

LAPORAN
UJIAN AKHIR SEMESTER
PEMOGRAMAN ORIENTASI OBJEK
“Crud Application With JavaFX”

Dosen Pengajar : Ir.Gede Humaswara Prathama,S.T.,M.T.



Disusun Oleh :

NAMA : Dede Juniawan
NIM : (42030066)

JURUSAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA UNIVERSITAS
PENDIDIKAN NASIONAL (UNDIKNAS)
2021/2022

CRUD APPLICATION WITH JAVAFX

Saya telah membuat sebuah project aplikasi berbasis objek menggunakan bahasa pemrograman java yang dapat membuat(Create), membaca(Read), mengubah(Update/Edit), dan menghapus>Delete) data atau CRUD dengan GUI nya menggunakan tools javafx. Javafx adalah sebuah platform software untuk membangun sebuah aplikasi rich internet application (RIA) yang bisa berjalan pada berbagai macam perangkat. Seperti komputer dekstop, web browser di Windows, Linux dan Mac OSX. Sebelumnya saya telah membuat database di aplikasi MySQLyog dengan nama database **buku**, Dengan nama-nama field dan data sebagai berikut dapat di lihat digambar berikut:

<input type="checkbox"/>	Idbuku	Judul	Jenis
<input type="checkbox"/>	1	Programmer ?	Non Fiksi
<input type="checkbox"/>	2	Belajar C++	Non Fiksi
<input type="checkbox"/>	3	Keep On going	fiksi
<input type="checkbox"/>	4	Apa itu bahasa java	Non Fiksi
<input type="checkbox"/>	6	Always Be Oke	Fiksi
<input type="checkbox"/>	8	You can be	Fiksi

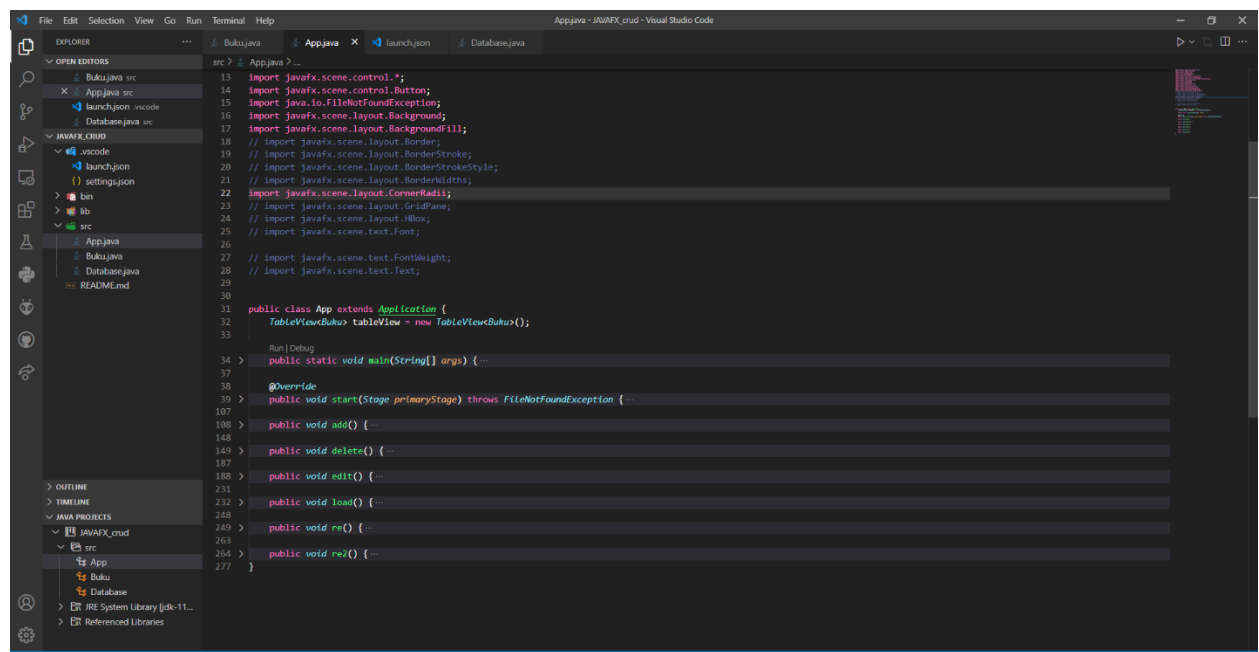
Setelah itu saya mendownload library javafx di [Getting Started with JavaFX \(openjfx.io\)](https://openjfx.io)

