
VERSION 1.0
14 MARET , 2024



PEMROGRAMAN BERORIENTASI OBJEK

MODUL 6 - PEMROGRAMAN GUI DENGAN JAVA FX.

DISUSUN OLEH:
TAUFIQ RAMADHAN
SUTRISNO ADIT PRATAMA

DIAUDIT OLEH:
IR. GALIH WASIS WICAKSONO, S.KOM, M.CS

LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN BERORIENTASI OBJEK

PERSIAPAN MATERI

1. Java Development Kit.
2. Text Editor / IDE (Visual Studio Code, Netbeans, IntelliJ IDEA, atau yang lainnya).

TUJUAN

1. Mahasiswa dapat memahami java gui .
2. Mahasiswa dapat memahaami komponen dalam JavaFx

TARGET MODUL

1. Mahasiswa dapat mengimplementasikan java gui.
2. Mahaasiswa Dapat mengimplementasikan javafx

KEYWORDS

1. JavaFX
2. Java GUI
3. Form JavaFx

TEORI

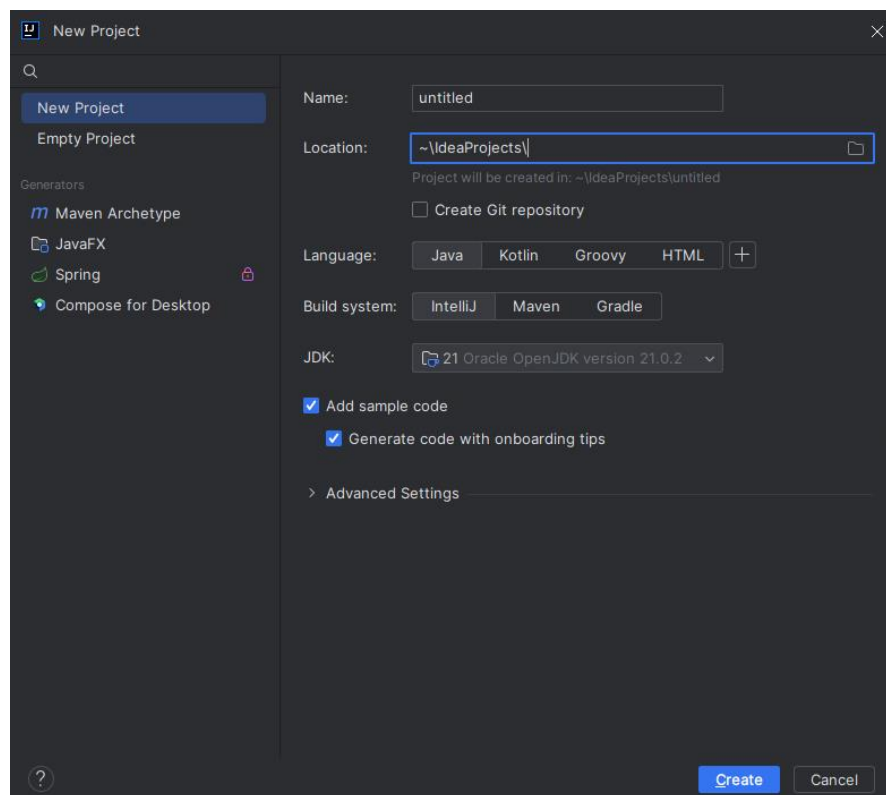
- **Pengertian JavaFx**

JavaFX adalah sebuah platform software untuk membangun sebuah Rich Internet Application (RIA) yang bisa berjalan pada berbagai macam perangkat. Seperti komputer dekstop, web browser di Windows, Linux dan Mac OSX. JavaFX dirancang untuk menyediakan para developer java sebuah platform yang baru, ringan, dengan performa yang tinggi. Tujuannya adalah ingin menggantikan swing untuk membangun aplikasi GUI. Namun itu bukan berarti Swing sudah tidak digunakan lagi. Sejumlah besar aplikasi telah dibangun dengan menggunakan swing itu berarti, Swing telah menjadi bagian dari Java API untuk waktu yang cukup lama. Terutama bahwa aplikasi ini bisa digabungkan dengan JavaFX secara fungsi.

