



# Modul Praktikum Pemrograman Berorientasi Objek

Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam

**Universitas Lampung**

**2022**  
Semester Ganjil

## Modul 14. JavaFX : Component

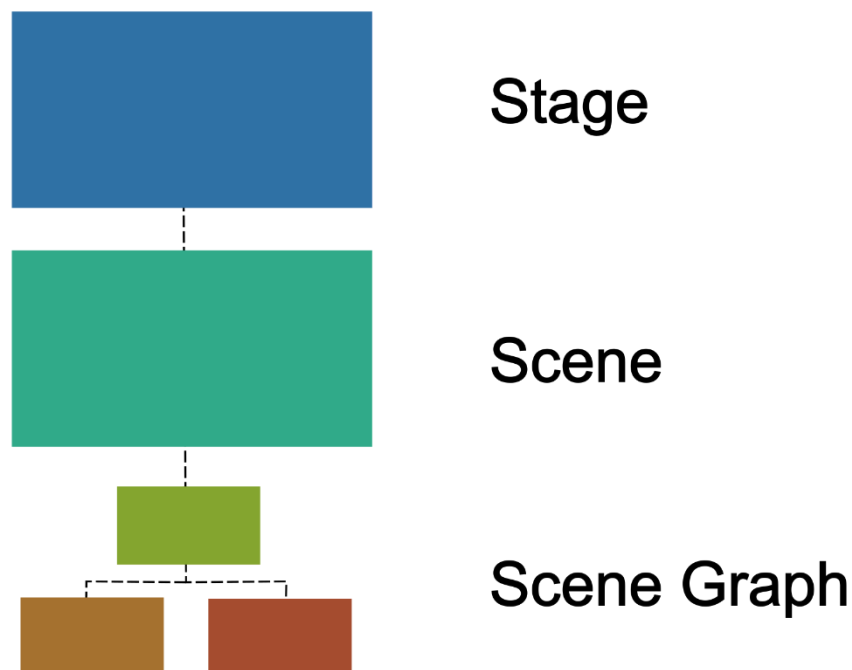
JavaFX adalah library Java yang digunakan untuk membangun Rich Internet Applications. Aplikasi yang dikembangkan menggunakan JavaFX dapat berjalan di berbagai perangkat seperti Komputer Desktop, Mobile Phones, TV, Tablet, dll.

Secara umum, aplikasi JavaFX berisi satu atau lebih tahapan yang sesuai dengan windows. Setiap Stage memiliki Scene yang melekat padanya. Setiap Scene dapat memiliki grafik dan objek kontrol, tata letak, dll.

### Capaian Pembelajaran

1. Mahasiswa mampu membuat program java yang menggunakan javaFX
2. Mahasiswa mampu membuat *User Interface* sederhana menggunakan JavaFX
3. Mahasiswa mampu membuat User Interface sederhana menggunakan *SceneBuilder*

### Materi



#### Stage

Stage adalah bingkai paling luar untuk aplikasi JavaFX. Stage Biasanya sesuai dengan Windows (jendela).

## Scene

Untuk menampilkan apapun di atas Stage dalam aplikasi JavaFX, Anda memerlukan Scene. Stage hanya dapat menampilkan satu adegan pada satu waktu, tapi dimungkinkan untuk bertukar adegan pada saat runtime. Sama seperti panggung teater yang dapat diatur ulang untuk menampilkan beberapa adegan selama pertunjukan, objek Stage di JavaFX dapat menampilkan beberapa Scene (satu per satu) selama aplikasi JavaFX dijalankan.

