



# Tutorial JavaFX

---

By Seseorang

## 1. Pengenalan Library JavaFX

### Apa itu JavaFX?

JavaFX adalah library Java yang digunakan untuk membangun Rich Internet Applications. Aplikasi yang dikembangkan menggunakan JavaFX dapat berjalan di berbagai perangkat seperti Komputer Desktop, Mobile Phones, TV, Tablet, dll

Untuk mengembangkan Aplikasi GUI menggunakan bahasa pemrograman Java, programmer mengandalkan perpustakaan seperti Advanced Windowing Toolkit dan Swings. Setelah munculnya JavaFX, programmer java sekarang dapat mengembangkan aplikasi GUI dengan konten yang banyak. Dengan menggunakan teknologi Java, aplikasi ini memiliki tingkat penetrasi browser 76%. Untuk mengembangkan Aplikasi Client Side yang kaya dengan fitur, programmer menggunakan berbagai library untuk menambahkan fitur seperti Media, UI controls, Web, 2D dan 3D, dll. JavaFX mencakup semua fitur dalam satu library. Selain itu, para pengembang juga dapat mengakses fitur yang ada java library seperti Swings.

JavaFX menyediakan pengaturan grafis dan media API dan memanfaatkan Graphical Processing Unit modern melalui akselerasi perangkat keras grafis. JavaFX juga menyediakan interface yang dapat digunakan pengembang untuk menggabungkan animasi grafis dan UI controls. Penggunaan JavaFX dengan teknologi berbasis JVM seperti Java, Groovy dan JRuby. Jika pengembang memilih untuk JavaFX, tidak ada kebutuhan untuk belajar teknologi tambahan, seperti pengetahuan sebelumnya dari salah satu teknologi yang disebutkan di atas akan cukup baik untuk mengembangkan RIA menggunakan JavaFX.

### Fitur JavaFX

Berikut ini adalah beberapa fitur penting dari JavaFX :

- **Ditulis di Java** - The JavaFX perpustakaan ditulis di Java dan tersedia untuk bahasa yang dapat dijalankan pada JVM, yang meliputi - **Java, Groovy dan JRuby**. Ini aplikasi JavaFX juga platform independen.
- **FXML** - JavaFX memiliki bahasa yang dikenal sebagai FXML, yang merupakan HTML seperti bahasa markup deklaratif.

- **Scene Builder** - JavaFX menyediakan aplikasi bernama Scene Builder. Pada mengintegrasikan aplikasi ini di IDE seperti Eclipse dan NetBeans, pengguna dapat mengakses drag dan drop antarmuka desain, yang digunakan untuk mengembangkan aplikasi FXML (seperti swing Drag & Drop dan Aplikasi Dreamweaver).
- **Swing Interoperability** - Dalam aplikasi JavaFX, Kalian dapat menanamkan konten swing menggunakan kelas swing Node. Demikian pula, Kalian dapat memperbarui aplikasi Swing yang ada dengan fitur JavaFX seperti konten web tertanam dan media yang kaya grafis.
- **Built-in UI Controls** - JavaFX perpustakaan melayani kontrol UI menggunakan yang kita dapat mengembangkan aplikasi dengan fitur lengkap.
- **CSS Styling** - JavaFX menyediakan CSS seperti styling. Dengan menggunakan ini, Kalian dapat meningkatkan desain aplikasi Kalian dengan pengetahuan sederhana CSS.
- **Canvas dan Printing API** - JavaFX menyediakan kanvas, gaya modus langsung rendering API. Dalam package `javafx.scene.canvas` memegang satu set class untuk kanvas, menggunakan yang dapat kita tarik langsung dalam area scene JavaFX. JavaFX juga menyediakan calss untuk tujuan printing di package `javafx.print`.
- **Rich set of API's** - JavaFX perpustakaan menyediakan kaya set API untuk mengembangkan aplikasi GUI, 2D dan 3D grafis, dll. Set API ini juga mencakup kemampuan dari platform Java. Oleh karena itu, menggunakan API ini, Kalian dapat mengakses fitur bahasa Java seperti Generik, Anotasi, Multithreading, dan Ekspresi Lambda. Java collections library ditingkatkan dan konsep seperti Lists dan Map yang termasuk di dalamnya. Menggunakan ini, pengguna dapat mengamati perubahan dalam data model.
- **Integrated Graphics library** - JavaFX menyediakan class untuk grafis 2d dan 3d.
- **Graphics pipeline** - JavaFX mendukung grafis didasarkan pada pipa grafis Hardware-accelerated dikenal sebagai Prism. Ketika digunakan dengan Kartu Grafis yang didukung atau GPU ia menawarkan grafis yang halus. Dalam kasus sistem tidak mendukung kartu grafis maka default prisma untuk perangkat lunak render stack.

## Sejarah JavaFX

JavaFX pada awalnya dikembangkan oleh Chris Oliver, ketika dia bekerja untuk sebuah perusahaan bernama See Beyond Technology Corporation, yang kemudian diakuisisi oleh Sun Microsystems pada tahun 2005.

Poin-poin berikut memberikan informasi lebih lanjut tentang proyek ini

Awalnya proyek ini disebut sebagai F3 (Form Follows Functions) dan dikembangkan dengan maksud untuk menyediakan interface yang lebih kaya untuk mengembangkan Aplikasi GUI.

**Sun Microsystems** mengakuisisi perusahaan See luar pada Juni 2005, itu diadaptasi proyek F3 sebagai **JavaFX**.

Pada tahun 2007, JavaFX diumumkan secara resmi di Java One, sebuah dunia luas konferensi web yang diadakan setiap tahun.

Pada tahun 2008, **Netbeans** terintegrasi dengan JavaFX yang tersedia. Pada tahun yang sama, **Java Standar Development Kit** untuk JavaFX 1.0 dirilis.

Pada tahun 2009, Dirilis JavaFX 1.2 dan pada tahun yang sama Oracle Corporation mengakuisisi Sun Microsystems.

Pada tahun 2010, JavaFX 1.3 keluar dan pada tahun 2011 JavaFX 2.0 dirilis.

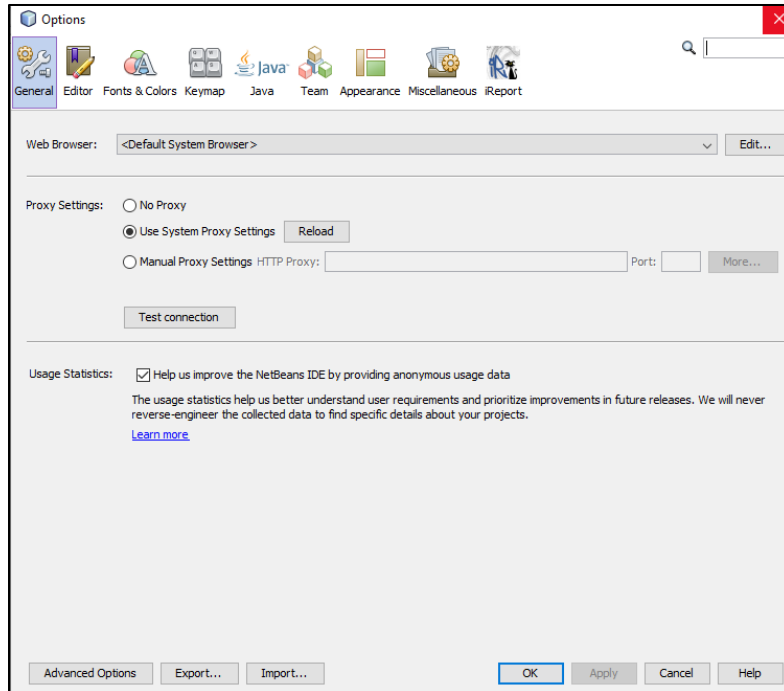
Versi terbaru, JavaFX8, dirilis sebagai bagian integral dari Java pada 18 Maret 2014.

