

VERSION 1.0
AGUSTUS 17, 2023



PEMROGRAMAN LANJUT

MODUL 6 - GUI (GRAPHICAL USER INTERFACE) USING JAVA FX

DISUSUN OLEH:

- WILDAN SUHARSO, S.KOM., M.KOM
- LARYNT SAWFA KENANGA
- AHYA NIKA SALSABILA

LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN LANJUT

PERSIAPAN MATERI

Mahasiswa harus memahami konsep OOP yang telah dipelajari sebelumnya

TUJUAN

Mahasiswa mampu membuat GUI JavaFX

TARGET MODUL

Mahasiswa mampu membuat GUI JavaFX

PERSIAPAN SOFTWARE/APLIKASI

1. Java Development Kit
2. Text Editor / IDE (**Preferably** IntelliJ IDEA, Visual Studio Code, Netbeans, etc).
3. Scene Builder

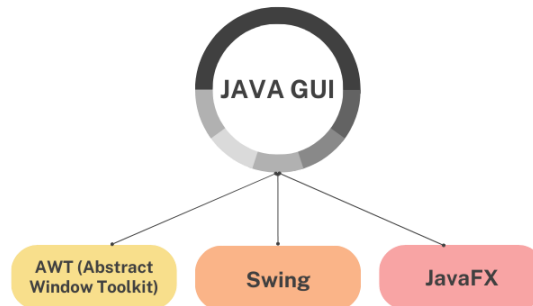
MATERI POKOK

A. KONSEP GUI

Java GUI (Graphical User Interface) adalah cara bagi pengguna untuk berinteraksi dengan perangkat lunak melalui elemen visual seperti tombol, kotak teks, gambar, dan jendela. GUI memberikan pengalaman yang lebih intuitif dan interaktif daripada antarmuka teks yang hanya menggunakan baris perintah. Keuntungan utama penggunaan GUI adalah peningkatan pengalaman pengguna dan produktivitas. Dengan menyediakan elemen-elemen seperti tombol dan kotak teks, GUI memungkinkan pengguna untuk berinteraksi secara langsung dengan perangkat lunak tanpa perlu pengetahuan mendalam tentang sintaks perintah. Interaktivitas yang ditawarkan oleh GUI memungkinkan pengguna melakukan tindakan seperti mengklik, mengetik, dan menggeser dengan mudah, sambil menerima umpan balik visual yang jelas tentang efek dari tindakan mereka.

B. JENIS PUSTAKA GUI

Terdapat beberapa pustaka GUI yang dapat digunakan, seperti AWT, Swing, dan JavaFX. Masing-masing pustaka ini memiliki ciri khas dan kelebihan tersendiri dalam hal penampilan, fleksibilitas, dan fitur.



1. AWT

AWT adalah pustaka GUI asli yang diperkenalkan bersamaan dengan Java pertama kali. AWT menyediakan antarmuka ke komponen GUI yang ditawarkan oleh sistem operasi tempat program Java dijalankan. Hal tersebut memungkinkan pengembang untuk membuat aplikasi dengan tampilan yang lebih sesuai dengan platform, tetapi memiliki keterbatasan dalam hal fungsionalitas dan tampilan yang modern. Berikut adalah kelebihan dan kekurangan AWT:

a. Kelebihan

- Penampilan natif pada platform.
- Kinerja yang relatif baik karena menggunakan komponen sistem operasi.

b. Kekurangan

- Terbatas dalam fitur dan tampilan modern.
- Tidak konsisten di seluruh platform.
- Tidak mendukung komponen kustom yang fleksibel.

2. Swing

Swing muncul sebagai jawaban atas keterbatasan AWT. Diperkenalkan pada Java 1.2, Swing menyediakan pustaka GUI yang ditulis sepenuhnya dalam Java. Hal ini berarti komponen Swing tidak tergantung pada komponen GUI native sistem operasi, sehingga dapat menawarkan penampilan yang konsisten di berbagai platform. Swing menghadirkan antarmuka yang lebih kaya, komponen yang lebih fleksibel, dan dukungan untuk tampilan kustom. Berikut adalah kelebihan dan kekurangan Swing:

a. Kelebihan

- Penampilan yang konsisten di seluruh platform.

- Komponen yang fleksibel dan bervariasi.
- Dukungan untuk tampilan kustom.

b. Kekurangan

- Performa kurang optimal karena tidak menggunakan komponen GUI native.
- Beberapa komponen dapat terasa kurang responsif.

3. JavaFX

JavaFX diperkenalkan oleh Sun Microsystems (kemudian diakuisisi oleh Oracle) sebagai teknologi modern yang berfokus pada pengembangan antarmuka pengguna yang kaya dan interaktif. JavaFX diluncurkan sebagai alternatif yang lebih canggih untuk Swing, dengan dukungan untuk efek visual, animasi, tata letak yang lebih baik, integrasi dengan CSS, dan dukungan multimedia. JavaFX juga lebih dioptimalkan untuk pengembangan aplikasi seluler dan web. Berikut adalah kelebihan dan kekurangan JavaFX:

a. Kelebihan

- Tampilan yang kaya dengan efek visual dan animasi.
- Tata letak yang lebih fleksibel.
- Dukungan multimedia yang kuat.
- Dukungan integrasi dengan CSS.
- Lebih dioptimalkan untuk pengembangan aplikasi seluler dan web.

b. Kekurangan

- Kurang dikenal dan kurang umum digunakan dibandingkan Swing.
- Dalam beberapa kasus, mungkin memerlukan waktu lebih lama untuk mempelajari dan mengembangkan aplikasi.

C. JAVA FX TUTORIAL

1. Istilah-Istilah dalam JavaFX

Sebelum memulai belajar JavaFX, alangkah baiknya kalian mengetahui istilah-istilah di bawah ini terlebih dahulu.

- Scene: Mewakili konten yang dapat dilihat dan diinteraksikan dalam jendela aplikasi. Scene adalah layar di aplikasi, seperti halaman dalam buku cerita. Di sini Anda menaruh tombol, teks, gambar, dan hal lain yang bisa Anda lihat dan gunakan.
- Stage: Jendela utama aplikasi JavaFX. Setiap aplikasi JavaFX memiliki setidaknya satu stage di mana scene-scene dapat ditampilkan. Ilustrasikan stage sebagai bingkai utama aplikasi.