Dan mendownload driver mysql connector di [MySQL :: Download Connector/J](#)

Selanjutnya saya memakai Text Editor IDE Visual Studio Code,dengan langkah-langkah berikut ini:

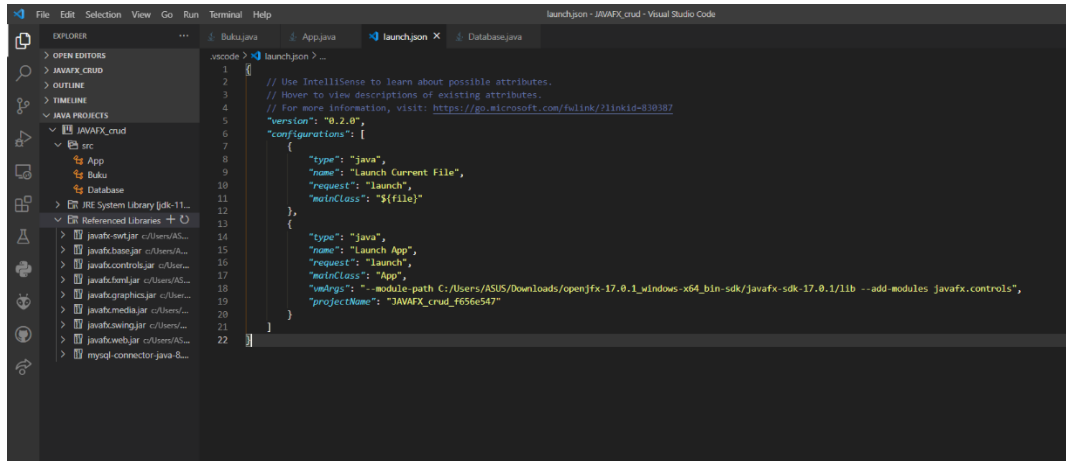
1. Membuat project java menggunakan IDE Visual Studio Code

Seperti berikut ini :

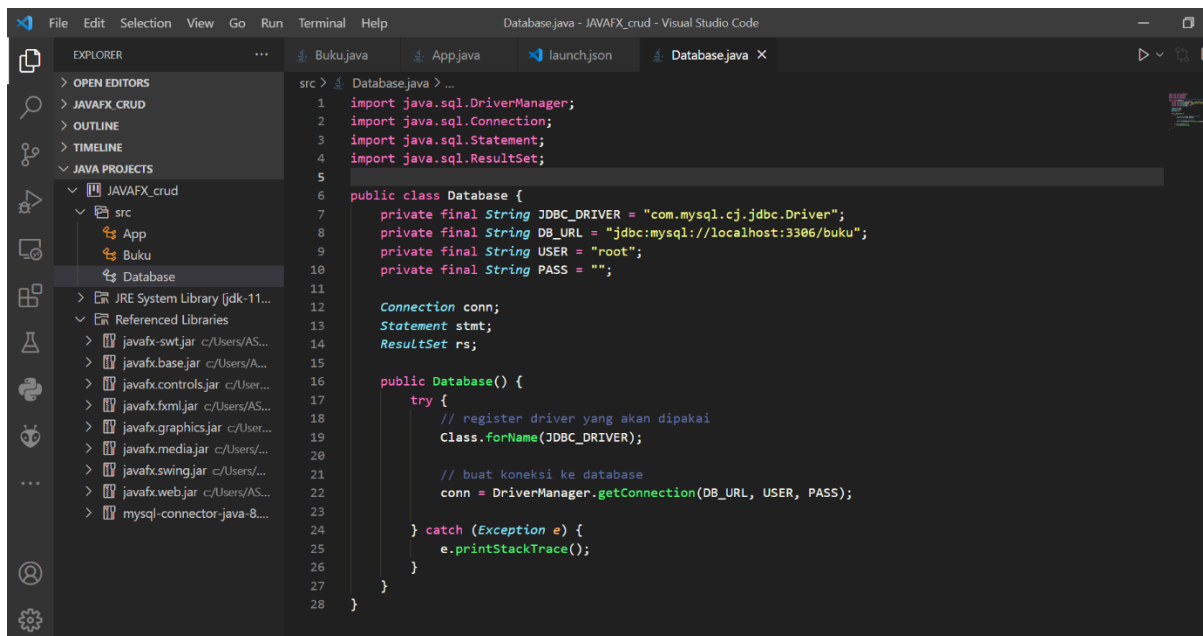


Saya telah membuat project dengan tiga Class,yaitu App.java , Buku.java dan Database.java