## 2. Instalasi JavaFX

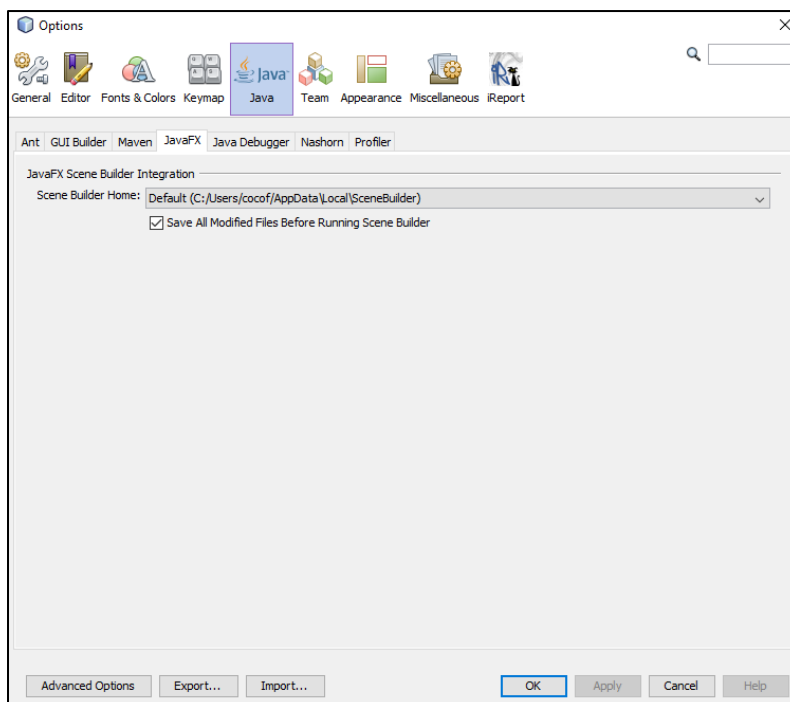
Untuk Instalasi JavaFX, bisa memakai Netbeans lalu untuk mendesain layout menggunakan software Scene Builder di link [Download](#), langkah-langkah mengunduh Scene Builder seperti berikut :

Download Scene Builder for Java 8		
Scene Builder 8.5.0 is for users who are still on Java 8. It was released on Jun 5, 2018.		
Product	Platform	Download
Scene Builder	Executable Jar	<a href="#">Download</a>
Scene Builder	Windows Installer 64-bit	<a href="#">Download</a>
Scene Builder	Windows Installer 32-bit	<a href="#">Download</a>
Scene Builder	Mac OS X dmg	<a href="#">Download</a>
Scene Builder	Linux Deb 64-bit	<a href="#">Download</a>
Scene Builder	Linux Deb 32-bit	<a href="#">Download</a>
Scene Builder	Linux RPM 64-bit	<a href="#">Download</a>
Scene Builder	Linux RPM 32-bit	<a href="#">Download</a>
Scene Builder Kit <a href="#">info</a>	Jar File	<a href="#">Download</a>
License: Scene Builder 8 is licensed under the BSD license.		

Setelah redirect ke website Gluon Scene Builder, download Scene Builder yang sama dengan Versi JDK, Gluon menyediakan 2 versi Scene Builder yaitu Scene Builder 17.0.0 yang kompatibel dengan JDK Java 11, dan Scene Builder 8.5.0 yang kompatibel dengan JDK Java 8, jadi sesuaikan dengan versi JDK Java masing-masing, jika belum mempunyai JDK bisa download disini untuk JDK 8, dan untuk JDK 11. Setelah itu buka Netbeans, pastikan kalau Scene Builder dengan Netbeans sudah terhubung. Pada Netbeans pilih Tools > Options maka akan muncul tampilan seperti berikut,

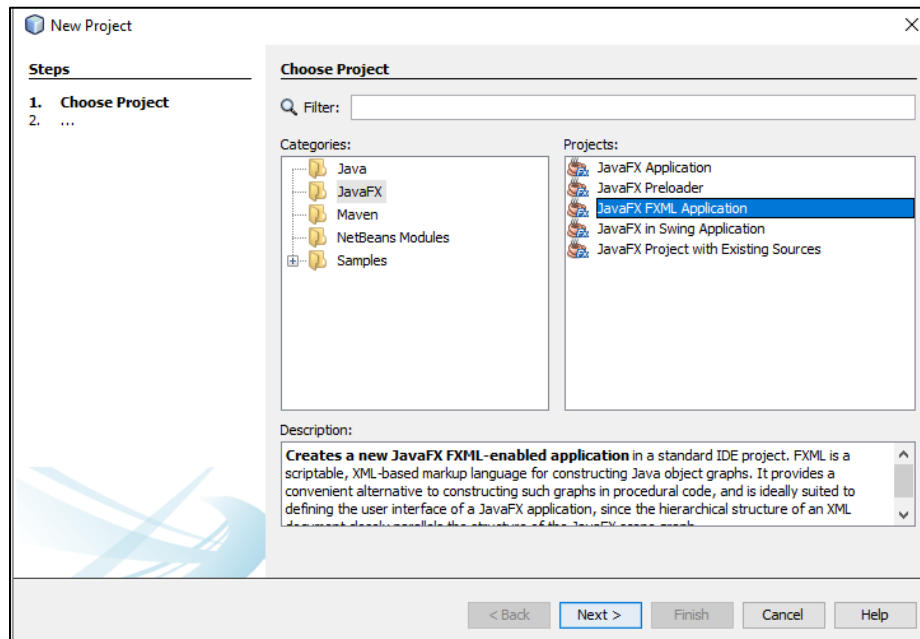


Gambar diatas merupakan tampilan Options Netbeans, lalu pada Options pilih opsi Java,

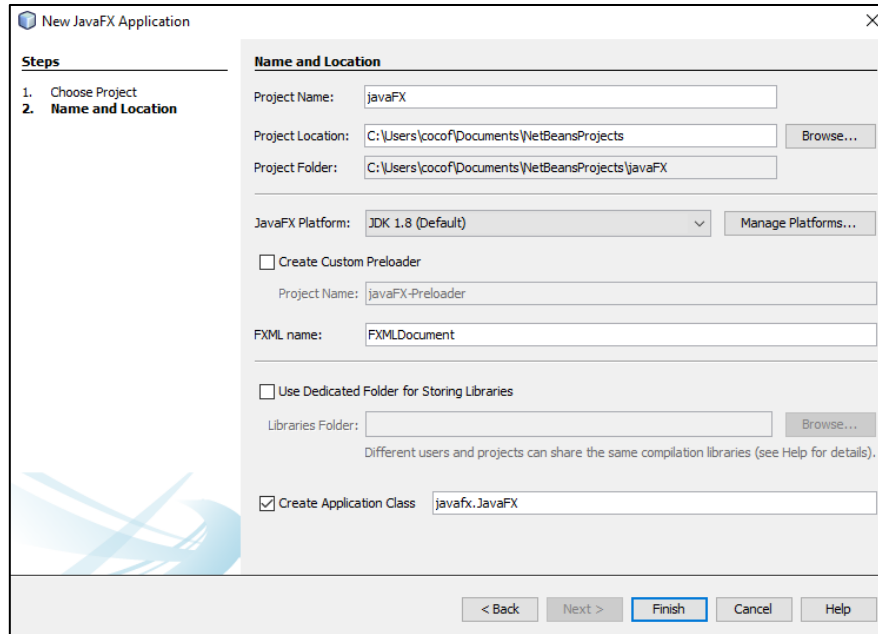


Pilih opsi Java lalu pilih JavaFX, pastikan kalau Scene Builder yang sudah terinstall tadi muncul, lalu centang opsi *Save All Modified Before running Scene Builder*, Ok. Maka Netbeans dan Scene Builder sudah terhubung.