- c. Node: Dasar dari hirarki elemen-elemen UI dalam JavaFX. Semua elemen visual seperti tombol, teks, gambar, dan layout adalah jenis Node.
- d. Control: Control adalah subkelas dari Node yang mewakili elemen-elemen antarmuka pengguna yang dapat diinteraksikan seperti tombol, kotak teks, dan lainnya.
- e. Dialog: Jendela kecil yang muncul untuk berinteraksi dengan pengguna dalam konteks tertentu. Dialog seperti kotak percakapan. Biasanya muncul ketika Anda perlu memberi tahu pengguna sesuatu atau mendapatkan input dari mereka.
- f. EventHandler: Objek yang menangani peristiwa (event) yang terjadi dalam antarmuka pengguna, seperti mengklik tombol atau memasukkan teks.
- g. CSS Styling: JavaFX mendukung gaya dan tata letak menggunakan CSS (Cascading Style Sheets) yang memungkinkan untuk menyesuaikan tampilan elemen-elemen UI.

2. JavaFX Layouts

Tata letak (Layouts) mendefinisikan gaya antarmuka pengguna (UI) untuk objek-objek grafik tampilan. JavaFX menyediakan berbagai jenis panel tata letak yang mendukung gaya tata letak yang berbeda. Semua kelas ini termasuk dalam paket **javafx.scene.layout**. Kelas `javafx.scene.layout.Pane` adalah kelas dasar untuk semua kelas tata letak bawaan dalam JavaFX. Berikut adalah beberapa jenis layout classes:

Layout	Deskripsi
BorderPane	Mengatur node dalam bagian atas, kiri, kanan, tengah, dan bawah layar.
FlowPane	Mengatur node dalam baris horizontal sesuai dengan ruang horizontal yang tersedia. Meliputi node ke baris berikutnya jika ruang horizontal kurang dari lebar total node.
GridPane	Mengatur node dalam bentuk baris dan kolom.
HBox	Mengatur node dalam satu baris tunggal.
Pane	Kelas dasar untuk semua kelas tata letak.
StackPane	Mengatur node dalam bentuk tumpukan, satu di atas yang lain.
VBox	Mengatur node dalam kolom vertikal.

Contoh: Layout HBox

Berikut adalah contoh GUI yang menggunakan layout HBox (tanpa menggunakan file FXML). Contoh di bawah ini menggunakan HBox (Horizontal Box) sebagai tata letak utama untuk

mengatur posisi dan tampilan tombol-tombol di dalam antarmuka pengguna. Seperti yang terlihat, terdapat dua button yang diletakkan di atas layout. Nah karena pada program ini menggunakan layout HBox, maka susunan button pun terseusun berjejer secara horizontal.

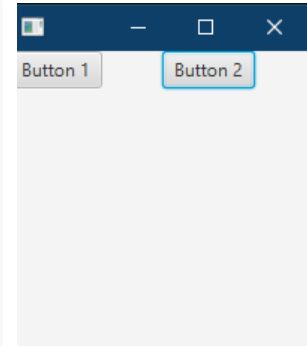
```
package com.example.hellofx;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;

public class HelloApplication extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception {
        Button btn1 = new Button("Button 1");
        Button btn2 = new Button("Button 2");
        HBox root = new HBox();
        Scene scene = new Scene(root, 200, 200);
        root.getChildren().addAll(btn1, btn2);
        root.setSpacing(40);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

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



3. JavaFX 2D Shapes

Setiap bentuk direpresentasikan oleh kelas tertentu dari paket **javafx.scene.shape**. Untuk membuat bentuk dua dimensi, instruksi berikut perlu diikuti.

- Menginstansiasi kelas yang sesuai: misalnya, `Rectangle rect = new Rectangle()`
- Mengatur properti yang diperlukan untuk kelas tersebut menggunakan metode setter instance: misalnya,

```
rect.setX(10);
rect.setY(20);
rect.setWidth(100);
rect.setHeight(100);
```

- Menambahkan objek kelas ke layout Group: misalnya,

```
Group root = new Group();
root.getChildren().add(rect);
```

Jenis-jenis kelas JavaFX shape selengkapnya kalian bisa pelajari lebih lanjut di link yang tersedia pada referensi.

4. JavaFX Text

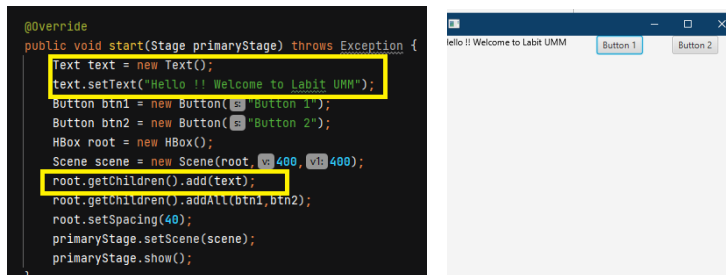
Dalam beberapa kasus, kita perlu menyediakan informasi berbasis teks pada antarmuka aplikasi kita. Pustaka JavaFX menyediakan kelas bernama **javafx.scene.text.Text** untuk tujuan ini. Kelas ini menyediakan berbagai metode untuk mengubah berbagai properti teks yang selengkapnya kalian bisa pelajari lebih lanjut di link yang tersedia pada referensi.

a. Membuat Teks

Gunakan metode setter `setText(string)` untuk mengatur string sebagai teks untuk objek kelas teks. Ikuti sintaks di bawah ini untuk menginstansiasi kelas Text.