Lalu menambahkan library javafx untuk menampilkan Graphic User Interface (GUI) dan mysql connector untuk menghubungkan java dengan database yang saya buat
Sekaligus membuat konfigurasi untuk menempatkan directory file library javafx yang sudah didownload sebelumnya

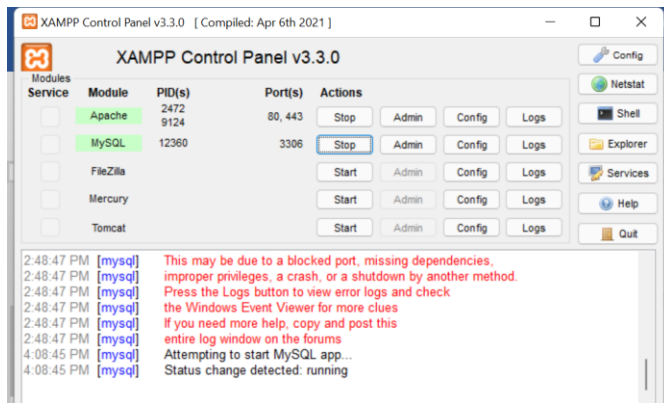


2.Membuat Koneksi Database MySQL

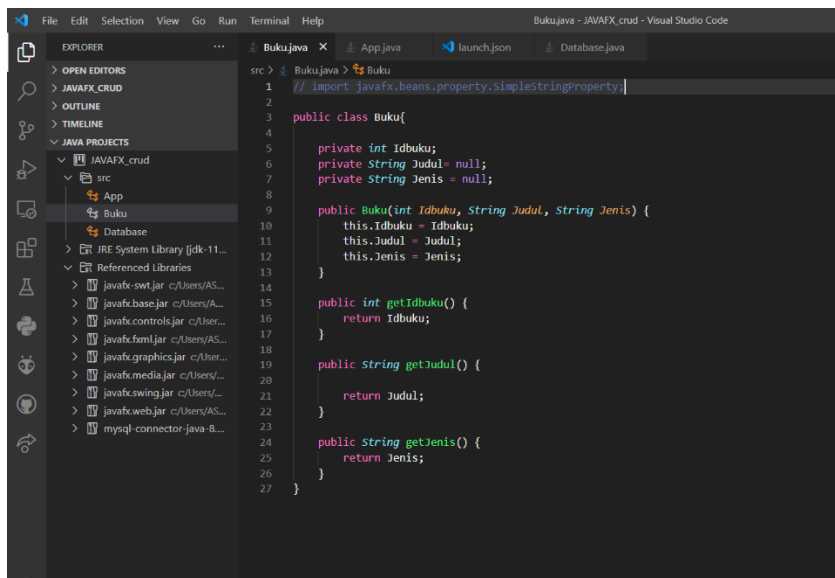


lalu memasukan/ import beberapa library yang dibutuhkan untuk membuat program seperti gambar di atas
saya membuat sebuah class yang berfungsi untuk membuat koneksi ke MySQL, saya deklarasi interface Connection dengan akses modifier private final agar tidak dapat diubah untuk mendapatkan koneksi. Di dalam method tersebut terdapat pengecekan untuk memastikan bahwa koneksi hanya dilakukan sekali(Try)ketika aplikasi berjalan, ini biasa disebut sebagai **singleton**.
gambar diatas yang dibuatkan variable adalah url, user, dan pass. Dan diakhir program saya kembalikan atau return value / nilai pada variable connect

pastikan localhost portnya sesuai dengan versi Xamp yang digunakan karena akan berpengaruh dengan penyimpanan database
disini saya menggunakan port 3306 sesuai dengan syntax diatas,atau bisa dilihat dibawah ini:



3.Membuat Class untuk modifikasi Database



Lalu saya membuat class sebuah tabel yang dimanfaatkan untuk menampung informasi database Buku saya, sehingga saya representasikan ke dalam sebuah class yaitu Buku.java. Jika di Java sering disebut dengan entitas atau juga POJO, dimana di dalam class tersebut terdapat getter dan setter. karena saya tidak mengizinkan bebas data untuk mengubah dan mengambil nilai yang terdapat pada class Bukue ini. Konsep ini biasa disebut dengan **Engkapsulasi**