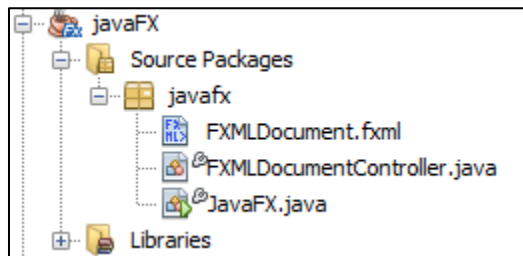
Setelah Netbeans dan Scene Builder sudah terhubung sekarang kita mulai membuat project baru berbentuk JavaFX, buat File > New Project >



Pilih JavaFX, pada pilihan Projects pilih JavaFX FXML Application, untuk pemilihan projects sebenarnya boleh antara JavaFX Application atau JavaFX FXML Application, bedanya kalau JavaFX FXML Application maka dari project akan langsung generate file controller dan file fxml untuk mengedit layout menggunakan Scene Buillder, sedangkan kalau JavaFX Application tidak men generate otomatis file FXML dan Controller, jadi untuk lebih baik memilih JavaFX FXML Application, setelah memilih projects, klik Next,



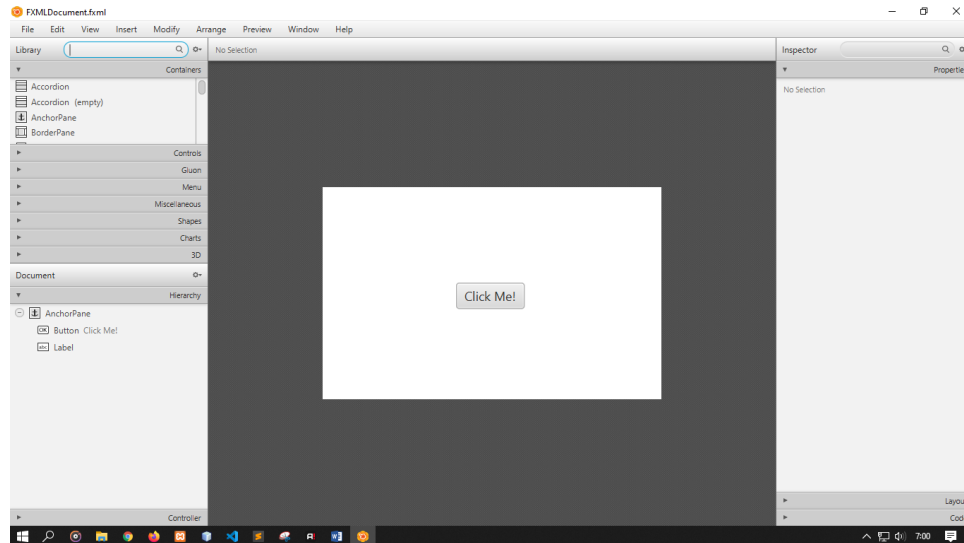
Sebelum mem-finish cek untuk JDK yang digunakan, memang hanya masalah kecil tetapi banyak yang mengabaikan dan mengalami masalah ketika membuat project dengan ketidak sesuaian JDK, setelah itu klik Finish, tampilan susunan folder projet akan terlihat seperti ini,



Muncul 3 File yang sudah ter generate secara otomatis, file fxml untuk mendesain layout dari aplikasi, lalu untuk controller digunakan untuk memasukkan query program yang akan dijalankan pada program aplikasi ketika di Run, sedangkan untuk file JavaFX.java berlaku sebagai parent dari folder javafx.



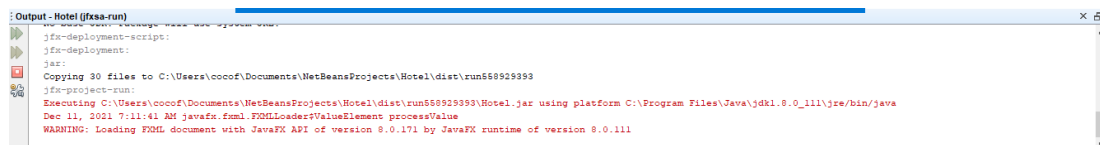
Setelah tampilan susunan folder sudah seperti gambar, double-klik pada file fxml atau bisa dengan klik kanan lalu pilih Open, maka file akan terbuka di Scene Builder untuk mendesain layout aplikasi,



Tampilan awal dari file fxml yang dibuka di dalam Scene Builder akan seperti gambar diatas, konsep dari scene builder hampir sama seperti Java GUI pada umumnya yaitu dengan Drag and Drop komponen untuk membuat desain.

Jika tampilan Scene Builder sudah seperti gambar diatas, sekarang kita coba Run project dengan klik F6 dan klik Button Click Me!, maka akan menampilkan pesan *Hello World*, jadi ini untuk menampilkan Hello World pada JavaFX.

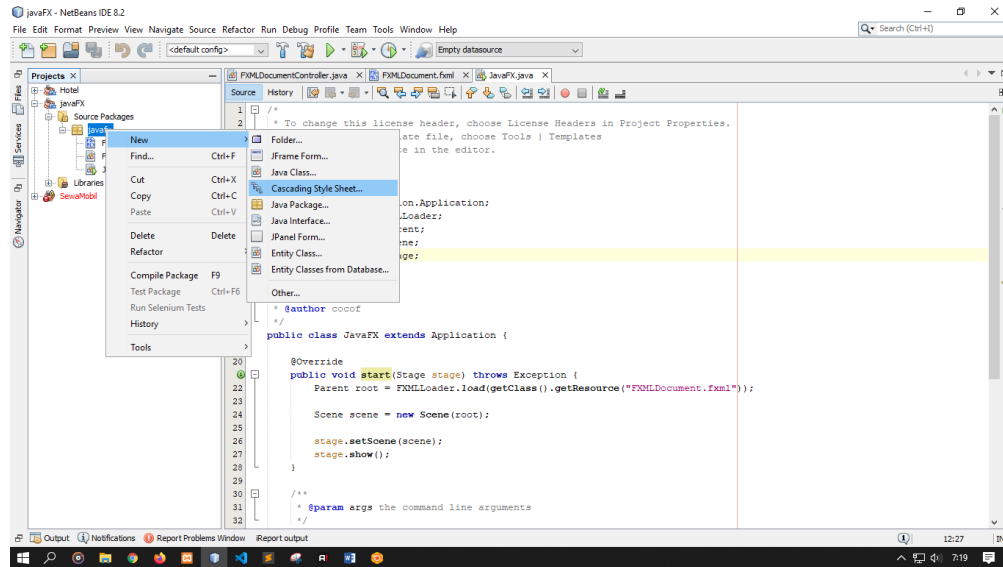
Perlu diketahui, jika sewaktu project di Run dan pada Output muncul seperti berikut,



Jika muncul Output dengan font merah seperti itu, abaikan karena itu bukan error message melainkan hanya message proses berjalannya aplikasi JavaFX.

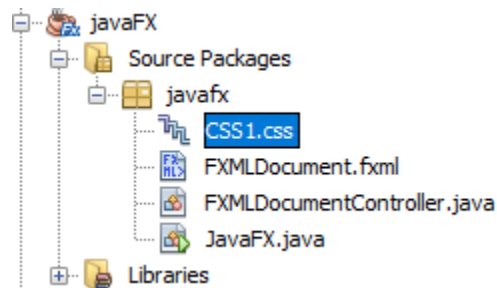
### 3. CSS Style JavaFX

Unik dari JavaFX antara lain bisa di beri style dari CSS, setelah membuat project seperti diatas, buat file baru di dalam folder satu tempat dengan file fxm1,



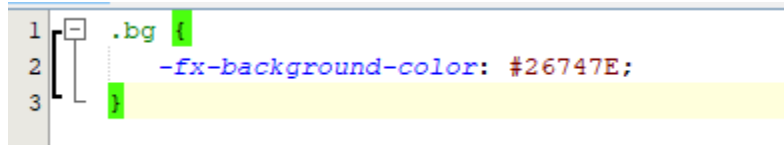
New > pilih Cascading Style Sheet.