Konfigurasi IntelliJ IDEA

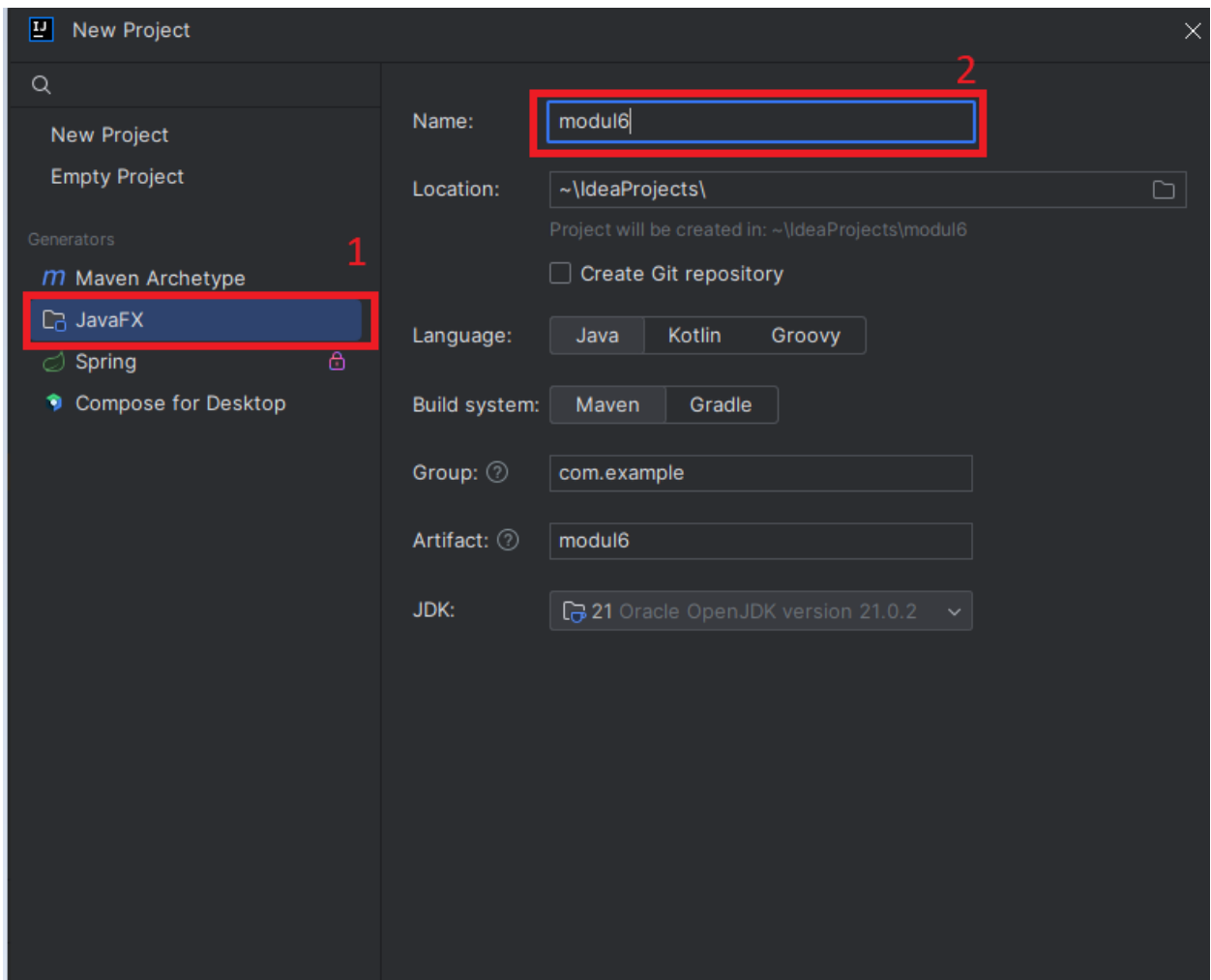
- **Pilih New Project**

Maka jendela new project keluar seperti contoh dibawah ini



- **Arahkan Ke Generator JavaFx**

Pada sidebar kiri pilih generator javaFx seperti petunjuk dibawah ini dan juga jangan lupa mengisi nama project nya. Setelah itu Pilih **NEXT**



MATERI

- **Default JavaFX Project**

```
package org.example.modul0prak;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.io.IOException;

public class HelloApplication extends Application {
    @Override
    public void start(Stage stage) throws IOException {
    }

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

Penjelasan:

1. Metode **start()** dalam program Java yang menggunakan JavaFX adalah titik masuk utama untuk aplikasi JavaFX.
2. main Method: Memanggil **launch(args)** yang memanggil fungsi **launch()** aplikasi JavaFX.

- **Library javafx.scene.layout**