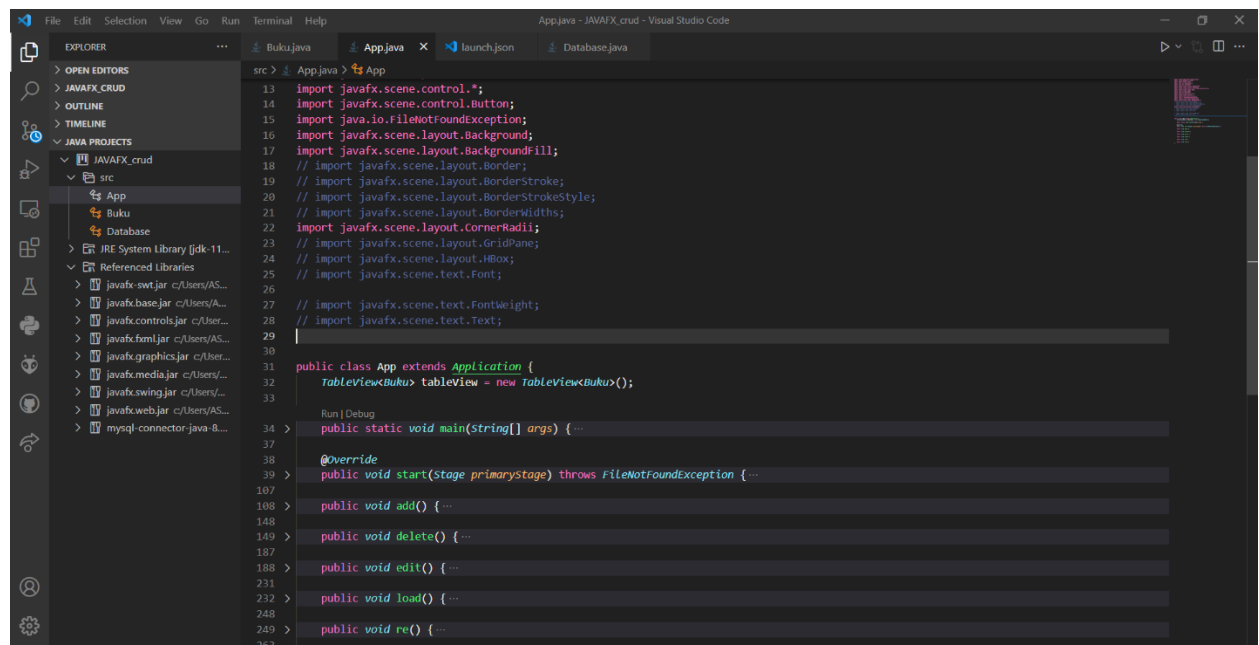
Disini saya menambahkan property dengan type data string,dan integer dan Private , karena disini saya tidak akan mengubah-ubah lagi propertynya. Property yang saya tambahkan yakni Idbuku, Judul,Jenis, dan status Lalu saya membuat sebuah constructor yang berfungsi untuk membuat proses awal dalam persiapan object, seperti memberi nilai kepada property, memanggil method internal serta beberapa proses lain yang dirasa perlu. Saya

menambahkan parameter sesuai dengan property yang sudah saya buat. Selanjutnya saya menambahkan method getter dan setternya untuk setiap property yang telah dibuat. Pada dasarnya metode **get** digunakan untuk mengembalikan nilai pada variable dengan menggunakan fungsi **return**

4.Membuat Class untuk Run Class

Sebelum membuat function dan sintak nya,terlebih dahulu harus mengimport atau memasukan library dari referenced Libraries ke Class yang kita gunakan untuk menjalankan Database dan JavaFx

Seperti gambar dibawah ini:



Dapat terlihat pada gambar diatas bahwa ada beberapa library yang saya import untuk memasukan suatu method atau perintah dalam bahasa pemrograman Java sehingga perintah tersebut dapat aktif dan digunakan atau berfungsi. Semua import diatas saya gunakan untuk mendukung program yang saya buat selanjutnya

Sebagai catatan kita bisa mengimport semua library dengan syntak `Import .sql.java.;`
Jika kita membutuhkan semua library sql dan javafxnya.*