```
Text <text_Object> = new Text();
text.setText(<string-teks>);
```

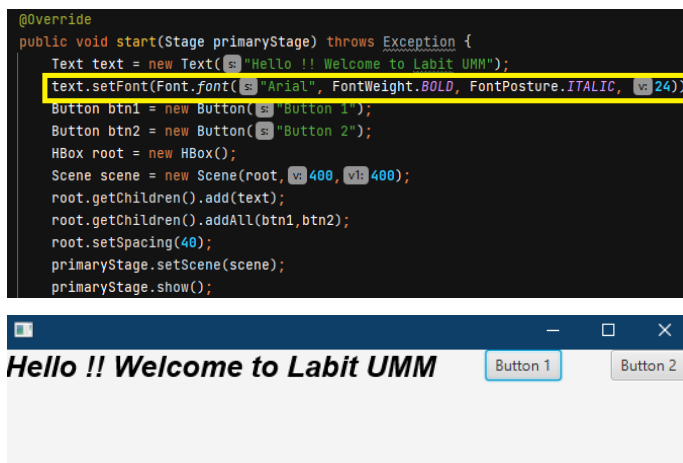
Contoh:



b. Mengatur Font dan Posisi Teks

Kita perlu mengatur properti font kelas Text dengan menggunakan metode setter `setFont()` yang menerima objek kelas Font. Metode `Font.font()` menerima parameter Family, Weight, Posture, dan Size.

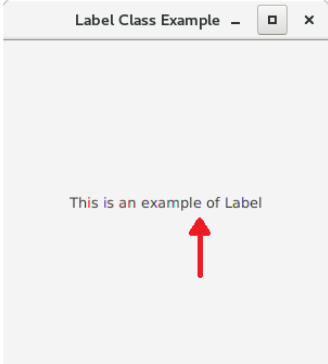



Contoh:

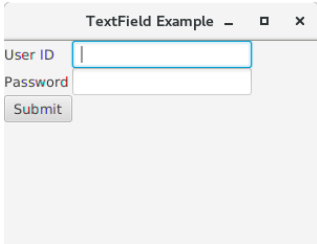
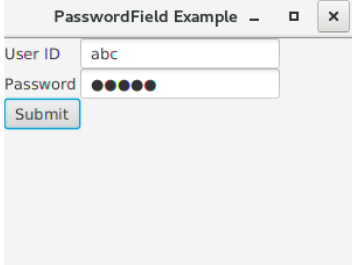
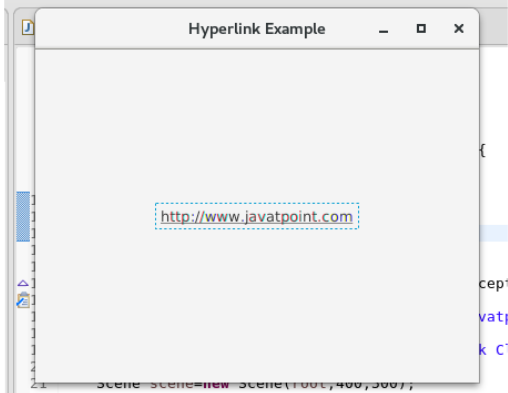
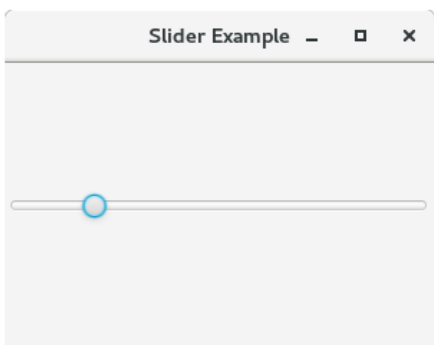


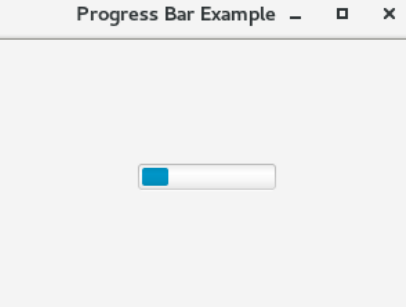
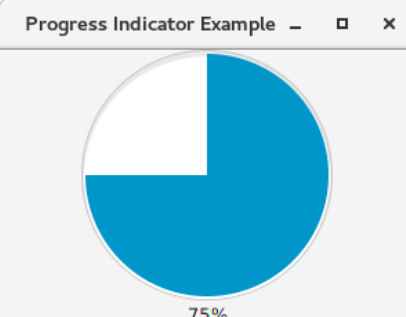

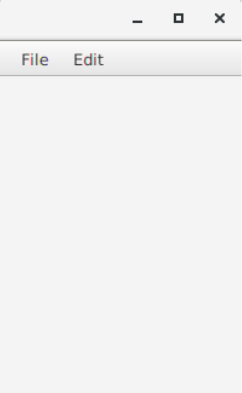
Penerapan dekorasi lain pada teks seperti warna bisa dipelajari lebih lanjut di link yang tersedia pada referensi.

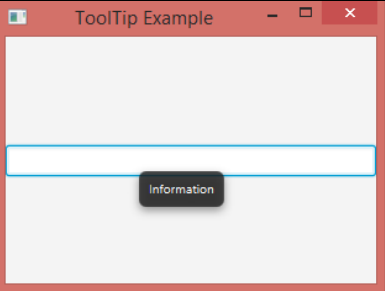
5. JavaFX UI

Elemen UI adalah elemen-elemen yang sebenarnya ditampilkan kepada pengguna untuk interaksi atau pertukaran informasi. Paket **javafx.scene.control** menyediakan semua kelas yang diperlukan untuk komponen-komponen UI seperti Button, Label, dll. Berikut adalah Daftar Kontrol UI JavaFX:

<p>Label</p> 	<p>Komponen yang digunakan untuk mendefinisikan teks sederhana di layar.</p>
<p>Button</p> 	<p>Button adalah komponen yang mengendalikan fungsi aplikasi. Kelas Button digunakan untuk membuat tombol berlabel.</p>
<p>RadioButton</p> 	<p>Digunakan untuk memberikan berbagai opsi kepada pengguna. Pengguna hanya dapat memilih satu opsi di antara semuanya. Radio button bisa dipilih atau tidak dipilih.</p>
<p>CheckBox</p> 	<p>Digunakan untuk mendapatkan jenis informasi dari pengguna yang berisi berbagai pilihan. Pengguna menandai kotak centang menjadi (benar) atau tidak (salah).</p>

<p>TextField</p> 	<p>untuk mendapatkan masukan dari pengguna dalam bentuk teks.</p>
<p>PasswordField</p> 	<p>Digunakan untuk mendapatkan kata sandi pengguna. Apapun yang diketik di dalam passwordfield tidak ditampilkan di layar kepada siapa pun.</p>
<p>HyperLink</p> 	<p>Digunakan untuk merujuk ke salah satu halaman web melalui aplikasi Anda.</p>
<p>Slider</p> 	<p>digunakan untuk menyediakan jajaran pilihan kepada pengguna dalam bentuk grafis di mana pengguna harus menggeser slider di seluruh rentang nilai untuk memilih salah satu dari mereka.</p>