```
import javafx.scene.layout.*;
```

dalam JavaFX menyediakan berbagai kelas layout yang membantu mengatur tata letak komponen UI di dalam aplikasi. Setiap jenis layout memiliki aturan dan fitur khusus yang memungkinkan pengembang untuk mengelola tata letak elemen secara fleksibel dan efektif. Berikut adalah beberapa kelas layout utama yang tersedia dalam javafx.scene.layout beserta penjelasannya:

- **CLASS LAYOUT**

1. **VBox (Vertical Box)**

- Menyusun anak-anaknya dalam kolom vertikal.
- Komponen ditempatkan dari atas ke bawah.

Contoh penggunaan:

```
VBox vbox = new VBox(10);
Button btn1 = new Button("Button 1");
Button btn2 = new Button("Button 2");
vbox.getChildren().addAll(btn1, btn2);
```

2. HBox(Horizontal Box)

- Menyusun anak-anaknya dalam baris horizontal.
- Komponen ditempatkan dari kiri ke kanan.

Contoh penggunaan:

```
HBox hbox = new HBox(10);
Button btn3 = new Button("Button 3");
Button btn4 = new Button("Button 4");
hbox.getChildren().addAll(btn3, btn4);
```

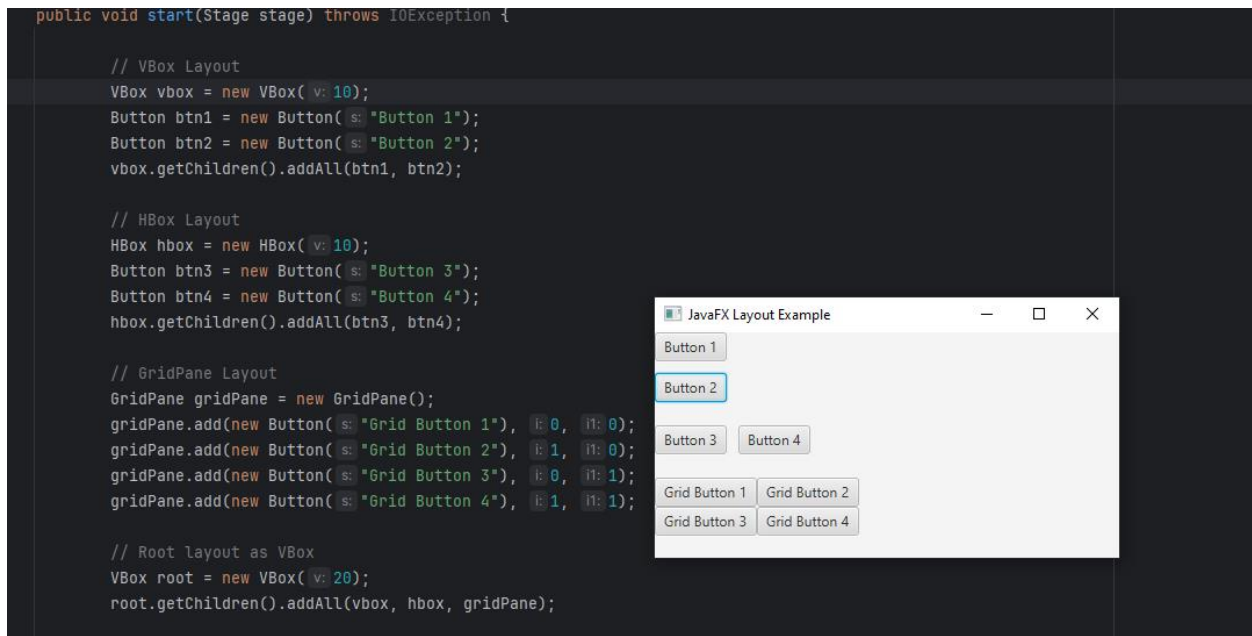
3. GridPane:

- Mengatur anak-anaknya dalam grid (tabel) yang terdiri dari baris dan kolom.
- Komponen dapat ditempatkan di sel tertentu dengan spesifikasi baris dan kolom.

Contoh Penggunaan:

```
GridPane gridPane = new GridPane();
gridPane.add(new Button("Grid Button 1"), 0, 0);
gridPane.add(new Button("Grid Button 2"), 1, 0);
gridPane.add(new Button("Grid Button 3"), 0, 1);
gridPane.add(new Button("Grid Button 4"), 1, 1);
```

Hasil Run



Membuat Button di JavaFX

1. **Import library 'javafx.scene.control.*'** Dalam JavaFX, paket `javafx.scene.control` menyediakan berbagai komponen kontrol (UI controls) yang dapat digunakan untuk membuat antarmuka pengguna (user interface). Komponen-komponen ini mencakup berbagai elemen yang sering digunakan dalam aplikasi GUI, seperti tombol, label, text field, menu, dan banyak lagi.

