | VARCHAR(255) | NO | None | *Attribute non key* |
| Umur | Berisi umur guru | INT(11) | NO | None | *Attribute non key* |
| Mata pelajaran | Berisi mata pelajaran yang diajarkan | VARCHAR(255) | NO | None | *Attribute non key* |

## **4.3 TABEL KELAS**

Identifikasi/ Nama : Kelas

Deskripsi Isi : Tabel ini berisi tentang data kelas

Jenis : Tabel Master

Volum : 2 *field*

Primary Key : Id

**Tabel 3 Tabel Kelas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Deskripsi** | **Type & Lenght** | **Boleh NULL** | **Default** | **Keterangan** |
| Id | Berisi id kelas | INT(11) | NO | None | *Primary Key* |
| Nama | Berisi nama kelas | VARCHAR(255) | NO | None | *Attribute non key* |

## **TABEL STAF**

Identifikasi/Nama : Staf

Deskripsi Isi : Tabel ini berisikan informasi tentang staf

Jenis : Tabel Master

Volum : 4 *field*

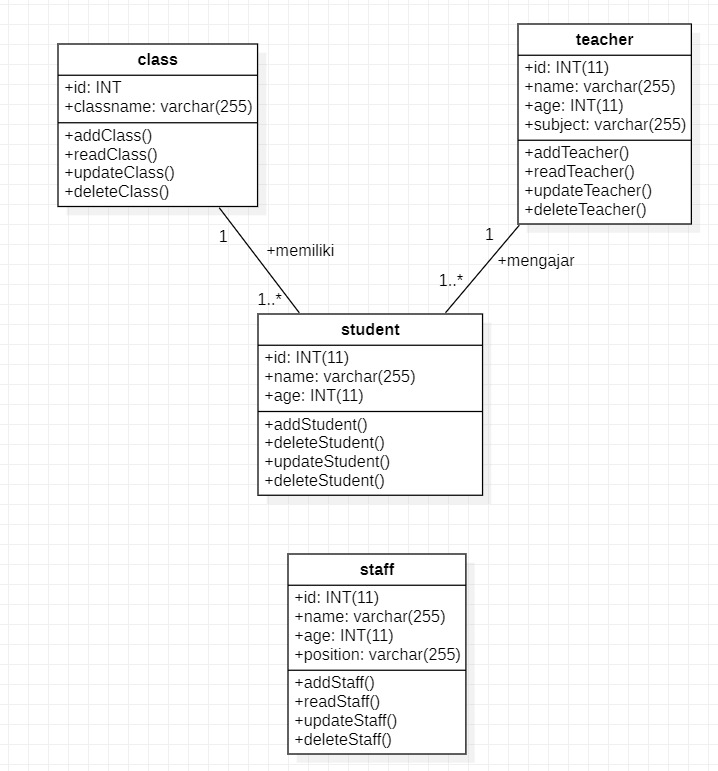
Primary Key : Id

**Tabel 4 Tabel Staf**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Deskripsi** | **Type & Lenght** | **Boleh NULL** | **Default** | **Keterangan** |
| Id | Berisi id staf | INT(11) | NO | None | *Primary Key* |
| Nama | Berisi nama staf | VARCHAR(255) | NO | None | *Attribute non key* |
| Umur | Berisi umur staf | INT(11) | NO | None | *Attribute non key* |
| Posisi atau jabatan | Berisi informasi mengenai posisi atau jabatan staf | VARCHAR(255) | NO | None | *Attribute non key* |

# **DESAIN CLASS DIAGRAM**

Berikut merupakan design class diagram pada sistem sekolah yang dapat dilihat pada gambar 3.



**Gambar 3 Class Diagram Sistem Sekolah**

# **OOP IMPLEMENTATION**

* 1. **Abstraction**

Salah satu contoh abstraksi ialah saat memulai window baru untuk menampilkan form/table pada program ini. Disini terdapat kode yang saya buat untuk membuat sebuah window baru untuk form memasukkan data kelas yang baru.