Membuat Class yang digunakan untuk eksekusi

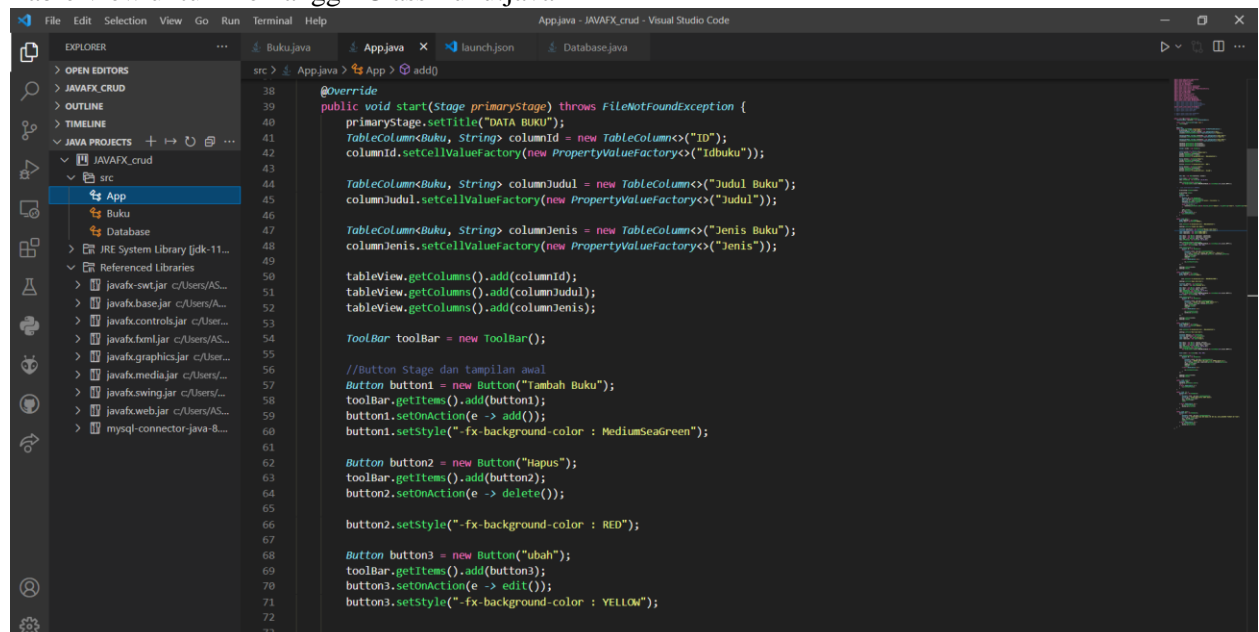

```

107
108     public void add() {
109         Stage addStage = new Stage();
110         Button save = new Button("SIMPAN");
111
112         save.setStyle("-fx-background-color : MediumSeaGreen");
113
114         addStage.setTitle("Tambah Data BUKU");
115
116         TextField JudulField = new TextField("Masukan Judul Buku");
117         TextField JenisField = new TextField("Masukan Jenis Buku");
118         Label labelJudul = new Label("JUDUL BUKU");
119         Label labelJenis = new Label("JENIS BUKU");
120
121         VBox hbox1 = new VBox(5, labelJudul, JudulField);
122         VBox hbox2 = new VBox(5, labelJenis, JenisField);
123         VBox vbox = new VBox(20, hbox1, hbox2, save);
124         vbox.setPadding(new Insets(10, 10, 10, 10));
125
126         vbox.setBackground(new Background(
127             new BackgroundFill(Color.MEDIUMSLATEBLUE, new CornerRadii(10), Insets.EMPTY)));
128         Scene scene = new Scene(vbox, 300, 400);
129
130         save.setOnAction(e -> {
131             Database db = new Database();
132             try {
133                 Statement state = db.conn.createStatement();
134                 String sql = "insert into daftar SET Judul='%s', Jenis='%s'";
135                 sql = String.format(sql, JudulField.getText(), JenisField.getText());
136                 state.execute(sql);
137                 addStage.close();
138                 load();
139             } catch (SQLException e1) {
140
141                 e1.printStackTrace();
142             }
143         });

```

Dengan menambahkan `setStyle` untuk mengatur warna button save, lalu dengan `Stage` untuk menampilkan database dan `VertikalBox(Vbox)` untuk menampilkan stage dan scene. Dan saya juga menambahkan `setPadding` untuk mengatur Layout fieldnya

Untuk Mengatur Table view Class Model database yang kita gunakan, disini saya menggunakan Table view untuk memanggil Class `Buku.java`



Dapat dilihat gambar diatas table viewnya dan Button yang sudah saya buat syntaknya

6. Menampilkan GUI javaFx

Saya akan mencoba menampilkan GU Interface javafx yang telah saya buat sebelumnya