<p>ProgressBar</p>	<p>sdigunakan untuk menunjukkan kemajuan pekerjaan kepada pengguna.</p>

 <p>A screenshot of a JavaFX window titled "Progress Bar Example". It contains a horizontal progress bar that is approximately 25% filled with blue.</p>	
<p>ProgressIndicator</p>  <p>A screenshot of a JavaFX window titled "Progress Indicator Example". It contains a circular progress indicator (gauge) that is 75% filled with blue. The text "75%" is displayed below the gauge.</p>	<p>Daripada menunjukkan kemajuan analog kepada pengguna, elemen ini menampilkan kemajuan digital sehingga pengguna dapat mengetahui jumlah pekerjaan yang selesai dalam persentase.</p>
<p>ScrollBar</p>  <p>A screenshot of a JavaFX window titled "ScrollBar Example". It contains a vertical scrollbar with a slider in the middle.</p>	<p>digunakan untuk menyediakan bilah gulir kepada pengguna sehingga pengguna dapat menggulir halaman aplikasi.</p>
<p>Menu</p>  <p>A screenshot of a JavaFX window showing a menu bar with two menu items: "File" and "Edit".</p>	<p>JavaFX menyediakan kelas Menu untuk mengimplementasikan menu. Menu adalah komponen utama dari setiap aplikasi.</p>
<p>ToolTip</p>	<p>ToolTip JavaFX digunakan untuk memberikan petunjuk kepada pengguna tentang komponen apa pun. Biasanya digunakan untuk</p>

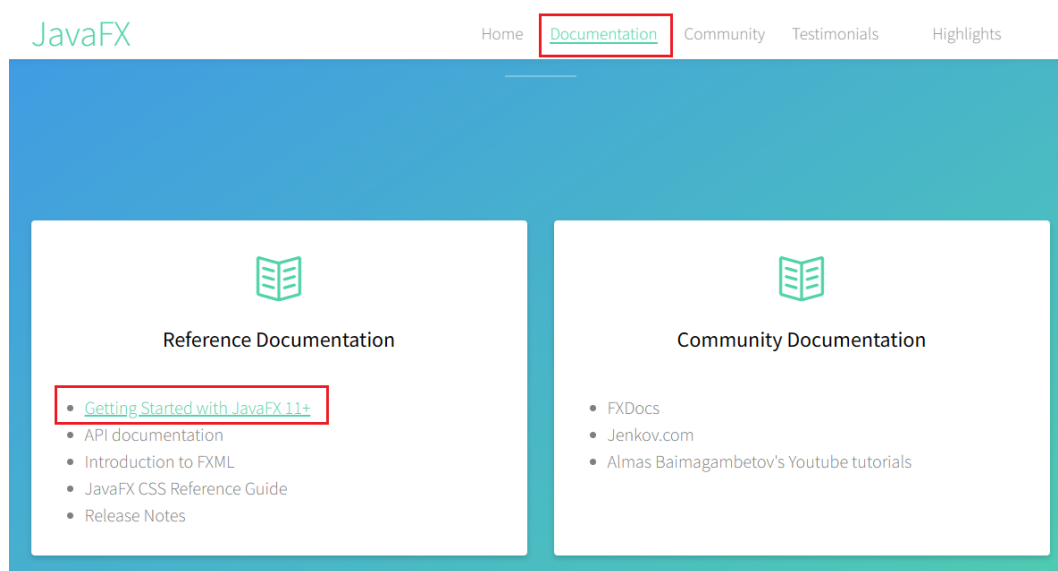
	<p>memberikan petunjuk tentang bidang teks atau bidang kata sandi yang digunakan dalam aplikasi.</p>
---	--

MATERI PRATIKUM

A. INSTALASI DAN KONFIGURASI JAVA FX

1. Download JavaFX SDK

- Bukalah openjfx.io. Kemudian pada bagian **documentation**, klik **getting started with JavaFX**.



- Buka bagian **JavaFX and IntelliJ** jika kamu menggunakan IDE IntelliJ. Kemudian klik link **JavaFX SDK** seperti gambar di bawah ini.

JavaFX

Introduction

Install Java

Run HelloWorld using JavaFX

Run HelloWorld via Maven

Run HelloWorld via Gradle

Runtime images

JavaFX and IntelliJ

Non-modular from IDE

Non-modular with Maven

Non-modular with Gradle

Modular from IDE

Modular with Maven

Modular with Gradle

JavaFX and NetBeans

JavaFX and Eclipse

JavaFX and Visual Studio Code

Next Steps

This section explains how to create a JavaFX application in IntelliJ IDEA. JavaFX 15.0.1 and IntelliJ IDEA 2020.1 were used for the IDE screenshots.