```
import javafx.scene.control.*;
```

2. Tulis Code berikut

```
VBox vbox = new VBox(10);
Button btn1 = new Button("Button 1");
vbox.getChildren().addAll(btn1);

VBox root = new VBox(20);
root.getChildren().addAll(vbox);

Scene scene = new Scene(root, 400, 600);
stage.setTitle("Button Test");
stage.setScene(scene);
stage.show();
```

Penjelasan :

- **VBox vbox = new VBox(10);**
Membuat sebuah VBox dengan jarak antar komponen (children) sebesar 10 piksel. VBox adalah layout container yang mengatur komponen-komponen secara vertikal.
- **Button btn1 = new Button("Button 1");**
Membuat sebuah tombol dengan teks "Button 1".
- **vbox.getChildren().addAll(btn1);**
Menambahkan tombol btn1 ke dalam vbox sebagai anak (child) dari vbox.
- **VBox root = new VBox(20);**
Membuat sebuah VBox lain bernama root dengan jarak antar komponen sebesar 20 piksel. root ini akan digunakan sebagai layout utama dari scene.
- **root.getChildren().addAll(vbox);**
Menambahkan vbox ke dalam root. Ini berarti vbox, yang berisi tombol btn1, sekarang menjadi bagian dari root.
- **Scene scene = new Scene(root, 400, 600);**
Membuat sebuah scene dengan root sebagai layout utama dan ukuran jendela sebesar 400 piksel lebar dan 600 piksel tinggi. Scene adalah wadah untuk semua konten yang ditampilkan di stage.
- **stage.setTitle("Button Test");**
Mengatur judul dari jendela (stage) menjadi "Button Test".
- **stage.setScene(scene);**

Mengatur scene dari stage ke scene yang baru saja dibuat. Ini menghubungkan layout yang sudah disusun ke jendela aplikasi.

- **stage.show();**

Menampilkan jendela aplikasi. Metode ini memulai tampilan GUI aplikasi.

3. Membuat Button berfungsi

Skenario: saat button di klik maka dibawahnya menampilkan output “ button berfungsi”

- Buat objek label untuk pesan button berfungsi

```
Label outputLabel = new Label();
```

- Tambah code berikut

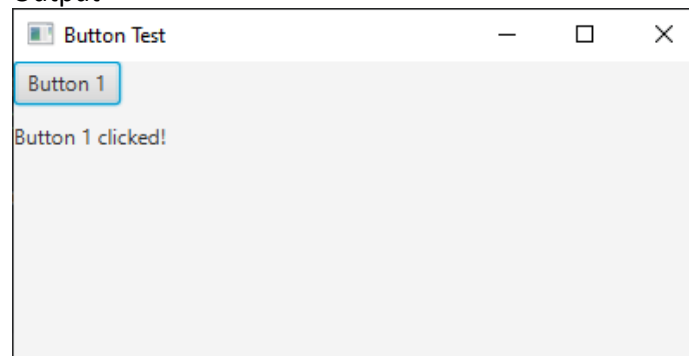
```
btn1.setOnAction(event -> {
    outputLabel.setText("Button 1 clicked!");
});
```

Penjelasan :

1. **btn1** adalah objek dari kelas Button, yaitu tombol yang telah dibuat sebelumnya dengan label "Button 1".
 2. **setOnAction** adalah metode yang digunakan untuk menetapkan event handler ke tombol. Event handler ini akan menentukan tindakan yang dilakukan ketika tombol diklik.
 3. **event -> { ... }** adalah lambda expression yang digunakan untuk mendefinisikan event handler. Lambda expression ini adalah cara singkat untuk menulis instance dari antarmuka fungsional, dalam hal ini, antarmuka `EventHandler<ActionEvent>`.
 4. **setText("Button 1 clicked!")** adalah metode yang mengubah teks yang ditampilkan oleh outputLabel.
- Tambahkan label dan button ke vbox