Setelah finish, dan file CSS akan muncul di dalam folder satu tempat dengan file fxm1,



Untuk mengedit CSS, hanya perlu double-klik pada file CSS, sekarang kita buat class baru pada file CSS,

Pembeda dari CSS pada JavaFX, kalau di JavaFX hanya menambahkan (-fx-) pada awalan query



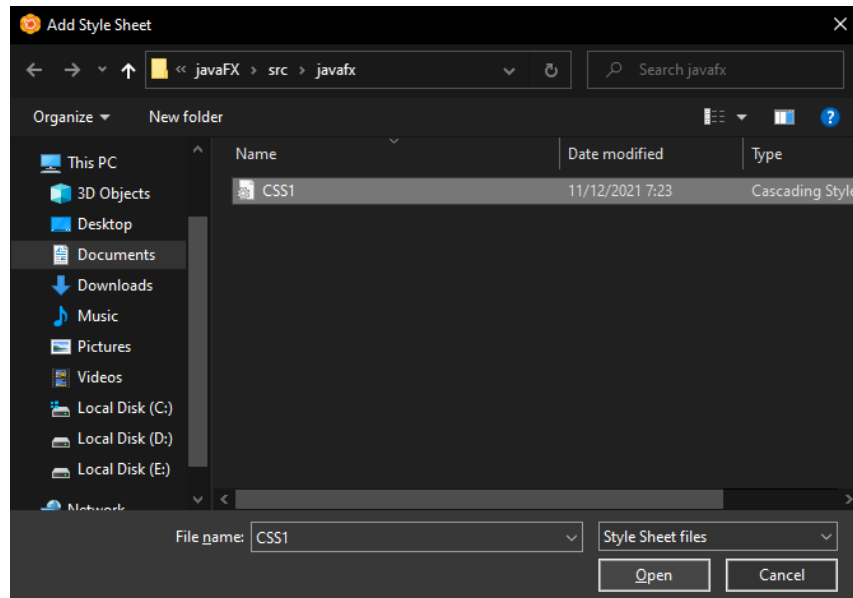
CSS contoh untuk ,

background color -> **-fx-background-color : warna;**

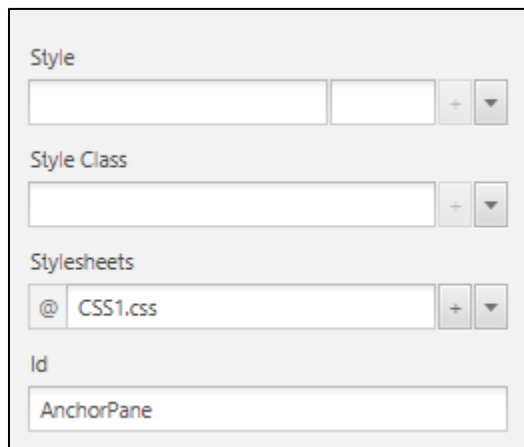
warna form -> **-fx-text-fill: warna;**

warna border -> **-fx-border-color: warna;** dan Query CSS lainnya.

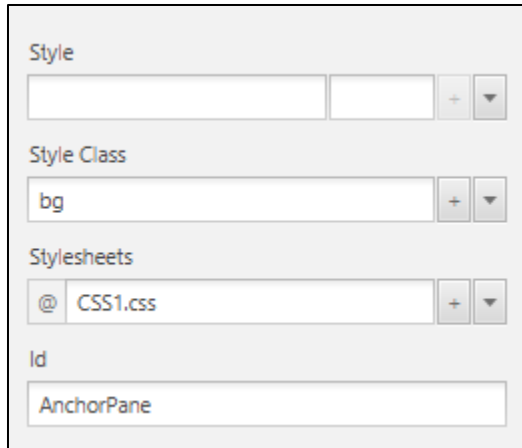
buat class .bg untuk mengatur warna background pada CSS, maka untuk memanggil class tersebut kita kembali ke Scene Builder, lalu klik bagian background aplikasi atau bagian AnchorPane, pada opsi Properties scroll kebawah akan muncul pilihan style, pada StyleSheet klik tombol (+) akan muncul open file dan pilih file CSS yang sudah dibuat tadi,



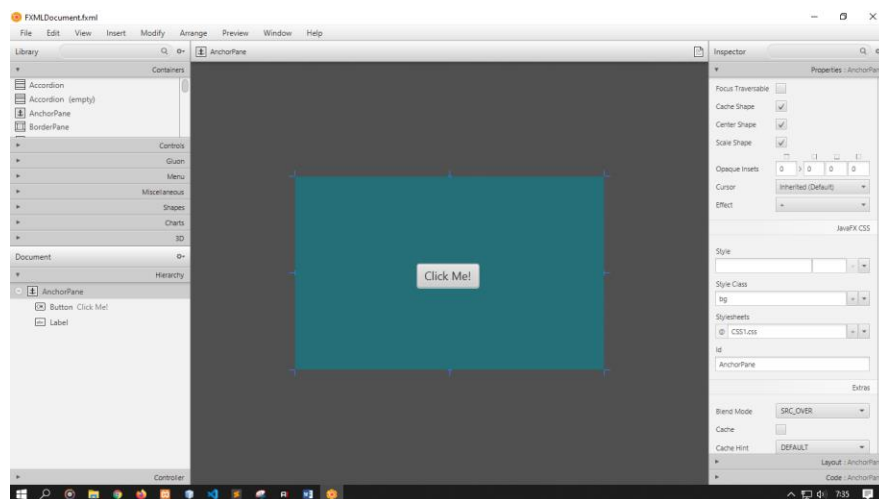
Selanjutnya pada kolom stylesheet di Scene Builder tadi akan muncul nama file CSS yang sudah dipilih.



Untuk memanggil class yang sudah kita ketik hanya tinggal menambahkan nama class pada kolom Style Class, ketik nama class (bg),



Jika class sudah terpilih maka design query akan langsung mengubah design dari aplikasi.

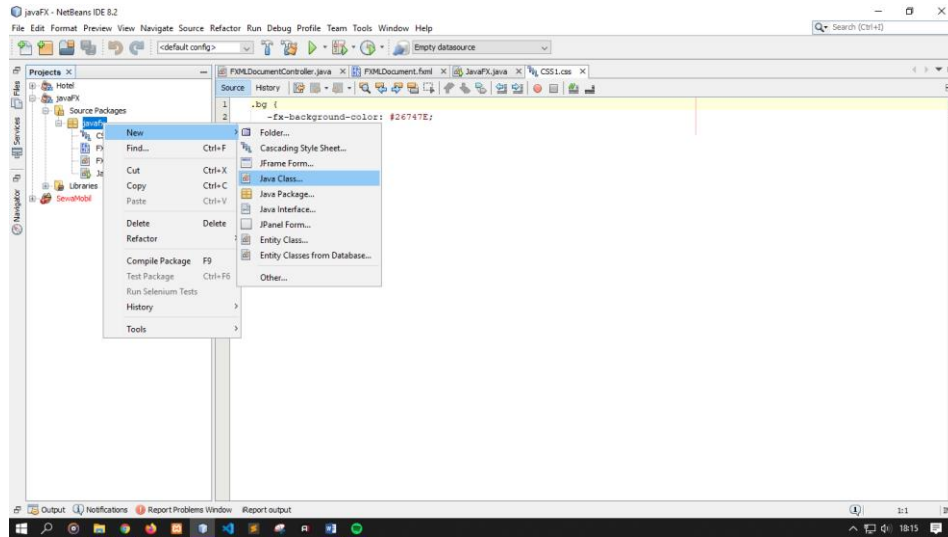


Query CSS akan langsung mengubah design dari layout aplikasi.