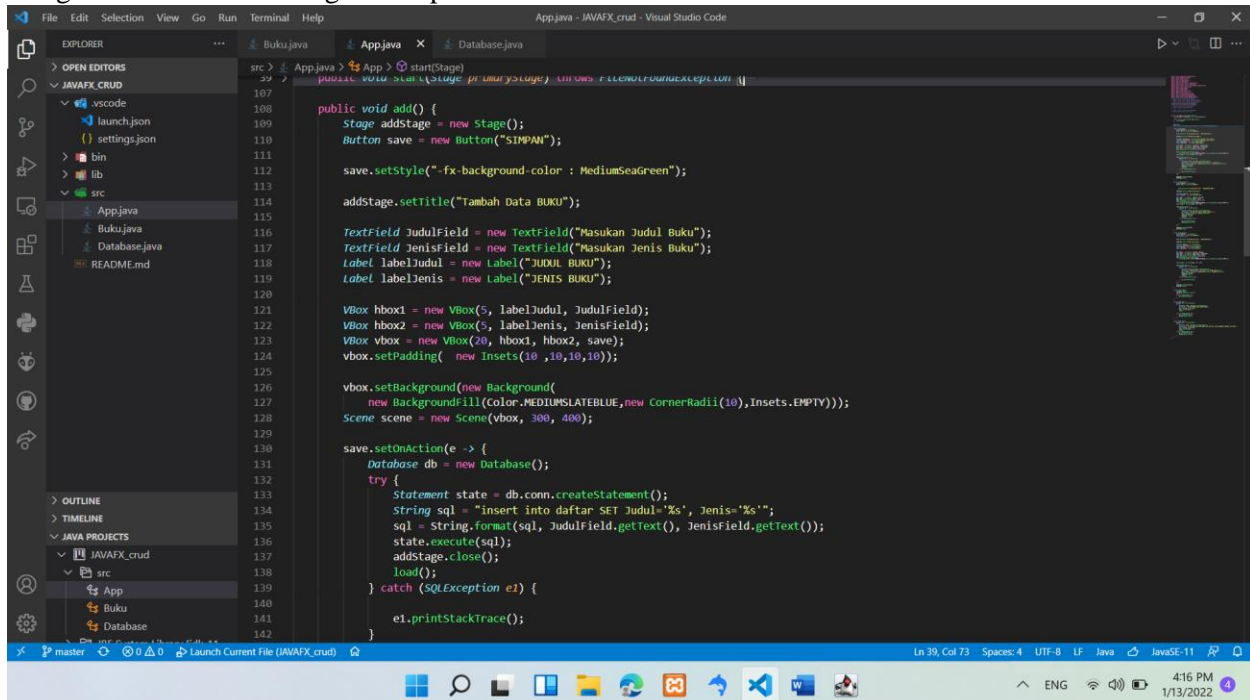
[illegible]

Maka akan menampilkan GUI dan stagenya dari JavaFX nya seperti berikut:

[illegible]

a. Create

Dengan menambahkan function seperti berikut:



```
src > App.java > App > start(Stage)
29 / **
107
108
109 public void add() {
110     Stage addStage = new Stage();
111     Button save = new Button("SIMPAN");
112     save.setStyle("-fx-background-color : MediumSeaGreen");
113
114     addStage.setTitle("Tambah Data BUKU");
115
116     TextField JudulField = new TextField("Masukan Judul Buku");
117     TextField JenisField = new TextField("Masukan Jenis Buku");
118     Label labelJudul = new Label("JUDUL BUKU");
119     Label labelJenis = new Label("JENIS BUKU");
120
121     VBox hbox1 = new VBox(5, labelJudul, JudulField);
122     VBox hbox2 = new VBox(5, labelJenis, JenisField);
123     VBox vbox = new VBox(20, hbox1, hbox2, save);
124     vbox.setPadding( new Insets(10,10,10,10));
125
126     vbox.setBackground(new Background(
127         new BackgroundFill(Color.MEDIUMSLATEBLUE,new CornerRadii(10),Insets.EMPTY)));
128     Scene scene = new Scene(vbox, 300, 400);
129
130     save.setOnAction(e -> {
131         Database db = new Database();
132         try {
133             Statement state = db.conn.createStatement();
134             String sql = "insert into daftar SET Judul='%s', Jenis='%s'";
135             sql = String.format(sql, JudulField.getText(), JenisField.getText());
136             state.executeUpdate(sql);
137             state.close();
138             addStage.close();
139             load();
140         } catch (SQLException e1) {
141             e1.printStackTrace();
142         }
143     });
144 }
```