```
vbox.getChildren().addAll(btn1, outputLabel)
```

- Output



Membuat TextField / TextBox di JavaFX

1. Buat Object TextField dan juga button

```
TextField textField = new TextField();
textField.setPromptText("Masukkan Text : "); // Berfungsi Sebagai
Placeholder Pada Textfield

Button btn1 = new Button("Submit"); //Sebagai tombol submit
Label outputLabel = new Label();
```

2. Tambah code action untuk menampilkan text yang sudah di input oleh user

```
btn1.setOnAction(event -> {

    String inputText = textField.getText(); // mengambil value dari
    inputan textfield

    outputLabel.setText("Output : " + inputText);
});
```

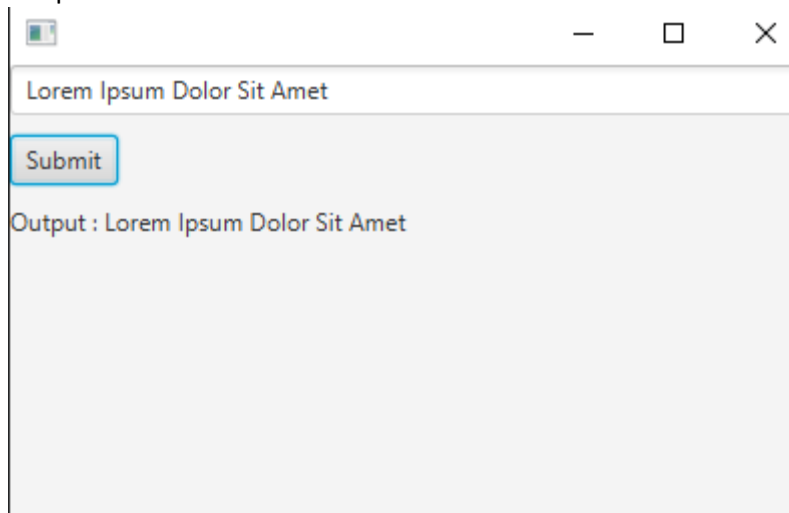
3. Tambahkan komponen ke vbox

```
vbox.getChildren().addAll(textField, btn1, outputLabel); //
Menambahkan komponen textfield,btn,outputlabel ke vbox

VBox root = new VBox(20);
root.getChildren().addAll(vbox);

Scene scene = new Scene(root, 400, 600);
stage.setScene(scene);
stage.show();
```

4. Output



KEGIATAN PERCOBAAN

PERCOBAAN 1

Membuat Form JavaFx

Membuat form adalah dasar dalam membangun aplikasi. Baik tidaknya aplikasi form akan menentukan keberhasilan aplikasi. Sebuah aplikasi biasanya tersusun dari beberapa form yang saling berinteraksi. Form yang baik adalah form yang user friendly dengan pengguna sehingga pengguna sangat nyaman menggunakan aplikasi. Salah satu form yang kita temui pada setiap aplikasi adalah form login. Berikut langkah-langkahnya :

1. Pada Method start() isikan script seperti berikut :

```
@Override
public void start(Stage primaryStage) throws Exception {
    primaryStage.setTitle("Form Login");

    primaryStage.show();
}
```

2. Setelah itu tambahkan script di bawah sebelum kode primaryStage.show();

```
GridPane grid = new GridPane();
grid.setAlignment(Pos.CENTER);
grid.setHgap(10);
grid.setVgap(10);
grid.setPadding(new Insets(25, 25, 25, 25));