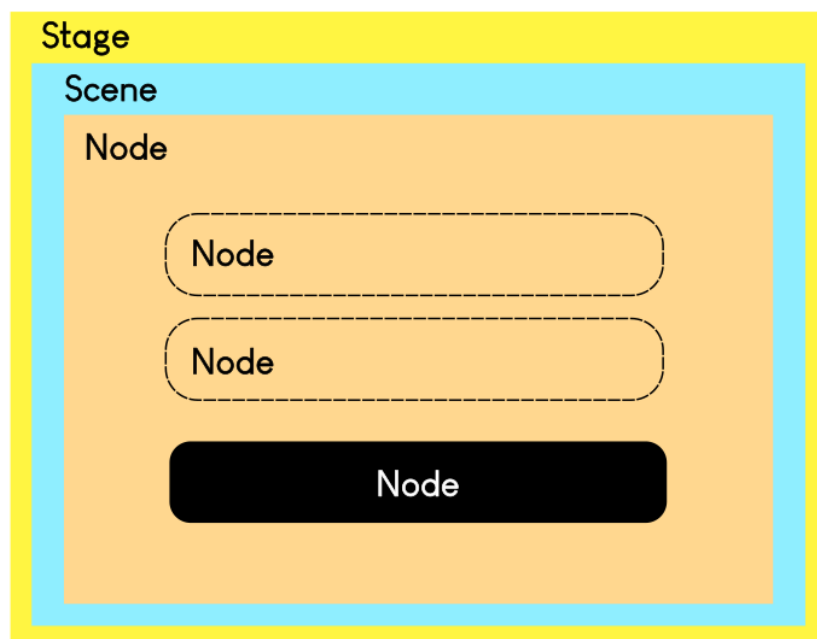
## Scene graph

Semua komponen visual (kontrol, tata letak, dll.) harus dilampirkan ke adegan yang akan ditampilkan, dan adegan itu harus dilampirkan ke panggung agar seluruh adegan terlihat. Grafik objek total dari semua kontrol, tata letak, dll. yang dilampirkan ke adegan disebut grafik adegan.

## Node

Semua komponen yang melekat pada grafik Scene disebut node, Semua node adalah subclass dari kelas JavaFX yang disebut `javafx.scene.Node`.

Ada dua jenis node: node cabang dan node daun. Sebuah node cabang adalah node yang dapat berisi node lain (node anak). Node cabang juga disebut sebagai node induk karena dapat berisi node anak. Sebuah node daun adalah node yang tidak dapat berisi node lain.



## Controll

Control JavaFX adalah komponen JavaFX yang menyediakan semacam fungsi kontrol di dalam aplikasi JavaFX. Misalnya, tombol, tombol radio, tabel, pohon, dll. Agar kontrol terlihat, kontrol harus sematkan ke grafik Scene dari beberapa objek Scene.

Kontrol biasanya bersarang di dalam beberapa komponen tata letak JavaFX yang mengelola tata letak kontrol relatif satu sama lain.

JavaFX berisi kontrol berikut:

Controi	Description
Label	Label adalah komponen untuk menempatkan teks.
Button	Class ini membuat tombol berlabel.
ColorPicker	ColorPicker menyediakan panel kontrol yang dirancang untuk memungkinkan pengguna memanipulasi dan memilih warna.
CheckBox	CheckBox adalah komponen yang dapat dalam keadaan aktif (benar) atau tidak aktif (salah).
RadioButton	RadioButton adalah komponen grafis, yang bisa dalam keadaan ON (benar) atau OFF (salah) dalam sebuah grup.
ListView	ListView memberi pengguna daftar gulir item teks.
TextField	TextField adalah komponen teks yang memungkinkan pengeditan satu baris teks.

PasswordField	PasswordField adalah komponen teks khusus untuk entri kata sandi.
Scrollbar	Scrollbar mewakili komponen scroll bar untuk memungkinkan pengguna memilih dari rentang nilai.
FileChooser	FileChooser mewakili jendela dialog dari mana pengguna dapat memilih file.
ProgressBar	Saat tugas berjalan menuju penyelesaian, ProgressBar menampilkan persentase penyelesaian tugas.
Slider	Slider memungkinkan pengguna memilih nilai secara grafis dengan menggeser kenop dalam interval terbatas.