**public class ClassInsertInit {  
  
 Scene scene;  
 Stage stage;  
  
 public ClassInsertInit() throws IOException{  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("classinsert.fxml"));  
 scene = new Scene(loader.load());  
 stage = new Stage();  
  
 stage.setScene(scene);  
 }  
  
 public void show(){  
 stage.setTitle("add a class in your school");  
 stage.initModality(Modality.WINDOW\_MODAL);  
 stage.show();  
 }  
  
 public void Close(){  
 stage.close();  
 }  
}**

Detail untuk membuat halaman baru di Controller Manajemen Kelas itu tidak perlu dilihat, disini kita hanya perlu objek baru dan viola… Window baru

@FXML  
public void StartAddClassWindow() throws IOException {  
 ClassInsertInit cii = new ClassInsertInit();  
 cii.show();  
}

* 1. **Encapsulation**

Salah satu contohnya ialah ada di kelas Home. Label untuk welcomeText ini hanya terbatas atau dapat diakses dari dalam class Home ini. Serta di fungsi-fungs apapun, variable local-lokal ini bisa jadi contoh karena mereka itu hanya bisa dipanggil didalam fungsi masing-masing

**Contoh class Home**

public class Home {  
 @FXML  
 private Label welcomeText;

...

}

**Contoh pada fungsi/method**

@FXML  
public void opentmpage(MouseEvent mouseEvent) throws IOException {  
 TeacherManagementInit tmi = new TeacherManagementInit();  
 tmi.show();  
}

* 1. **Inheritence**

Salah satu penggunaan Inheritence disini ialah ada di kelas Main yang menjadi entrypoint dari aplikasi. Disini kita menggunakan class parent Application. Selain itu kita menggunakan fungsi dari parent yakni launch() dan meng-override dari method start()

public class Main extends Application {  
 @Override  
 public void start(Stage stage) throws IOException {  
 FXMLLoader fxmlLoader = new FXMLLoader(Main.class.getResource("hello-view.fxml"));  
 Scene scene = new Scene(fxmlLoader.load(), 740, 400);  
  
 stage.setTitle("Aplikasi ");  
 stage.setScene(scene);  
 stage.setResizable(false);  
 stage.show();  
 }

public static void main(String[] args) {  
 launch();  
 }  
}

* 1. **Polymorphism**

Salah satu contoh polymorphism disini ialah pada data class people yang merupakan data class yang abstract. Class People memiliki method getAge() dan getId() yang di-implement ulang.

**Exibit a**

public class Staff extends People{  
  
 private String position;  
 private String name;  
  
...

@Override  
 public String getName() {  
 return this.name;  
 }

}

**Exibit b**

public abstract class People {

...  
 protected String name;  
...  
 public abstract String getName();  
...  
}

**Exibit c**

public class Student extends People {  
...  
 public String getName() {  
 return super.name;  
 }  
}

* 1. **Interface**

Salah satu contoh implementasinya ialah pada Interface Data yang menerapkan beberapa method yang harus di implementasikan, disini 3 method dari Interface Data yang berupa Insert(), Update() dan Drop().

public interface Data {  
 void Insert(MouseEvent event) throws SQLException;  
 void Update(MouseEvent event) throws SQLException;  
 void Drop(MouseEvent event) throws SQLException;  
}

* 1. **Exception Handling**

Salah satu contoh dari exception handling disini ialah penggunaan try…catch yang mencoba beberapa kode dan menangkap apabila ada terjadi error/kesalahan dalam program.

if (StudentTable != null) {  
 try {  
 ResultSet rs = (new DB()).Connect().createStatement().executeQuery(  
 "SELECT s.id as 'id', s.name as 'name', s.age as 'age', c.classname as 'classname', t.name as 'teacher\_name' FROM student s INNER JOIN class c ON s.class = c.id INNER JOIN teacher t ON s.homeroom\_teacher = t.id"  
 );  
 while (rs.next()) {  
 StudentTable.getItems().addAll(  
 new Student(  
 rs.getInt("id"),  
 rs.getString("name"),  
 rs.getString("classname"),  
 rs.getString("teacher\_name"),  
 rs.getInt("age")  
 )  
 );  
 }  
 } catch (SQLException sqe) {  
 System.err.println(sqe);  
 }

Selain itu juga terdapat penggunaan throws pada method untuk menjelaskan jenis Exception yang dikeluarkan oleh body (baik secara explicit maupun implicit).