Scene scene = new Scene(grid, 300, 275);
primaryStage.setScene(scene);
```

3. Kemudian tambahkan script dibawah ini untuk menambahkan title, label username, textfield username, label password, dan textfield password.

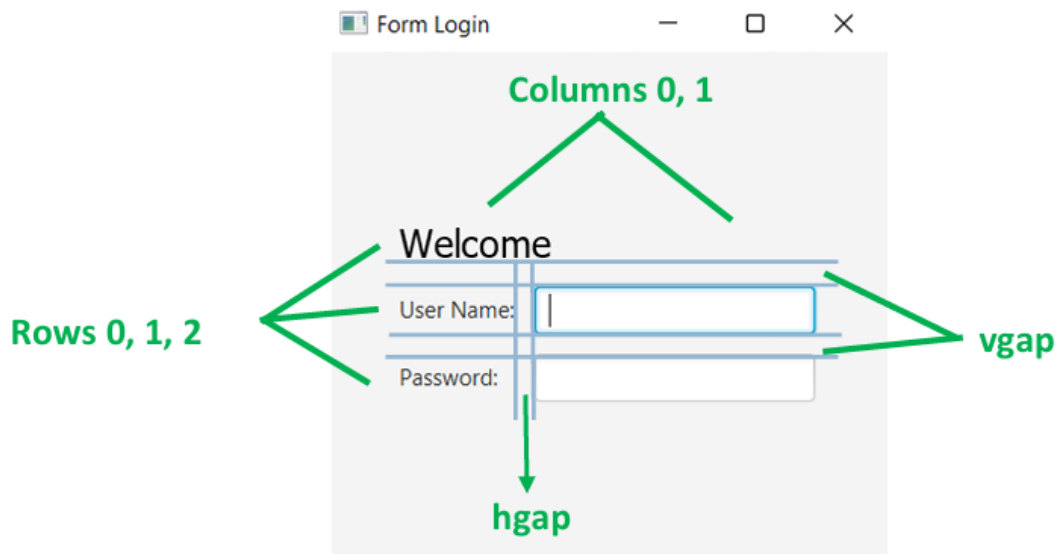
```
Text scenetitle = new Text("Welcome");
scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
grid.add(scenetitle, 0, 0, 2, 1);

Label userName = new Label("User Name:");
grid.add(userName, 0, 1);

Text scenetitle = new Text("Welcome");
scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
grid.add(scenetitle, 0, 0, 2, 1);

Label userName = new Label("User Name:");
grid.add(userName, 0, 1);
```

Tampilan program login akan seperti di bawah ini :



4. Kemudian kita akan menambahkan Button dan Text untuk menampilkan pesan dengan menambahkan script seperti berikut : `Button btn = new Button("Sign in");`

```
Button btn = new Button("Sign in");
HBox hbBtn = new HBox(10);
hbBtn.setAlignment(Pos.BOTTOM_RIGHT);
hbBtn.getChildren().add(btn);
```

Tambahkan Text control untuk mendisplay message.

```
final Text actiontarget = new Text();
grid.add(actiontarget, 1, 6);
```

Selanjutnya adalah menambahkan event handling dari button dengan menambahkan script seperti berikut ini :

```
btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent e) {
        actiontarget.setFill(Color.FIREBRICK);
        actiontarget.setText("Sign in button pressed");
    }

});
```

PERCOBAAN 2

Membuat Table JavaFx

1. Membuat tampilan awal

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.stage.Stage;

public class Main extends Application {

    private TableView table = new TableView();
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage stage) {
        Scene scene = new Scene(new Group());
        stage.setTitle("Test TableView");
        stage.setWidth(450);
        stage.setHeight(550);
        final Label label = new Label("Daftar Mahasiswa");
        label.setFont(new Font("Arial", 30));

        table.setEditable(true);

        TableColumn nameCol = new TableColumn("Nama");
        TableColumn nimCol = new TableColumn("NIM");
        TableColumn emailCol = new TableColumn("Email");

        table.getColumns().addAll(nameCol, nimCol, emailCol);

        final VBox vbox = new VBox();
        vbox.setSpacing(8);
        vbox.setPadding(new Insets(20, 10, 10, 10));
        vbox.getChildren().addAll(label, table);

        ((Group) scene.getRoot()).getChildren().addAll(vbox);

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

2. Menambahkan Data Pada Tabel

Untuk menambahkan data pada table, pertama buatlah sebuah class contohnya class Mahasiswa di dalam class Main yang isinya sebagai berikut :

```
public static class Mahasiswa {
    private final SimpleStringProperty name;
    private final SimpleStringProperty nim;
    private final SimpleStringProperty email;

    private Mahasiswa(String name, String nim, String email) {
        this.name = new SimpleStringProperty(name);
        this.nim = new SimpleStringProperty(nim);
        this.email = new SimpleStringProperty(email);
    }

    public String getName() {
        return name.get();
    }

    public void setName(String fName) {
        name.set(fName);
    }

    public String getNim() {
        return nim.get();
    }

    public void setNim(String fName) {
        nim.set(fName);
    }

    public String getEmail() {
        return email.get();
    }

    public void setEmail(String fName) {
        email.set(fName);
    }
}
```

3. Buatlah ObservableList array untuk menentukan seberapa banyak baris data yang ingin anda tampilkan di table anda.