Download an appropriate JDK for your operating system and set `JAVA_HOME` to the JDK directory. Refer to [Install Java](#) section for more information.

You can create a JavaFX modular or non-modular project and use the IDE tools, Maven or Gradle build tools.

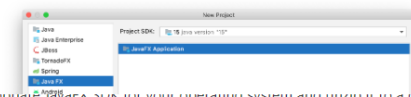
Non-modular projects

IDE

Follow these steps to create a JavaFX non-modular project and use the IDE tools to build it and run it. Alternatively, you can download a similar project from [here](#).

Download the appropriate [JavaFX SDK](#) for your operating system and unzip it to a desired location, for instance `/Users/your-user/Downloads/javafx-sdk-20`.

1. Create a JavaFX project



- Setelah itu halaman beralih ke JavaFX oleh [Gluon](#). Scroll ke bawah dan download JavaFX yang sesuai dengan perangkatmu.

GLUON Products - Developers Pricing Services Insights - Contact -

Downloads

JavaFX version: 20.0.2 Operating System: [any] Architecture: [any] Type: [any]

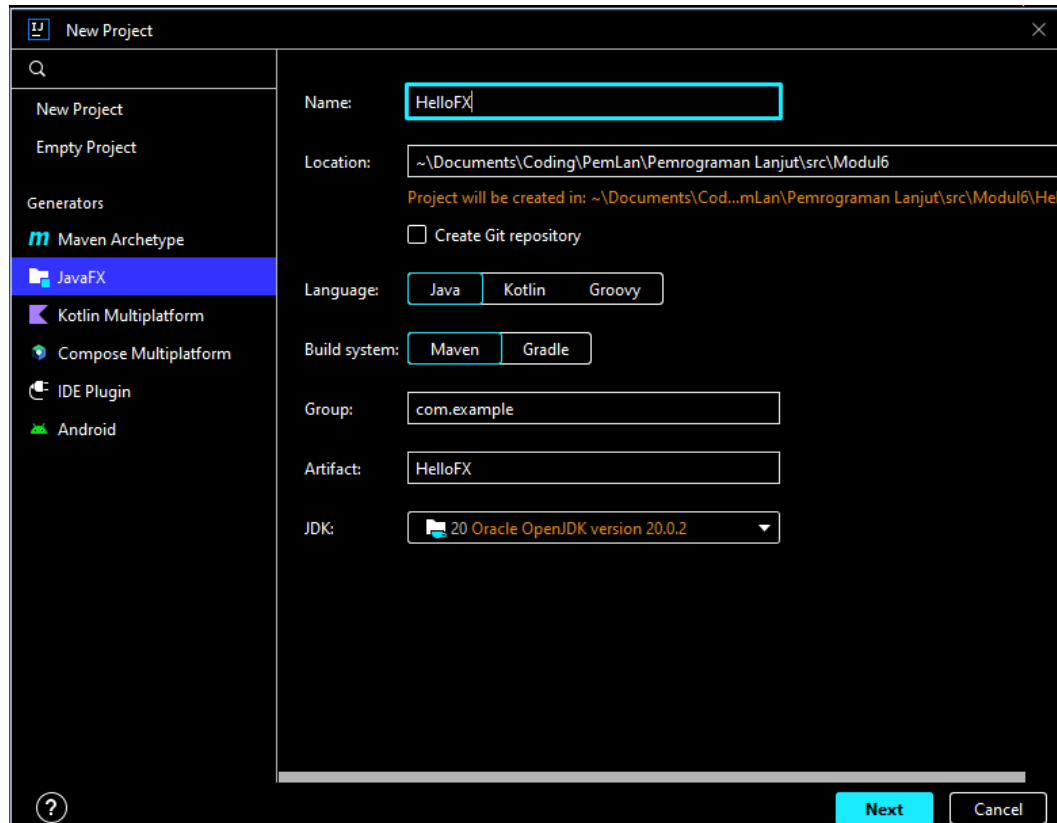
☐ Include older versions

OS	Version	Architecture	Type	Download
Linux	20.0.2	aarch64	SDK	Download (SHA256)
Linux	20.0.2	aarch64	jmods	Download (SHA256)
Linux	20.0.2	aarch64	Monocle SDK	Download (SHA256)
Linux	20.0.2	arm32	SDK	Download (SHA256)
Linux	20.0.2	x64	SDK	Download (SHA256)
Linux	20.0.2	x64	jmods	Download (SHA256)
macOS	20.0.2	aarch64	SDK	Download (SHA256)
macOS	20.0.2	aarch64	jmods	Download (SHA256)
macOS	20.0.2	aarch64	Monocle SDK	Download (SHA256)
macOS	20.0.2	x64	SDK	Download (SHA256)
macOS	20.0.2	x64	jmods	Download (SHA256)
macOS	20.0.2	x64	Monocle SDK	Download (SHA256)
Windows	20.0.2	x64	SDK	Download (SHA256)
Windows	20.0.2	x64	jmods	Download (SHA256)

- Ekstrak file hasil download ke lokasi penyimpanan yang kamu inginkan. Jangan sampai lupa lokasi penyimpanan JavaFX ini karena nanti lokasi penyimpanan tersebut diperlukan dalam set up di IDE.

2. Create JavaFX Project

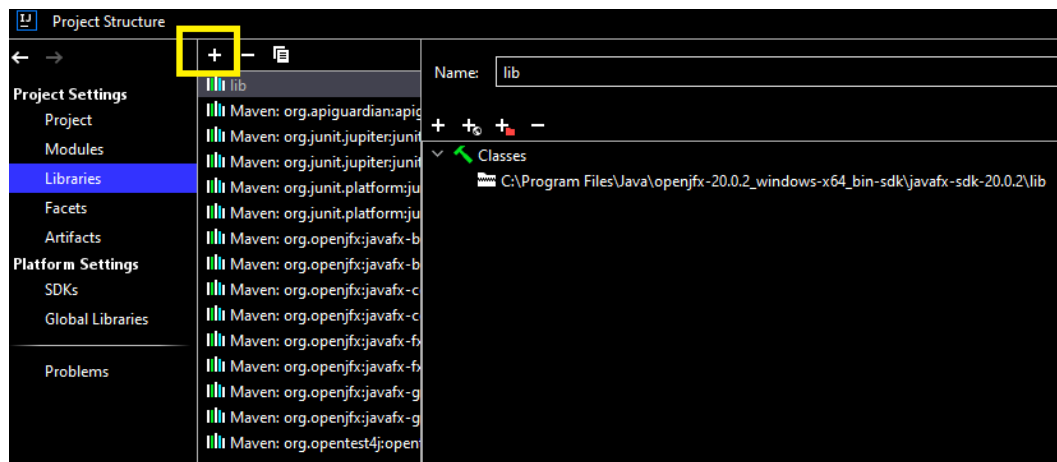
- Buat project JavaFX



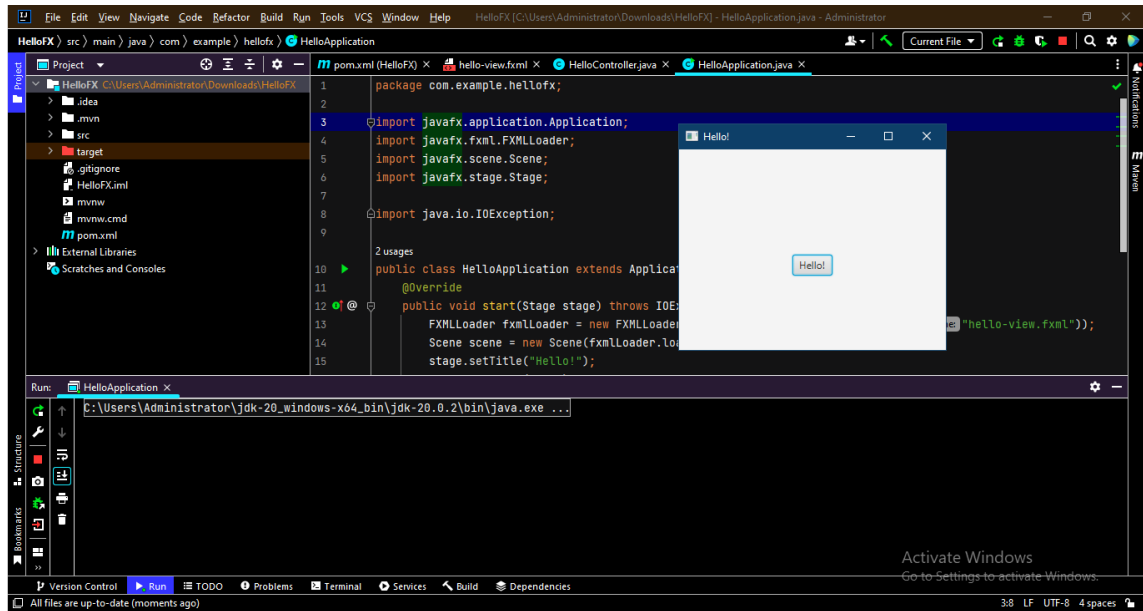
Klik next, lalu langsung create tanpa centang apapun di halaman selanjutnya.

3. Add SDK Library

- Bukalah File → Project Structure → Libraries. Tambahkan project library baru seperti pada bagian yang ditandai di gambar bawah ini. Kemudian select lokasi library file dimana tempat menyimpan file lib SDK JavaFX.