@FXML  
public void opentmpage(MouseEvent mouseEvent) throws IOException {

...  
}

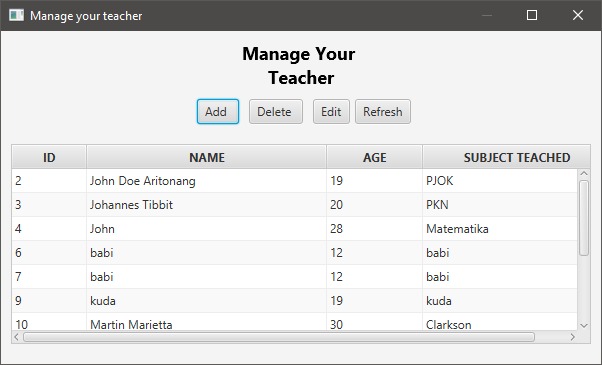
# **IMPLEMENTASI KESELURUHAN**

Implementasi dari aplikasi ini menggunakan sistem vie-controller yang mengikuti bawaan dari javaFX. Didalam aplikasi ini terdapat beberapa view-view seperti view untuk daftar data, view untuk menambahkan data, mengubah data dsb. Didalam implementasi kami setiap view terdapat dua komponen, yakni sebuah file FXML yang menjadi tempat desain layout UI dari view didefinisikan dan inisiator yang memulai window untuk view tersebut.

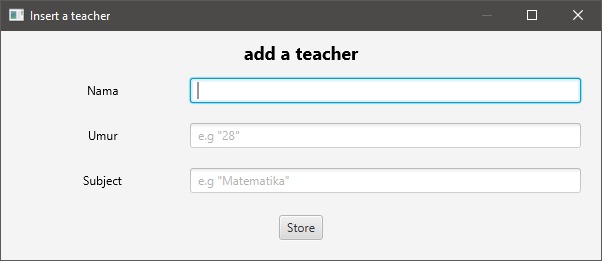
Controller dalam aplikasi ini akan menyediakan fungsi-fungsi agar komponen-komponen dalam aplikasi ini dapat bekerja, selain itu controller-controller ini juga bertugas untuk berkomunikasi dengan database (dengan menggunakan library JDBC) untuk menyimpan dan mengambil data dari database.



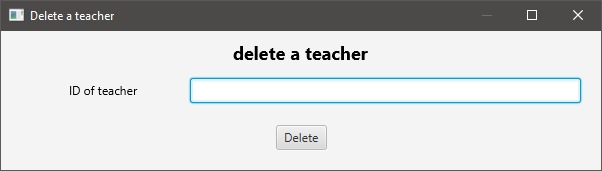
**Gambar 4 Implementasi Sistem Sekolah**



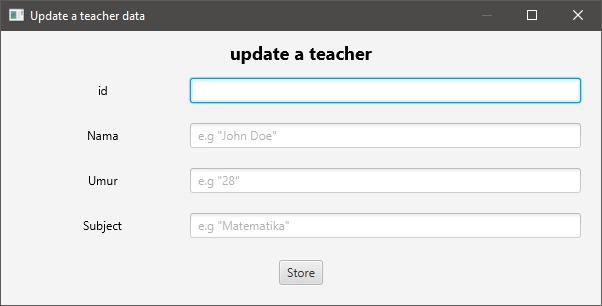
**Gambar 5 Implementasi Mengelola Data Guru**