```
final ObservableList<Mahasiswa> data =
FXCollections.observableArrayList(
    new Mahasiswa("Topek Ramadan", "202410370311200",
"radan@gmail.com" ),
    new Mahasiswa("Sutrisno Adit Pratama", "202410370311203",
"adit@gmail.com" ),
    new Mahasiswa("Wempy Aditya Wirawan", "202410370311233",
"radan@gmail.com" ),
    new Mahasiswa("Maylani Ku", "202410370311311",
"maylani@gmail.com" )
);
```

Langkah selanjutnya adalah mengasosiasikan data ke dalam kolom, bisa menggunakan code sebagai berikut:

```
nameCol.setCellValueFactory(  
    new PropertyValueFactory<Mahasiswa,String>("name")  
);  
nimCol.setCellValueFactory(  
    new PropertyValueFactory<Mahasiswa,String>("nim")  
);  
emailCol.setCellValueFactory(  
    new PropertyValueFactory<Mahasiswa,String>("email")  
);
```

Note : - Method `setCellValueFactory` mengimplementasikan class `PropertyValueFactory` yang menggunakan property `name`, `nim`, dan `email` sebagai referensi nilai ke class `Mahasiswa`.

Saat model data sudah ditentukan dan data ditambahkan serta sudah dikaitkan dengan kolom, kita dapat menambahkan data ke table dengan menggunakan method `setItems` pada table seperti berikut :

```
table.setItems(data);
```

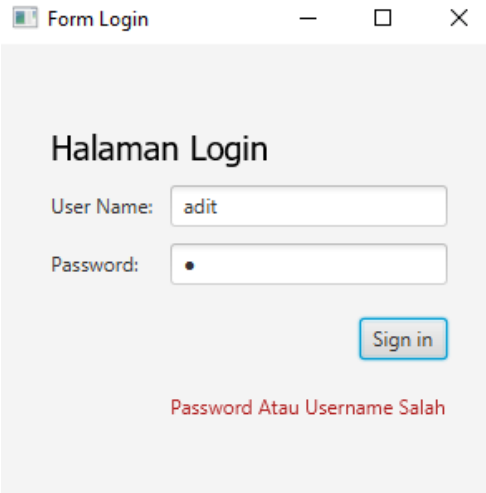
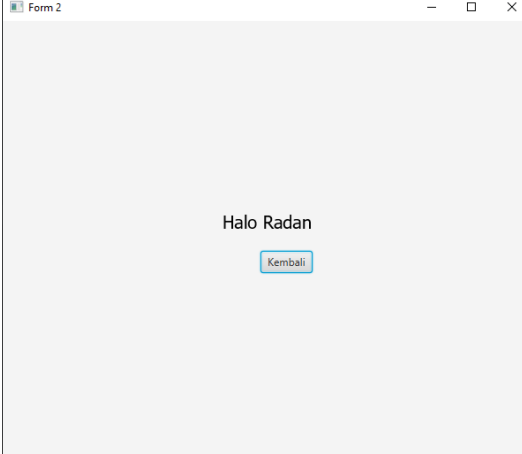
Berikut bisa dijadikan referensi tambahan untuk mempelajari javaFx :

<https://www.javatpoint.com/javafx-tutorial>

https://docs.oracle.com/javase/8/javafx/get-started-tutorial/get_start_apps.htm

CODELAB**Buatlah Program java Gui login sederhana****Ketentuan :**

1. Saat login gagal maka akan menampilkan pesan username / password salah
2. Saat login Sukses maka form akan di pindahkan ke halaman baru

Contoh**Kondisi Saat user tidak ditemukan****Kondisi saat user ditemukan**

TUGAS

Melanjutkan tugas pada tahap modul 5 buatlah aplikasi peminjaman buku tersebut menjadi GUI (GRAPHICAL USER INTERFACE)

Ketentuan :

- Diwajibkan menggunakan JavaFX
- Design GUI dibebaskan (Sesuai Kreativitas)
- Validasi validasi yang sebelumnya dibuat tetap jalan di gui

Seperti : NIM harus memiliki Panjang 15 angka dll.

- Error Handling / Exception Handling tetap di terapkan

RUBRIK PENILAIAN

Aspek Penilaian	Poin
Codelab	20
Tugas	40
Pemahaman	40
Total	100%