Control lainnya di javafx:

- Accordion
- Button
- CheckBox
- ChoiceBox
- ColorPicker
- ComboBox
- DatePicker
- Label
- ListView
- Menu
- MenuBar
- PasswordField
- ProgressBar
- RadioButton
- Slider
- Spinner
- SplitMenuButto  
n
- SplitPane
- TableView
- TabPane
- TextArea
- TextField
- TitledPane
- ToggleButton
- ToolBar
- TreeTableView
- TreeView

## Layout

Layout JavaFX adalah komponen yang berisi komponen lain di dalamnya. Komponen layout mengelola tata letak komponen yang bersarang di dalamnya.

Komponen tata letak JavaFX terkadang juga disebut komponen induk karena mengandung komponen turunan, dan karena komponen tata letak adalah subkelas dari kelas JavaFX `javafx.scene.Parent`.

JavaFX memiliki tata letak berikut:

Layout	Description
HBox	HBox mengatur semua node dalam aplikasi kita dalam satu baris horizontal.
VBox	VBox mengatur semua node dalam aplikasi kita dalam satu kolom vertikal.
BorderPane	BorderPane mengatur node dalam aplikasi kita di posisi atas, kiri, kanan, bawah dan tengah.
StackPane	StackPane mengatur node dalam aplikasi kita di atas yang lain seperti di tumpukan. Node yang ditambahkan pertama ditempatkan di bagian bawah tumpukan dan node berikutnya ditempatkan di atasnya.
TextFlow	TextFlow mengatur beberapa simpul teks dalam satu aliran.
AnchorPane	AnchorPane menjangkar node dalam aplikasi kita pada jarak tertentu dari panel.
TilePane	TilePane menambahkan semua node aplikasi kita dalam bentuk ubin berukuran seragam.
GridPane	GridPane mengatur node dalam aplikasi kita sebagai grid baris dan kolom. Tata letak ini berguna saat membuat formulir menggunakan JavaFX.
FlowPane	FlowPane membungkus semua node dalam aliran. FlowPane horizontal membungkus elemen panel pada ketinggiannya, sedangkan FlowPane vertikal membungkus elemen pada lebarnya.

Layout lainnya di javafx:

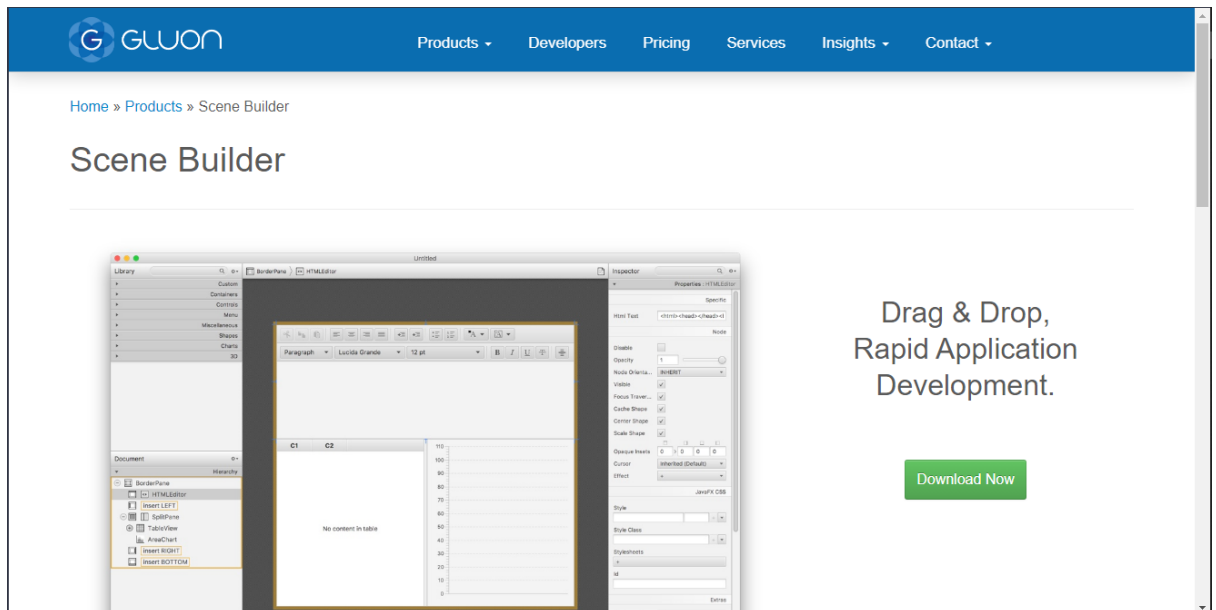
- Group
- Region
- Pane
- HBox
- VBox
- FlowPane
- BorderPane
- BorderPane
- StackPane
- TilePane
- GridPane
- AnchorPane
- TextFlow