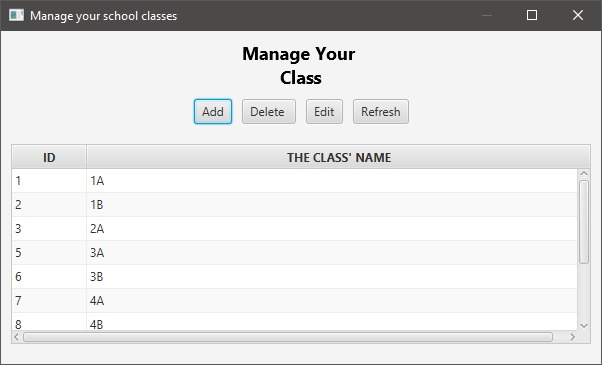
**Gambar 6 Implementasi Menambah Data Guru**



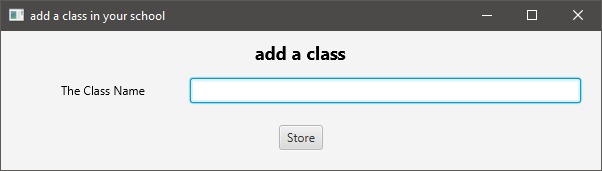
**Gambar 7 Implementasi Menghapus Data Guru**



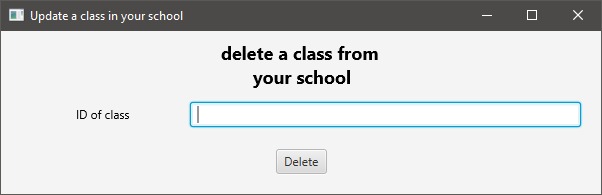
**Gambar 8 Implementasi Mengedit Data Guru**



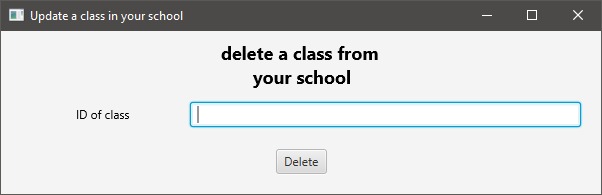
**Gambar 9 Implementasi Mengelola Data Kelas**



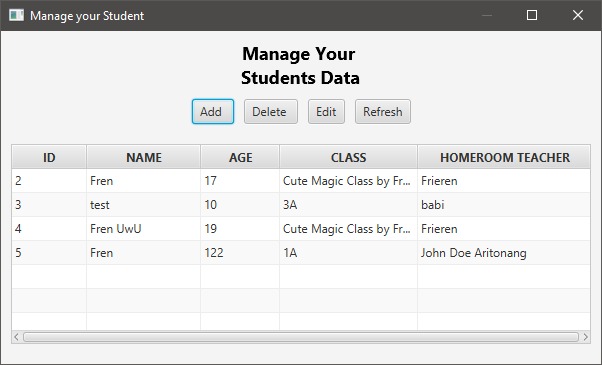
**Gambar 10 Implementasi Menambah Kelas**



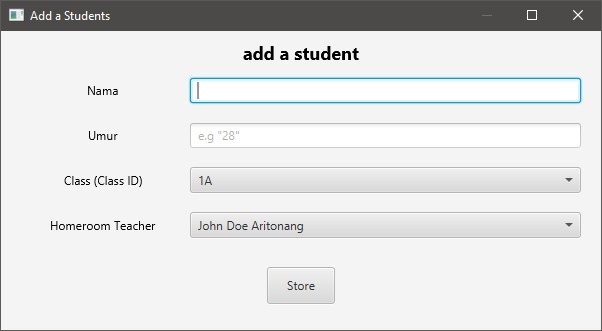
**Gambar 11 Implementasi Menghapus Kelas**



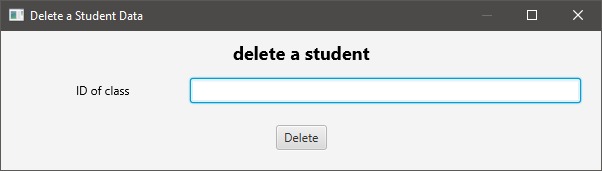
**Gambar 12 Implementasi Mengedit Kelas**