Saya juga menambahkan setStyle,SetColor dan SetPadding nya

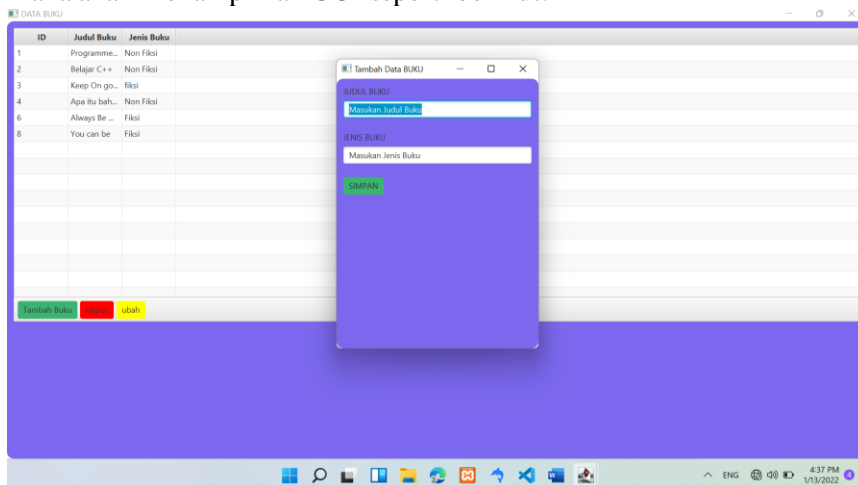
```
vbox.setPadding( new Insets(10,10,10,10));

vbox.setBackground(new Background(
    new BackgroundFill(Color.MEDIUMSLATEBLUE,new
    CornerRadii(10),Insets.EMPTY)));
```

Dengan button add untuk Create

```
Button button1 =new Button("Tambah Buku");
toolBar.getItems().add(button1);
button1.setOnAction(e -> add());
button1.setStyle("-fx-background-color : MediumSeaGreen");
```

maka akan menampilkan GUI seperti berikut:



b.READ(Select)

dengan syntax:

```
ResultSet rs = stmt.executeQuery("select * from daftar");
tableView.getItems().clear();
// tampilkan hasil query
while (rs.next()) {
    tableView.getItems().add(new Buku(rs.getInt("Idbuku"), rs.getString("Judul"),
rs.getString("Jenis")));
}
```

Untuk variabelnya harus menggunakan syntax Query karena sudah terkoneksi dengan tools database yang digunakan.

Maka akan menampilkan Field table dan data yang masukan ke tabel database yang digunakan

c.UPDATE(Edit)

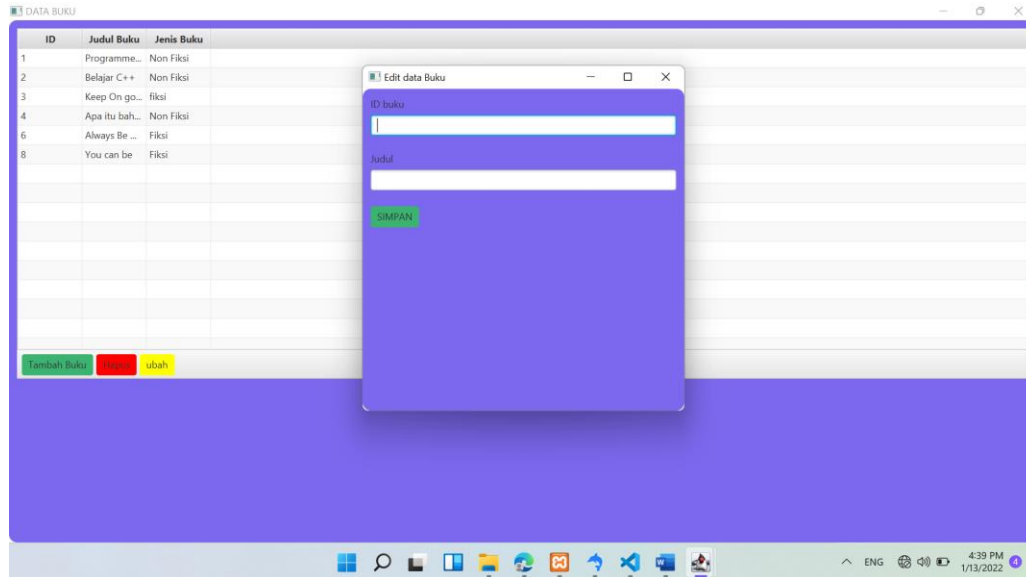
Membuat syntax seperti berikut:

```
save.setOnAction(e -> {
    Database db = new Database();
    try {
        Statement state = db.conn.createStatement();
        String sql = "UPDATE daftar SET Judul='%s' WHERE Idbuku='%s'";
        sql = String.format(sql, JudulField.getText(), IdField.getText());
        state.execute(sql);
        addStage.close();
        load();
    } catch (SQLException e1) {

        e1.printStackTrace();
    }
});

addStage.setScene(scene);
addStage.show();
}
```