4. Jalankan program, maka akan menampilkan GUI seperti di bawah ini.



B. SCENE BUILDER

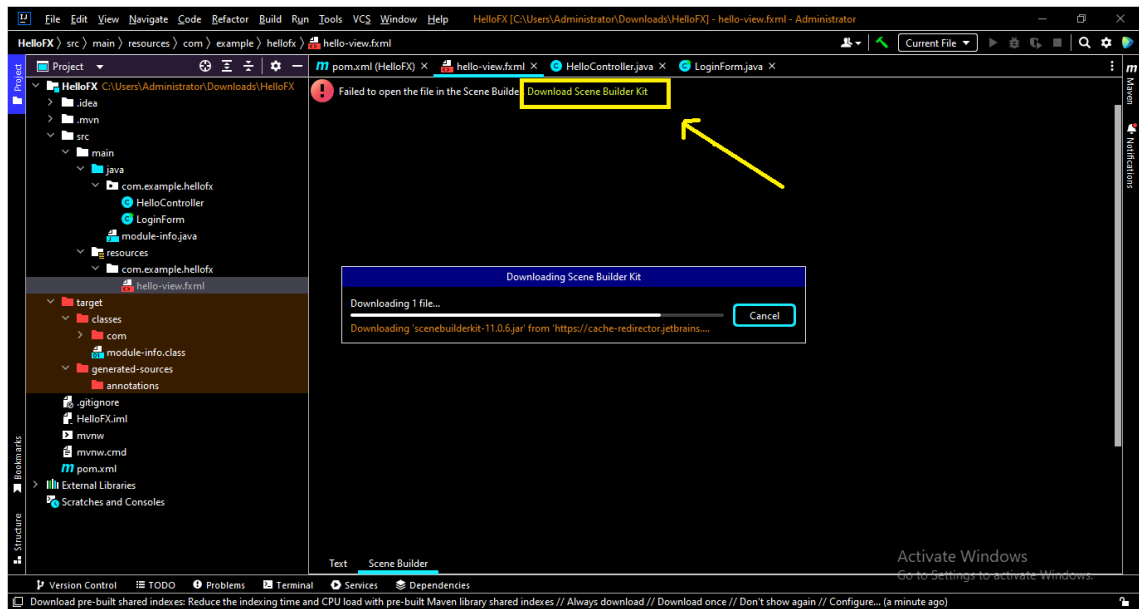
Scene Builder adalah alat yang memungkinkan kita untuk merancang antarmuka pengguna (UI) untuk aplikasi JavaFX secara visual. Dengan menggunakan JavaFX Scene Builder, kita dapat membuat UI dengan cara drag and drop elemen-elemen grafis seperti tombol, label, bidang teks, gambar, dan sebagainya. Hal Ini memudahkan pengembang untuk merancang tata letak dan tampilan aplikasi dengan cepat dan visual.

File FXML (FXML stands for "JavaFX Markup Language") adalah format file yang digunakan untuk mendefinisikan antarmuka pengguna (UI) dalam aplikasi JavaFX. Ketika anda menyusun tampilan UI dalam Scene Builder, ia akan menghasilkan file FXML yang sesuai yang berisi deskripsi tata letak dan properti elemen-elemen UI yang telah anda tambahkan.

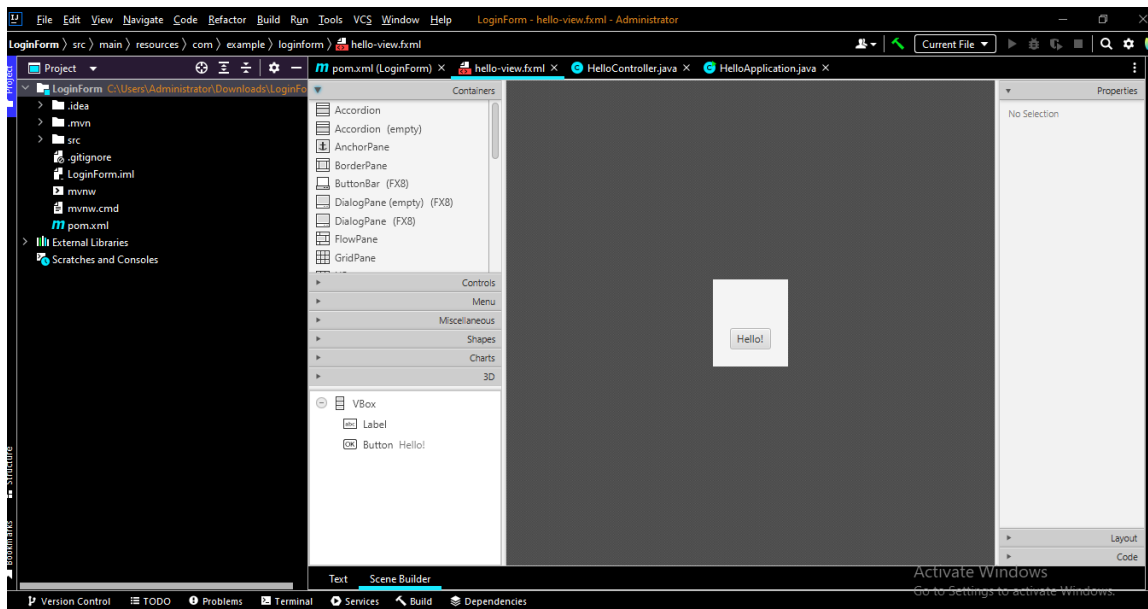
C. CARA PENGGUNAAN SCENE BUILDER

1. Cara 1: Menggunakan Scene Builder Kit pada IDE

Scene Builder Kit adalah API yang memungkinkan mengintegrasikan panel Scene Builder dan fungsionalitasnya langsung ke GUI dari IDE Java, seperti NetBeans, IntelliJ, dan Eclipse. Cara menggunakan Scene Builder Kit yaitu dengan membuka terlebih dahulu file fxml dari project anda. Kemudian download scene builder kit seperti gambar di bawah ini.

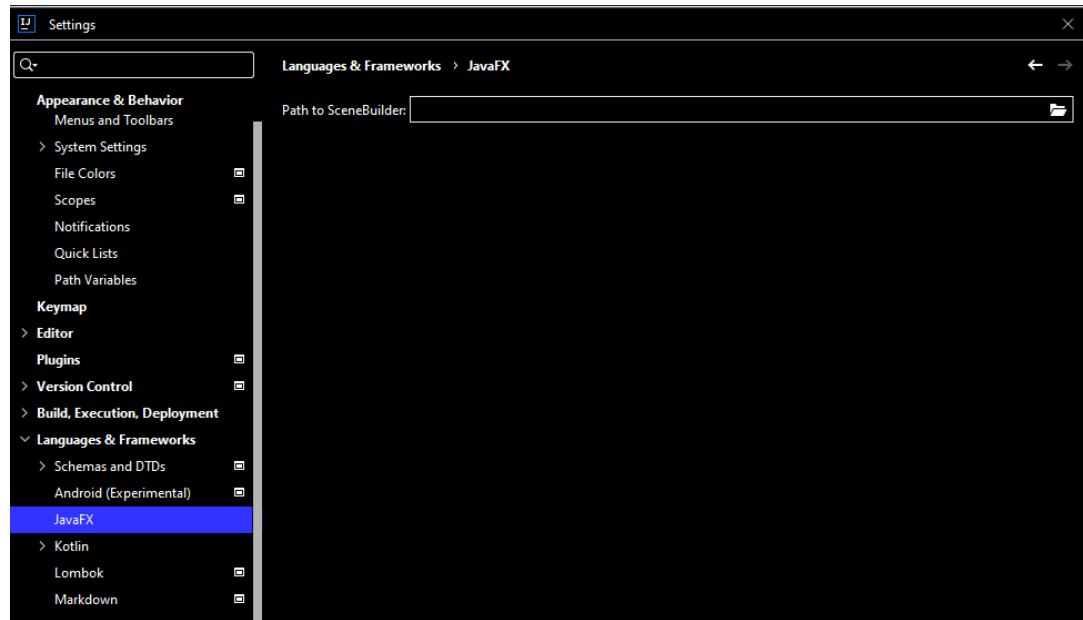


Setelah selesai, maka anda bisa membuat antarmuka pengguna dengan drag and drop untuk mengedit dan memanipulasi file FXML, dan mengotomatiskan tugas-tugas desain GUI yang berulang.

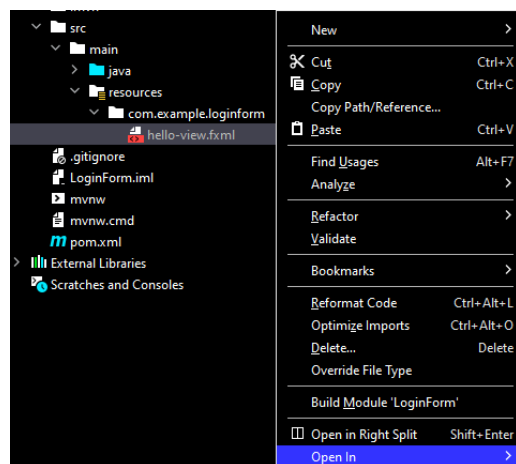


2. Cara 2: Menggunakan Aplikasi SceneBuilder.exe

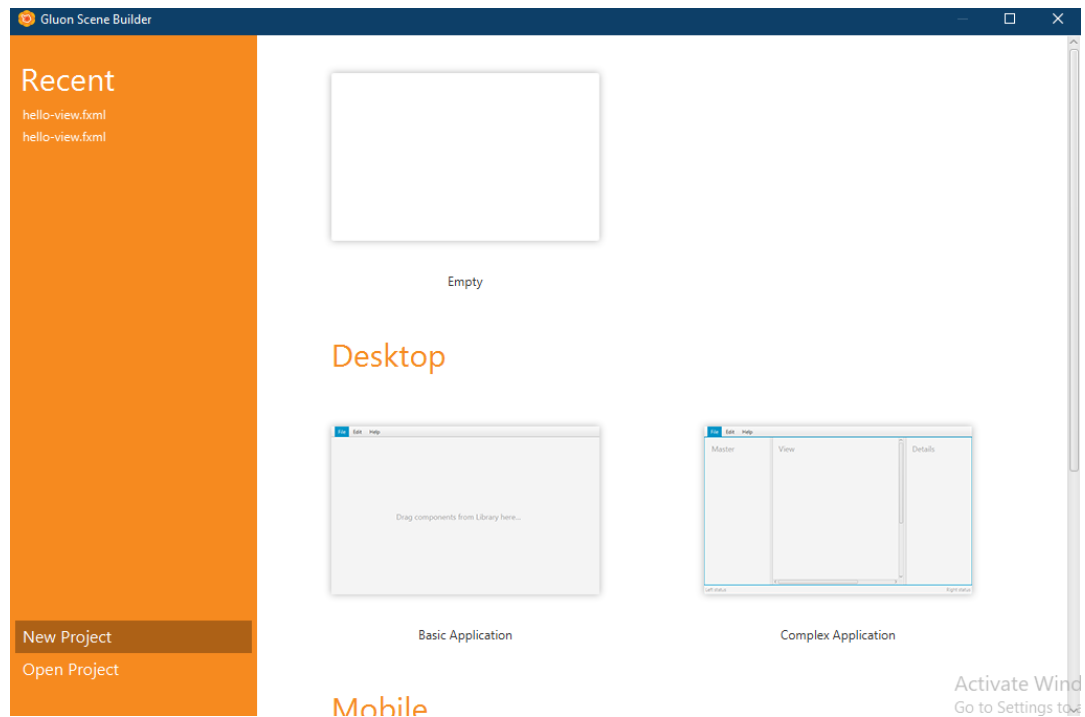
- Download scene builder dari laman <https://gluonhq.com/products/scene-builder/>
- Install scene builder pada perangkat anda.
- Setelah selesai, buka IDE ke bagian File → Settings → Languages & Frameworks → JavaFX. Kemudian isilah path/letak dari scenebuilder.exe. Setelah itu, apply dan oke.



- Open file fxml dengan cara klik kanan file fxml kemudian **open in scene builder**.



- Atau kalian juga bisa membuka file fxml dari aplikasi SceneBuilder.exe dengan cara open project file fxml seperti gambar di bawah ini.



LATIHAN PRATIUM

LATIHAN PRATIUM 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 di bawah ini