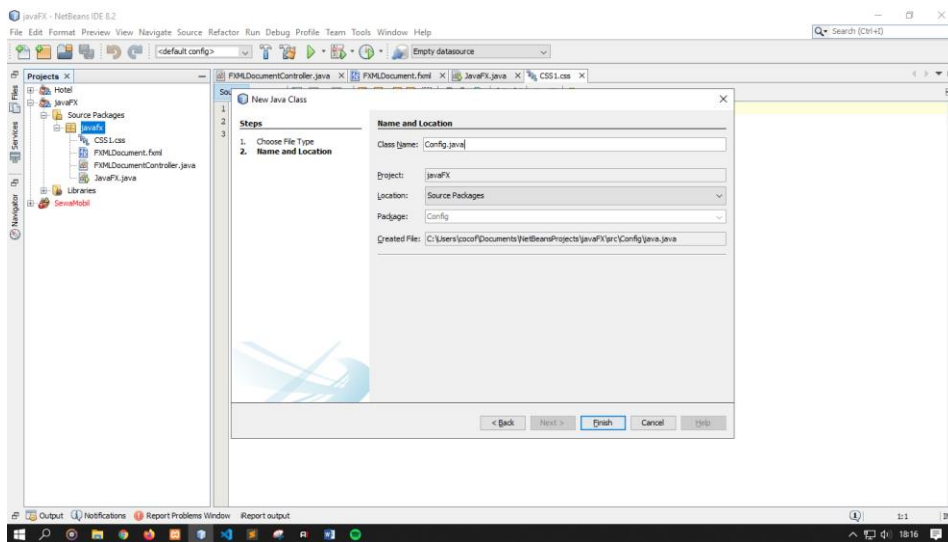
Seperti itu penggunaan CSS Style pada layout JavaFX menggunakan Scene Builder.

#### 4. Koneksi Database

Membuat sebuah project aplikasi CRUD sederhana, melalui proses login, folder project sama dengan yang sudah dibuat diatas, kita hanya perlu memodifikasi dan menambahkan beberapa bagian saja. Untuk bagian koneksi, kita buat file Java Class baru, klik kanan pada folder packages yang sudah dibuat tadi, New > kemudian pilih Java Class.



Beri nama file tersebut dengan nama Config.java.



Setelah membuat file, perlu menambahkan syntax query untuk menghubungkan dengan database MySQL, sebelum itu jangan lupa XAMPP harus sudah *Start*, setelah XAMPP sudah *Start*, buat Database di PhpMyAdmin atau lewat Command Prompt,

```
C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 40
Server version: 10.4.13-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

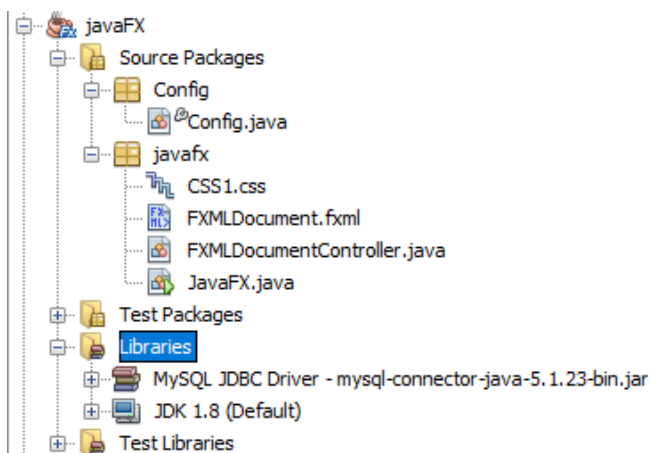
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database CobaFX;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| coba     |
| cobafx   |
| hotel    |
| information_schema |
| javafx   |
| mysql    |
| performance_schema |
| phpmyadmin |
| pos_db   |
| reksho_datatable |
| reksho_ecommerce |
| test     |
+-----+
12 rows in set (0.002 sec)

MariaDB [(none)]> use cobafx;
Database changed
MariaDB [cobafx]>
```

setelah database sudah terbuat tambah Library MySQL JDBC Driver pada Netbeans, di project Library > Add Library,



Setelah menambahkan Library, pada file Config.java tambahkan syntax berikut,

```
package Config;

import java.sql.Connection;
import java.sql.DriverManager;

/**
 *
 * @author cocof
 */
public class Config {
    Connection koneksi = null;
    public static Connection koneksiDB(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/cobaFX", "root", "");
            System.out.println("Koneksi Berhasil");
            return conn;
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Koneksi Gagal");
            return null;
        }
    }
}
```

Sedikit penjelasan, didalam file Config.java tadi kita tambahkan static method dengan nama koneksiDB, lalu di dalam method tersebut kita memanggil Library MySQL JDBC Driver yang sudah ditambah tadi, *Class.forName("com.mysql.jdbc.Driver)*, setelah itu memanggil database yang akan dihubungkan dengan syntax,

***DriverManager.getConnection("jdbc:mysql://localhost/nama\_database", "root", "");***

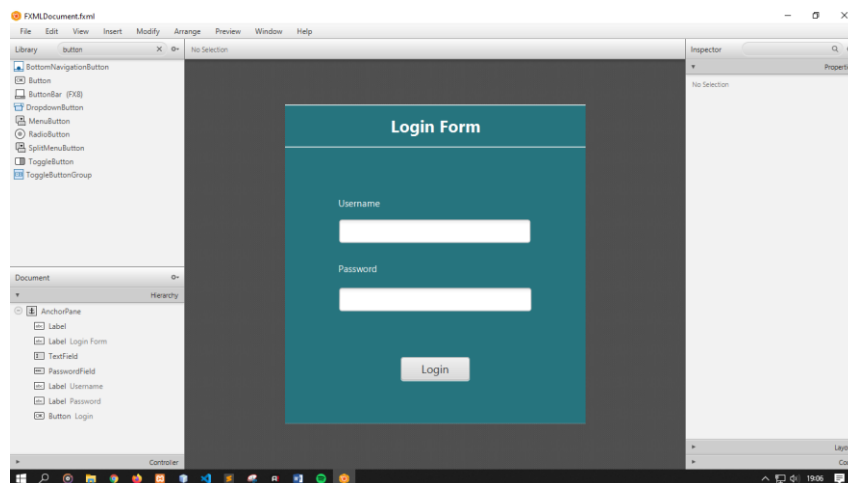
Untuk tes koneksi apakah sudah terkoneksi dengan database atau belum, kita cukup memanggil static method yang sudah dibuat di file parents untuk folder javafx yaitu file JavaFX.java .

```
public class JavaFX extends Application {  
    @Override  
    public void start(Stage stage) throws Exception {  
        Config.koneksiDB();  
        Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));  
  
        Scene scene = new Scene(root);  
  
        stage.setScene(scene);  
        stage.show();  
    }  
}
```

Setelah itu coba Run project, maka hasil yang muncul di dalam kolom output akan menampilkan hasil println apakah koneksi berhasil atau gagal,

```
Copying 13 files to C:\Users\cocof\Documents\NetBeansProjects\javaFX\dist\run1281278317  
jfx-project-run:  
Executing C:\Users\cocof\Documents\NetBeansProjects\javaFX\dist\run1281278317\javaFX.jar using platform C:\Program Files\Java\jdk1.8.0_111\jre\bin\java  
Koneksi Berhasil  
Dec 11, 2021 6:50:23 PM javafx.fxml.FXMLLoader$ValueElement processValue  
WARNING: Loading FXML document with JavaFX API of version 8.0.171 by JavaFX runtime of version 8.0.111
```

Sekarang ubah desain layout pada file fxml, double-klik pada file fxml, yang akan digunakan untuk desain layout form login, untuk mendesain layout cukup Drag and Drop komponen yang sudah disediakan sesuai kemauan,



Untuk desain warna boleh abaikan, hanya perlu untuk desain-desain komponen jika sudah mengatur tampilan desain tekan Ctrl + S untuk menyimpan progress pada Scene Builder, jika sudah seperti itu kita coba Run project lagi untuk memastikan bahwa layout design sudah tersimpan.