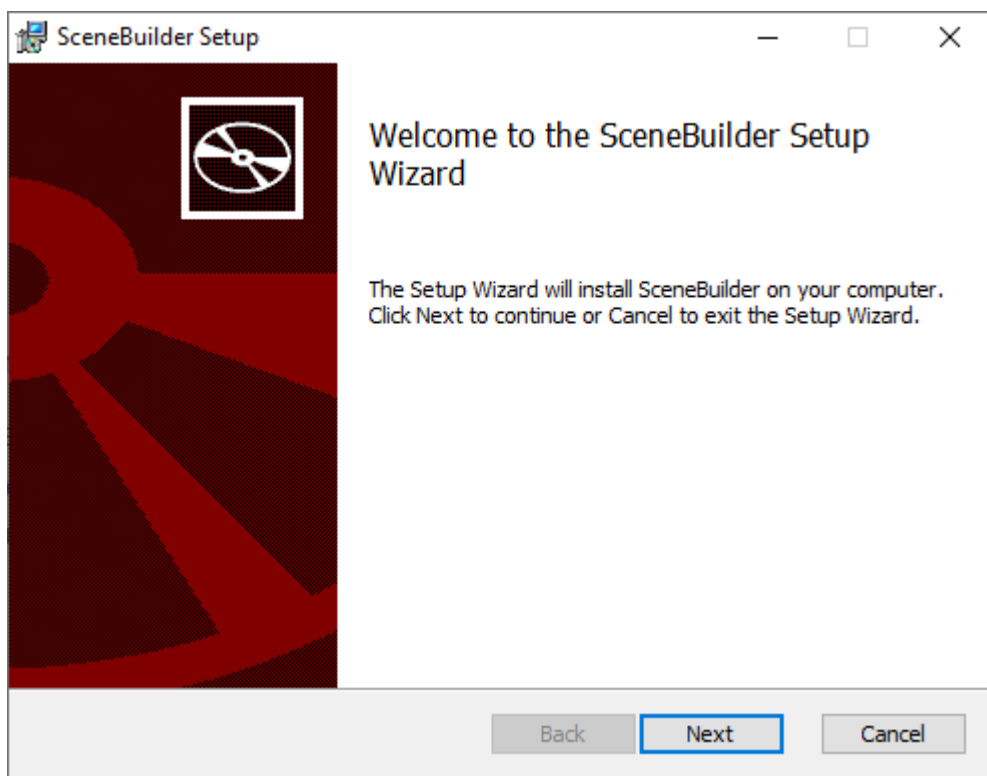
## Langkah Praktikum

### Instalasi Scene Builder

1. Unduh installer Scene Builder melalui link [Download Scene Builder](#)



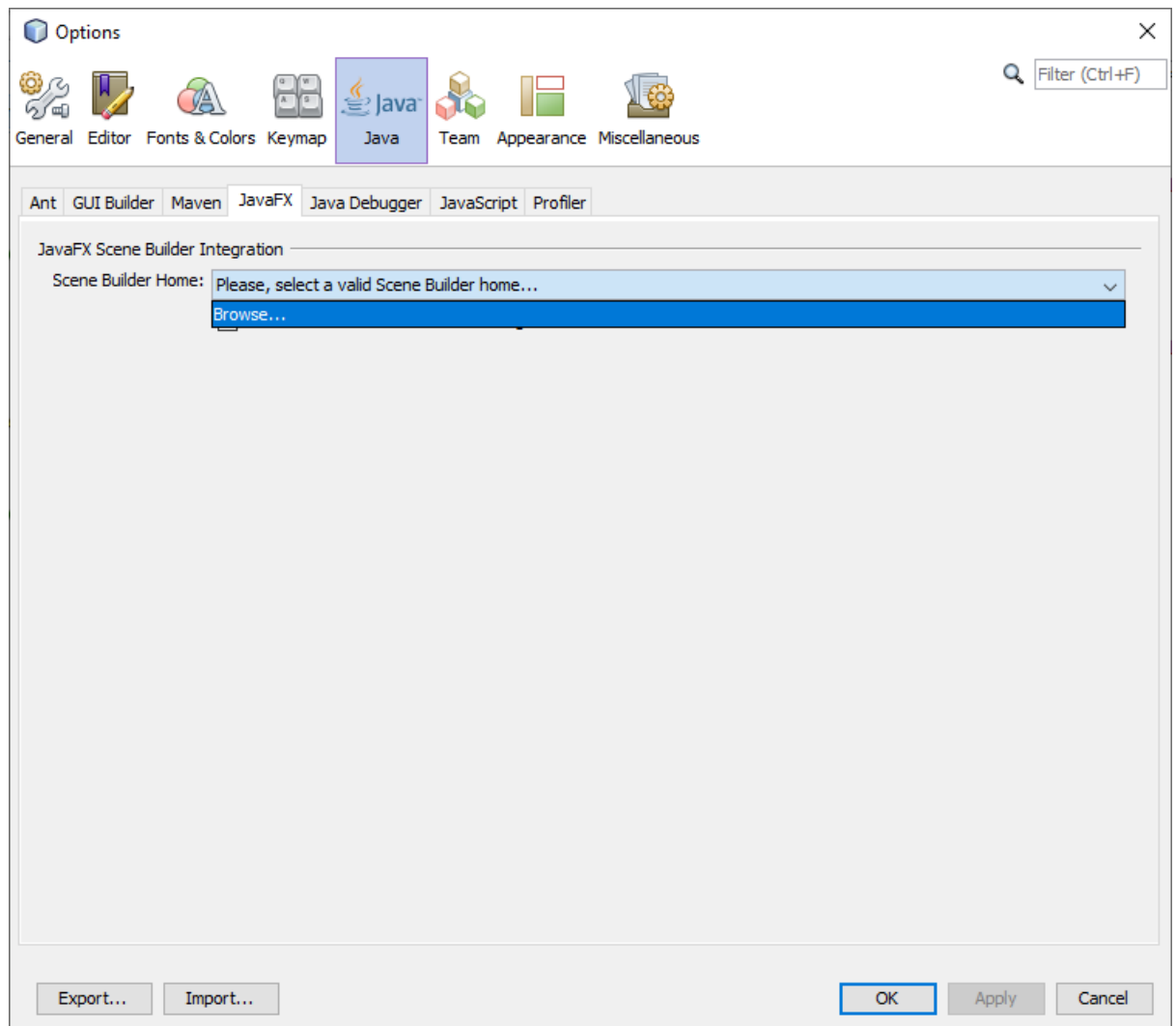
2. Jalankan file installernya



3. Lanjutkan langkah instalasi hingga selesai