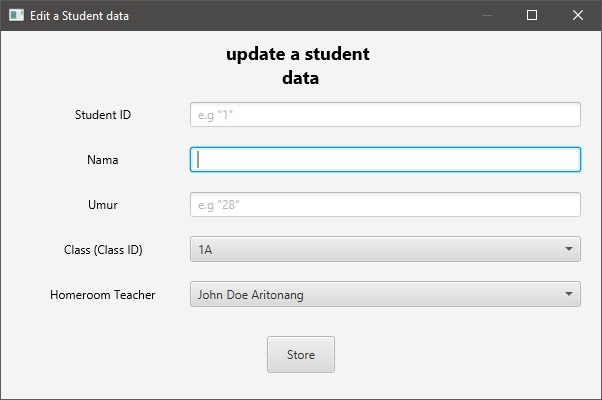
**Gambar 13 Implementasi Mengelola Data Siswa**



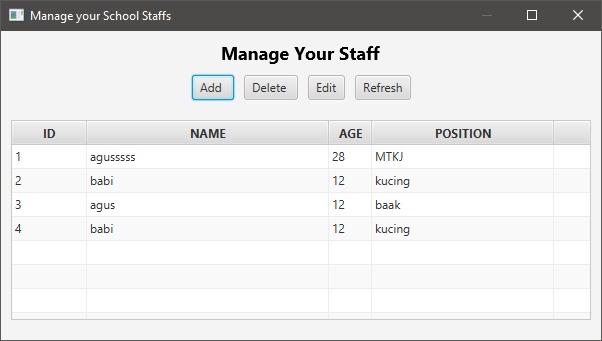
**Gambar 14 Implementasi Menambah Data Siswa**



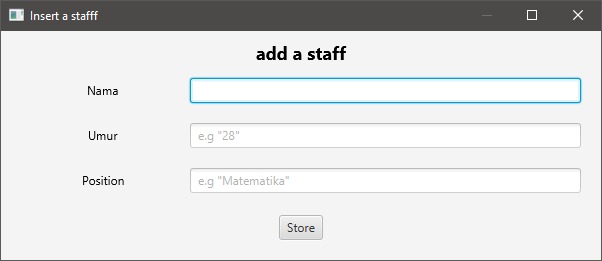
**Gambar 15 Implementasi Menghapus Data Siswa**



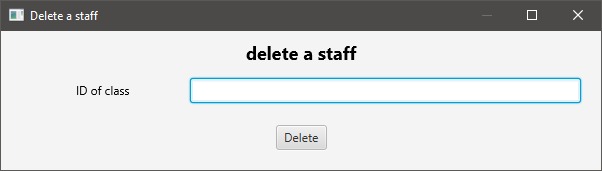
**Gambar 16 Implementasi Mengedit Data Siswa**



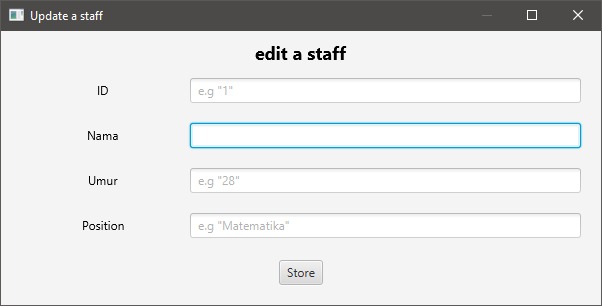
**Gambar 17 Implementasi Mengelola Data Staf**



**Gambar 18 Implementasi Data Staf**



**Gambar 19 Implementasi Menghapus Data Staf**



**Gambar 20 Implementasi Mengedit Data Staf**

# **LINK YOUTUBE**

[***https://youtu.be/v-CwxS7\_X1c***](https://youtu.be/v-CwxS7_X1c)