```
@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);

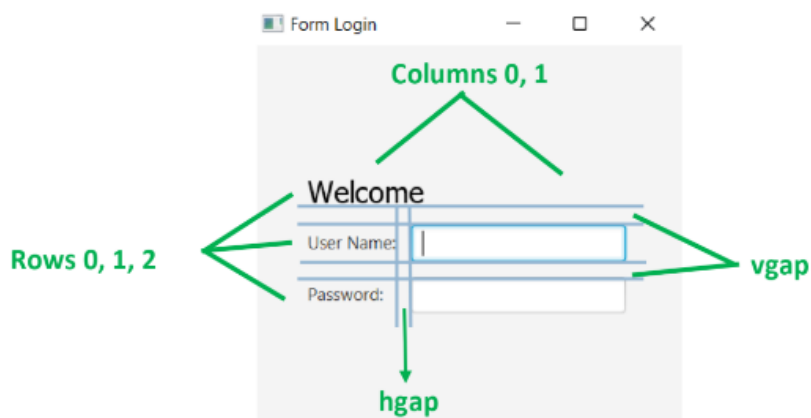
TextField userTextField = new TextField();
grid.add(userTextField, 1, 1);

Label pw = new Label ("Password:");
grid.add(pw, 0, 2);

PasswordField pwBox = new PasswordField();
grid.add(pwBox, 1, 2);

```

4. Tampilan akan Nampak seperti gambar di bawah ini



5. Kemudian tambahkan button dan text untuk menampilkan pesan dengan menambahkan script di bawah ini