Jika sudah, sekarang akan mengubah atau memberi query Syntax untuk proses validasi login berdasarkan data user yang sudah terdaftar pada database, di dalam database kita akan membuat tabel baru dengan nama tb\_admin,

```
MariaDB [(none)]> use cobafx
Database changed
MariaDB [cobafx]> show tables;
Empty set (0.001 sec)

MariaDB [cobafx]> create table tb_admin(id_admin int(11) not null auto_increment, username varchar(255) not null, password varchar(255) not null, primary key(id_admin));
Query OK, 0 rows affected (0.275 sec)

MariaDB [cobafx]> show tables;
+-----+
| Tables_in_cobafx |
+-----+
| tb_admin          |
+-----+
1 row in set (0.001 sec)

MariaDB [cobafx]> desc tb_admin;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id_admin | int(11) | NO | PRI | NULL | auto_increment |
| username | varchar(255) | NO | | NULL | |
| password | varchar(255) | NO | | NULL | |
+-----+-----+-----+-----+-----+
3 rows in set (0.104 sec)

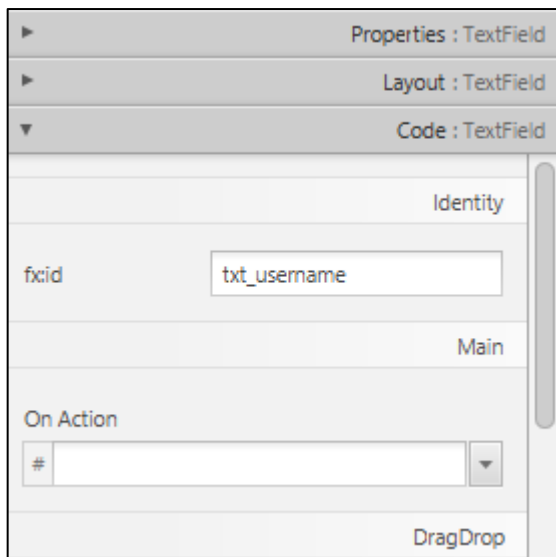
MariaDB [cobafx]>
```

Dengan kolom

- id\_admin int(11) PK
- username varchar(255)
- password varchar(255)

atau bisa membuat tabel melalui PhpMyAdmin untuk jalur lebih mudahnya, setelah membuat tabel sekarang akan mengubah syntax pada Controller dari fxml layout login form tadi.

Jangan lupa untuk memberi id sebagai nama dari setiap komponen di Scene Builder,



Sekarang ubah syntax pada file Controller seperti berikut,  
sebelum itu jangan lupa untuk membuat deklarasi nama dari komponen input field,

```
@FXML
private Label label;
@FXML
private TextField txt_username;
@FXML
private PasswordField txt_password;
```

pada file Controller ubah syntax menjadi seperti ini,

```
public class FXMLDocumentController implements Initializable {

    @FXML
    private Label label;
    @FXML
    private TextField txt_username;
    @FXML
    private PasswordField txt_password;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    Connection con;
    PreparedStatement ps;
    ResultSet rs;

    public FXMLDocumentController() {
        con = Config.koneksiDB();
    }

    //buat method baru
    public void login(ActionEvent event){
        String username = txt_username.getText();
        String password = txt_password.getText();

        String user = "";
        String pw = "";

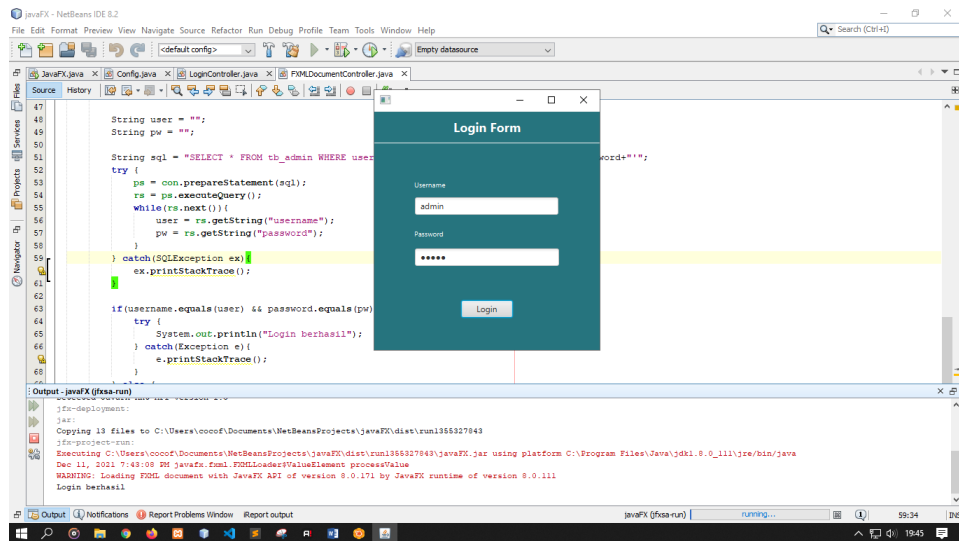
        String sql = "SELECT * FROM tb_admin WHERE username='"+username+"' AND password='"+password+"'";
        try {
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            while(rs.next()){
                user = rs.getString("username");
                pw = rs.getString("password");
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }

        if(username.equals(user) && password.equals(pw)){
            try {
                System.out.println("Login berhasil");
            } catch (Exception e) {
                System.out.println("Login Gagal");
            }
        }
    }
}
```

Kalau masih ada error, itu hanya kesalah import saja, biasanya meng alt + Enter untuk error.

Setelah mengetik syntax, tambahkan method `ActionEvent` yang sudah dibuat ke dalam komponen Button di Scene Builder, setelah itu coba Run program dan coba masukan username dan password yang sudah ditambahkan, maka hasilnya akan menghasilkan output

*Login Berhasil,*



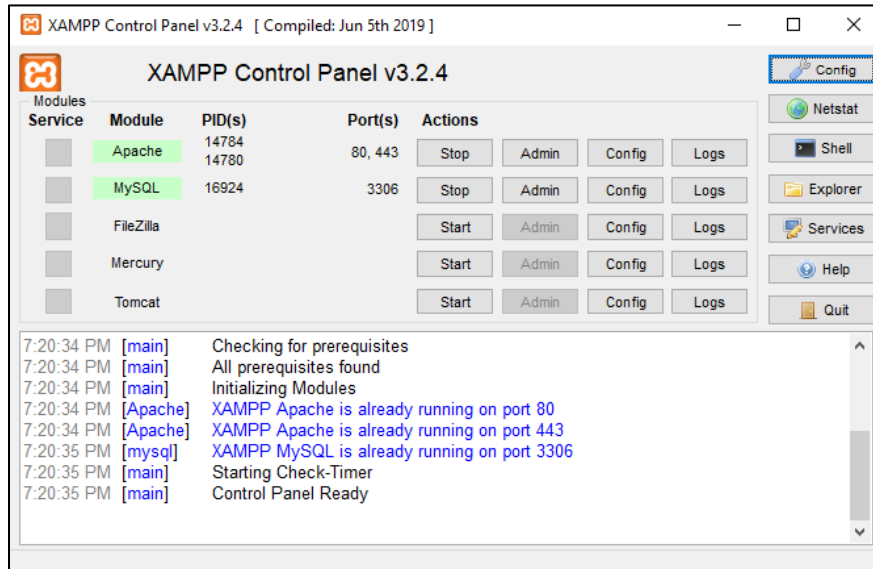
Sekarang kita berhasil menghubungkan koneksi ke dalam database hingga proses validasi login.