Maka akan menampilkan saat dirun seperti berikut:



d.Delete

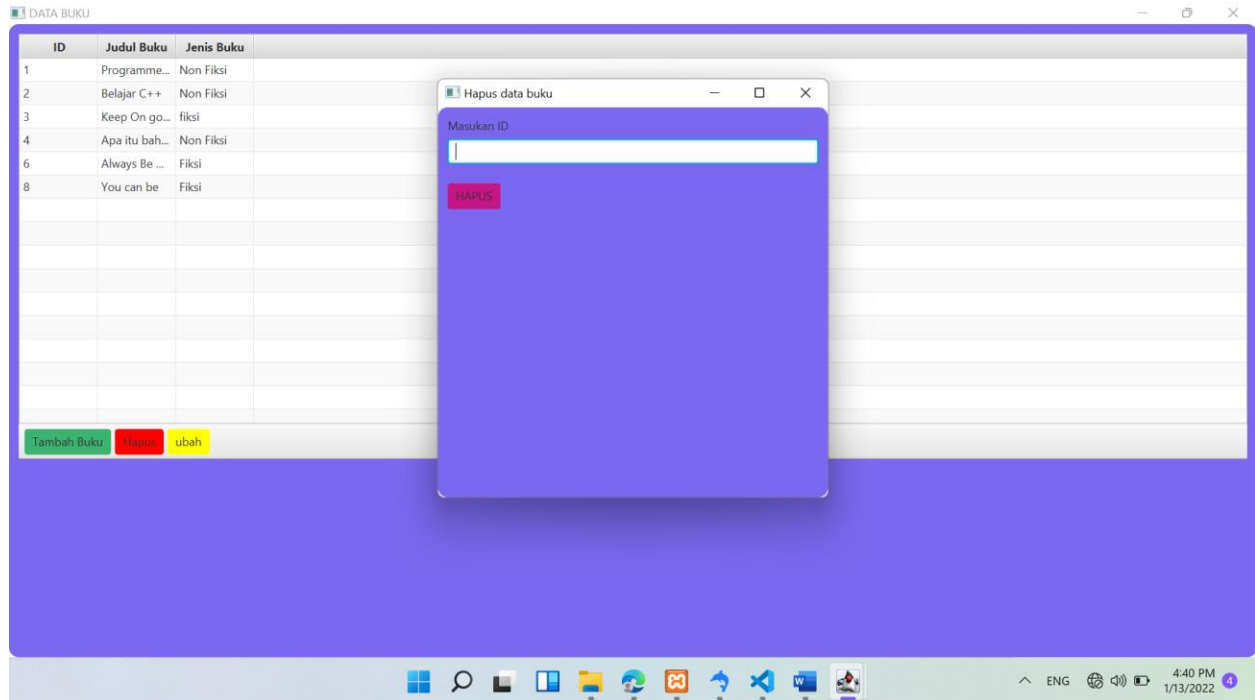
dengan syntax:

```
save.setOnAction(e -> {
    Database db = new Database();
    try {
        Statement state = db.conn.createStatement();
        String sql = "delete from daftar WHERE Idbuku='%s'";
        sql = String.format(sql, noField.getText());
        state.execute(sql);
        addStage.close();
        load();
    }
});
```

Untuk tambahan wajib menghidupkan xampp dan tools Database yang digunakan

Agar Button CRUD yang dipakai dapat digunakan

Maka akan menampilkan seperti berikut:



Setelah berhasil membuat Button CRUD,akhirnya kita berhasil menghubungkan java ke Mysql ,menampilkan Table database dan menerapkan CRUD.

Untuk mengenai codingan yang saya gunakan silahkan klik link github saya di bawah ini:
https://github.com/Djuni06/JAVAFX_crud.git