```

Button btn = new Button("Sign in");
HBox hbBtn = new HBox (10);
hbBtn.setAlignment (Pos. BOTTOM_RIGHT);
hbBtn.getChildren().add(btn);
grid.add(hbBtn, 1, 4);

```

Tambahkan Text control untuk mendisplay message.

```

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

```

6. Selanjutnya tambahkan 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");
    }

});

```

LATIHAN PRATIUM 2

Membuat Table JavaFx

- Membuat tampilan awal table

```

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() ;
    }
}

```

- Menambahkan data pada table

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);
    }
}

```

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

```

final ObservableList<Mahasiswa> data = FXCollections.observableArrayList (
    new Mahasiswa("Larynt", "202110370311189", "laryntsa@gmail.com"),
    new Mahasiswa("Ahya", "202110370311187", "ayaa@gmail.com")
);

```

Langkah selanjutnya adalah mengasosiasikan data ke dalam colom.

```

nameCol.setCellValueFactory(
    new PropertyValueFactory<Mahasiswa,String>("name")
);

nimCol.setCellValueFactory(
    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);
```

Lalu jalankan dan lihatlah hasil dari data yang anda buat sudah masuk ke dalam table.

- Menambahkan baris baru

Untuk menambahkan nilai baru pada kolom Nama, NIM, dan Email dengan menggunakan inputan dari user dapat menggunakan **TextField**. Buatlah tiga **TextField** untuk tiap kolomnya dan tambahkan sebuah **Add button** untuk menambahkan datanya.

```

final TextField addName = new TextField();
addName.setMaxWidth(nameCol.getPrefWidth());
addName.setPromptText("Nama Mahasiswa");

final TextField addNim = new TextField();
addNim.setMaxWidth(nimCol.getPrefWidth());
addNim.setPromptText("NIM");

final TextField addEmail = new TextField();
addEmail.setMaxWidth(emailCol.getPrefWidth());
addEmail.setPromptText("Email");

```

```
final Button addButton = new Button("Add");
addButton.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent e) {
        data.add(new Mahasiswa(
            addName.getText(),
            addNim.getText(),
            addEmail.getText()
        ));
        addName.clear();
        addNim.clear();
        addEmail.clear();
    }
});
```

Note : `addName.clear()`, `addNim.clear()`, `addEmail().clear` digunakan ketika mengklik Add button. Data sebelumnya yang ada pada TextField akan terhapus.

Setelah itu tambahkan Hbox sebagai layout secara horizontal untuk TextField dan button nya.

```
final HBox = hboxInput = new HBox();
hboxInput.getChildren().addAll(addName, addNim, addEmail, addButton);
hboxInput.setSpacing(10);
```

Setelah itu tambahkan variable hboxinput tadi ke dalam variable vbox sebelumnya agar posisi inputan berada pada bawah table.

```
vboxInput.getChildren().addAll(label, table, hboxInput);
```

Lalu coba jalankan dan coba masukan inputannya.

REFERENSI

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

https://youtube.com/playlist?list=PLZPZq0r_RZOM-8vJA3NQFZB7JroDcMwev

TUGAS

TUGAS 1

Buatlah form biodata mahasiswa dengan field: **Nama, NIM, Email, Fakultas, Jurusan, Alamat, Kota, dan button Create**. Terapkan table untuk menampung data hasil inputan dari user. Ketika button

create, maka muncul scene baru yang berisi data yang telah diinputkan user (**Create dan Read**).

Ketentuan inputan sebagai berikut:

- Inputan field tidak boleh ada yang kosong
- Inputan NIM harus berupa angka
- Inputan email harus diakhiri dengan **@webmail.umm.ac.id**

Apabila ada inputan yang tidak sesuai maka akan menampilkan alert yang memperingatkan pengguna untuk memasukkan inputan yang benar.

TUGAS 2

Buatlah aplikasi jadwal kuliah dengan field: **nama dosen, mata kuliah, GKB, waktu, dan ruangan** dengan catatan user dapat melakukan CRUD data pada tiap jadwal matkul. Tampung data inputan tersebut ke dalam sebuah table. Ketentuan inputan adalah sebagai berikut:

- Inputan field tidak boleh ada yang kosong
- Inputan NIM harus berupa angka

Apabila ada inputan yang tidak sesuai maka akan menampilkan alert yang memperingatkan pengguna untuk memasukkan inputan yang benar.

KRITERIA & DETAIL PENILAIAN

Kriteria Penilaian		Nilai
Latihan 1		15
Telah mengimplementasikan program sesuai ketentuan soal	10	
Program tidak error	5	
Latihan 2		15
Telah mengimplementasikan program sesuai ketentuan soal	10	
Program tidak error	5	
Tugas 1		30
Telah mengimplementasikan program sesuai ketentuan soal	20	
Program tidak error	10	
Tugas 2		30
Telah mengimplementasikan program sesuai ketentuan soal	20	

Program Tidak Error	10	
Pemahaman		10
Total		100