Sekarang kita akan mencoba membuat sebuah project CRUD sederhana menggunakan JavaFX, untuk source code dari proses CRUD memang tidak sepenuhnya benar, dan banyak petunjuk dari berbagai media seperti youtube, Google, dll.

## 5. Program CRUD Sederhana

Sebelum membuat sebuah project CRUD, pastikan sudah membuat database beserta tabel nya, Pada contoh ini akan membuat sebuah program CRUD pendataan mahasiswa sederhana menggunakan JavaFX dan MySQL.

- Pastikan XAMPP sudah siap



- Selanjutnya buat Database dengan nama Database 'mahasiswa', melalui command prompt atau phpMyAdmin. Contoh query pada command prompt,

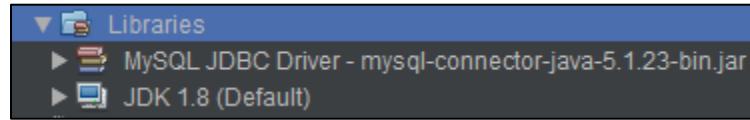
```
MariaDB [(none)]> create database mahasiswa;
```

- Setelah database terbuat, sekarang membuat tabel dengan nama table dt\_mhasiswa rincian tipe datanya seperti berikut,

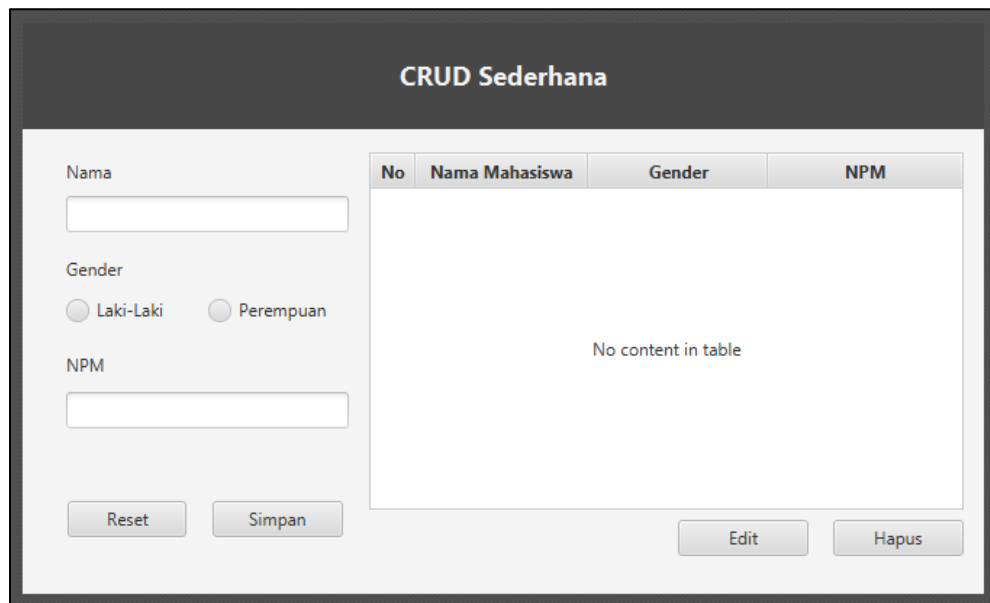
No.	Nama Kolom	Tipe Data	Value
1.	Id_mhsiswa (PK)	INT	11
2.	nama_mhsiswa	VARCHAR	255
3.	Gender	VARCHAR	255
4.	Npm	VARCHAR	255

- Setelah database dan tabel sudah dibuat, selanjutnya buat project baru pada Netbeans, File > New Project, pilih kategori JavaFX > JavaFX FXML Application.

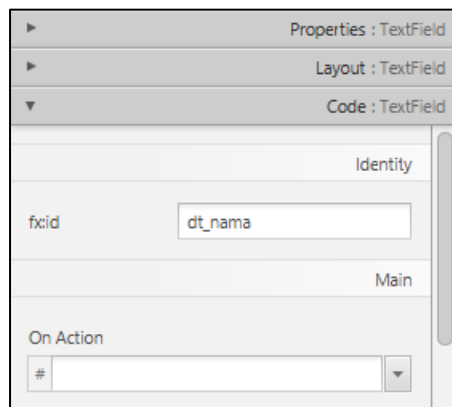
Jangan lupa untuk import libraries *MySQL JDBC Driver*.



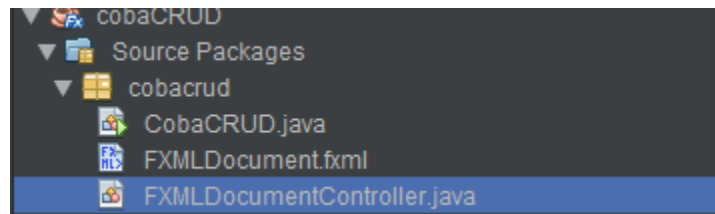
Setelah itu, buka file FXML seperti cara diatas, lalu buat desain pada file FXML di Scene Builder dengan Drag and Drop komponen yang diperlukan, minimal hingga tampilan menjadi seperti ini,



Setelah membuat desain dari layout, sekarang beri nama id pada masing-masing komponen dari Scene Builder seperti TextField, RadioButton, Button, hingga TableView, beri id pada pilihan **Code**,



Setelah semua komponen sudah diberi id, sekarang kita membuat syntax dalam file controller nya.



Lalu membuat sebuah syntax untuk fungsi Create, Read, Update dan Delete yang akan digunakan untuk data yang dimasukkan ke dalam aplikasi.

Sebelum membuat fungsi, panggil komponen yang sudah diberi id tadi dengan mendeklarasikan nama komponen tersebut,

```
@FXML
private RadioButton laki, perem;
@FXML
private TextField dt_nama, dt_npm;
@FXML
private Button reset, simpan;
@FXML
private TableView<ModelTable> tblView;
@FXML
private TableColumn<ModelTable, String> col_no;
@FXML
private TableColumn<ModelTable, String> col_nama;
@FXML
private TableColumn<ModelTable, String> col_gender;
@FXML
private TableColumn<ModelTable, String> col_npm;

ObservableList<ModelTable> list = FXCollections.observableArrayList();
```

Pastikan semua komponen sudah terpanggil agar komponen bisa saling terhubung ke dalam controller.

- **Koneksi Database**

Untuk koneksi bisa disamakan dengan cara membuat koneksi database contoh sebelumnya, hanya menyesuaikan nama database, dan jangan lupa untuk import library *MySQL JDBC Driver*, dengan syntax didalam file Config seperti berikut,

```
1 package cobacrud;
2
3 import java.sql.*;
4
5 /**
6  *
7  * @author ASUS
8  */
9 public class Config {
10     Connection con = null;
11     public static Connection koneksi() {
12         try {
13             Class.forName("com.mysql.jdbc.Driver");
14             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/mahasiswa","root","");
15             return conn;
16         } catch (Exception ex) {
17             ex.printStackTrace();
18         }
19         return null;
20     }
21 }
```

- **Insert Data**

Berikutnya untuk syntax yang berfungsi untuk memasukkan data yang sudah di-inputkan ke dalam database, dengan susunan query syntax seperti berikut,

```
30 public void dt_masuk(ActionEvent event) {
31     String nama = dt_nama.getText().toString();
32     String npm = dt_npm.getText().toString();
33     String gender = "";
34     if(laki.isSelected()){
35         gender = "Laki-Laki";
36     } else if(perem.isSelected()){
37         gender = "Perempuan";
38     }
39
40     try {
41         ps = con.prepareStatement("INSERT INTO dt_mhsiswa('nama_mhsiswa','gender','npm') "
42             + "VALUES(?, ?, ?)");
43         ps.setString(1, nama);
44         ps.setString(2, gender);
45         ps.setString(3, npm);
46         ps.execute();
47         Alert a = new Alert(AlertType.INFORMATION);
48         a.setTitle("Informasi Database");
49         a.setHeaderText(null);
50         a.setContentText("Data Berhasil Ditambahkan!!");
51         a.showAndWait();
52     } catch (SQLException e) {
53         e.printStackTrace();
54     }
55 }
```

**Penjelasan :**

Membuat sebuah method dengan nama `dt_masuk` dan dideklarasikan sebagai `ActionEvent`, didalam method tersebut berisi pertama yaitu pemanggilan data dari dalam komponen yang sudah dibuat, dengan membuat deklarasi tipe data `String` dengan nama

yang dibuat, lalu untuk pengambilan data dari RadioButton menggunakan percabangan if-else dengan logika kondisi jika RadioButton dengan id laki terpilih maka String gender akan terisi dengan data "Laki-Laki" sedangkan jika RadioButton perempuan terpilih maka String gender akan terisi dengan data "Perempuan" tiap nama dari variabel String akan digunakan untuk memasukkan data ke dalam database, buta preparedStatement dan variabel dari Connection seperti berikut,

```
Connection con = Config.koneksi();
PreparedStatement ps;
```

Setelah itu buat query SQL Insert dengan preparedStatement yang sudah dibuat, jika query berhasil dijalankan maka diujung program, Alert akan ter-execute menunjukkan pesan bahwa data berhasil disimpan.

- **Read Data**

Pada proses ini, dijelaskan proses dimana data yang sudah dimasukkan ataupun diubah di dalam database dapat ditampilkan ke dalam tampilan TableView,

Pertama, buat file Java Class baru dengan nama ModelTable, buat String dengan nama sebanyak kolom yang ada pada table di dalam database mahasiswa,

```
String id, nama, gender, npm;
```

Setelah membuat beberapa variabel, buat Constructor dari nama variabel yang sudah dibuat, dan dilanjutkan dengan menambahkan Getter and Setter pada variabel tersebut.

Ubah susunan kode di dalam file Controller untuk fungsi menampilkan data pada database ke dalam tableview,

```
74 private void load_table() {
75     col_no.setCellValueFactory(new PropertyValueFactory<>("id"));
76     col_nama.setCellValueFactory(new PropertyValueFactory<>("nama"));
77     col_gender.setCellValueFactory(new PropertyValueFactory<>("gender"));
78     col_npm.setCellValueFactory(new PropertyValueFactory<>("npm"));
79
80     try {
81         Connection con = Config.koneksi();
82         ResultSet rs = con.createStatement().executeQuery("SELECT * FROM dt_mhsiswa");
83
84         while(rs.next()) {
85             list.add(new ModelTable(rs.getString("id_mhsiswa"), rs.getString("nama_mhsiswa"),
86                                     rs.getString("gender"), rs.getString("npm")));
87         }
88         tblView.setItems(list);
89     } catch (SQLException ex) {
90         ex.printStackTrace();
91     }
92 }
```



### Penjelasan :

Membuat sebuah private method baru dengan nama load\_table, didalam berisi pemanggilan dari nama id setiap kolom table, lalu mengambil data dari database menggunakan query SQL Select, lalu setItems tableview dengan data yang sudah diambil dari database, tambahkan method load\_table pada method initialize sehingga waktu aplikasi di Run maka data yang diambil dari database akan tampil pada tableview.

- **Update Data**

Pada proses update data, menggunakan query SQL untuk update data dengan data yang diambil berdasarkan id dari kolom yang diambil, mengambil value dari kolom dengan method yang sudah otomatis terbuat ketika membuat Getter and Setter di dalam file ModelTable, Query Syntax seperti berikut,

```
private void update() {
    String nma = dt_nama.getText().toString();
    String npm = dt_npm.getText().toString();
    String gender = "";
    if(laki.isSelected()){
        gender = "Laki-Laki";
    } else if(perem.isSelected()){
        gender = "Perempuan";
    }
}

try {
    TableView(tableViewSelectionModel<ModelTable> selectionModel = tblView.getSelectionModel();
    selectionModel.setSelectionMode(SelectionMode.SINGLE);

    ObservableList<ModelTable> selectedItem = selectionModel.getSelectedItems();
    ModelTable tb = tblView.getSelectionModel().getSelectedItem();

    String id = tb.getId();

    String up = "UPDATE dt_mhsiswa SET nama_mhsiswa='"+nma+"', "
        + "gender='"+gender+"', npm='"+npm+"' WHERE id_mhsiswa = '"+id+"'";

    ps = con.prepareStatement(up);
    ps.execute();
    tblView.getItems().clear();
    load_table();

    Alert a = new Alert(AlertType.INFORMATION);
    a.setTitle("Informasi Database");
    a.setHeaderText(null);
    a.setContentText("Data Berhasil di Update!!");
    a.showAndWait();

} catch (SQLException ex) {
    ex.printStackTrace();
}
}
```

### Penjelasan :

Membuat sebuah method baru dengan nama method update, didalam method berisi query SQL update, dengan memanggil data dari variabel melalui file ModelTable, jika proses berhasil maka proses alert akan ter-execute dan menampilkan pesan bahwa update data berhasil.

- **Delete Data**

Pada proses delete data dimana proses ini akan menghapus data baik dari tampilan tableview maupun didalam database, menggunakan query sql delete untuk menghapus data didalam database, dengan Query Syntax seperti berikut,

```
public void delete(ActionEvent event){
    try {
        TableView<TableViewSelectionModel<ModelTable>> selectionModel = tblView.getSelectionModel();
        selectionModel.setSelectionMode(SelectionMode.SINGLE);

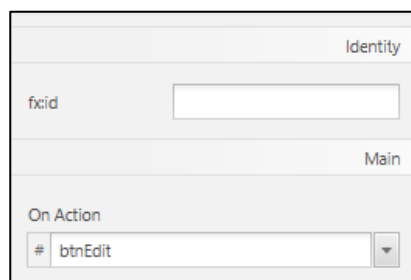
        ObservableList<ModelTable> selectedItem = selectionModel.getSelectedItems();
        ModelTable tb = tblView.getSelectionModel().getSelectedItem();

        String id = tb.getId();
        String del = "DELETE FROM dt_mhsiswa WHERE id_mhsiswa='"+id+"'";

        ps = con.prepareStatement(del);
        ps.execute();
        Alert a = new Alert(AlertType.INFORMATION);
        a.setTitle("Informasi Database");
        a.setHeaderText(null);
        a.setContentText("Data berhasil dihapus!!");
        a.showAndWait();
        tblView.getItems().clear();
        load_table();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}
```

### Penjelasan :

Hampir sama dengan proses update data, pada proses delete data, data sebagai where clause diambil dari id yang diambil dari dalam ModelTable lalu dilanjutkan dengan execute query SQL Delete, jika proses delete berhasil alert akan menunjukkan pesan bahwa data berhasil di hapus, jangan lupa untuk menambahkan nama method pada ActionEvent di setiap button pada SceneBuilder,



Source Code : [https://github.com/PororoChan/JavaFX\\_Sc\\_Modul](https://github.com/PororoChan/JavaFX_Sc_Modul)