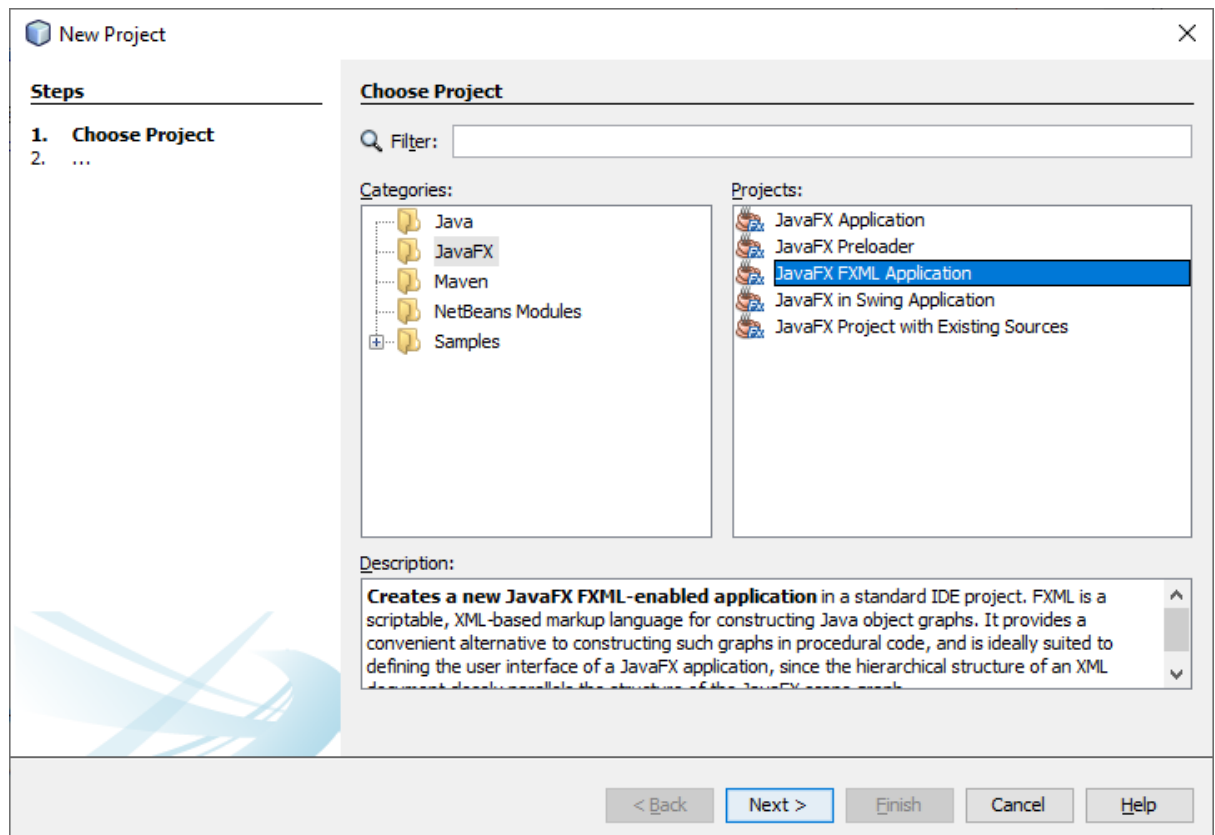
4. Tambahkan SceneBuilder ke dalam NetBeans dengan cara mengakses **Tools > Options > Java > JavaFX**. Kemudian pilih lokasi instalasi scene builder yang telah dipilih sebelumnya.



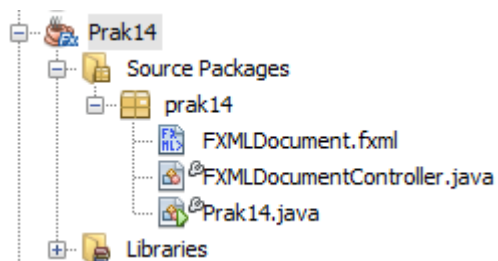
## Membuat Project

1. Buka NetBeans

2. Buat Project Baru, kali ini pilih **categories “JavaFX”** dan tipe Project **“JavaFX FXML Application”**



3. Beri nama project tersebut **prak14\_nama**
4. Struktur project yang pertama kali dihasilkan akan menjadi seperti ini



5. Class utama yang pertama kali dihasilkan akan memiliki 2 method utama seperti berikut

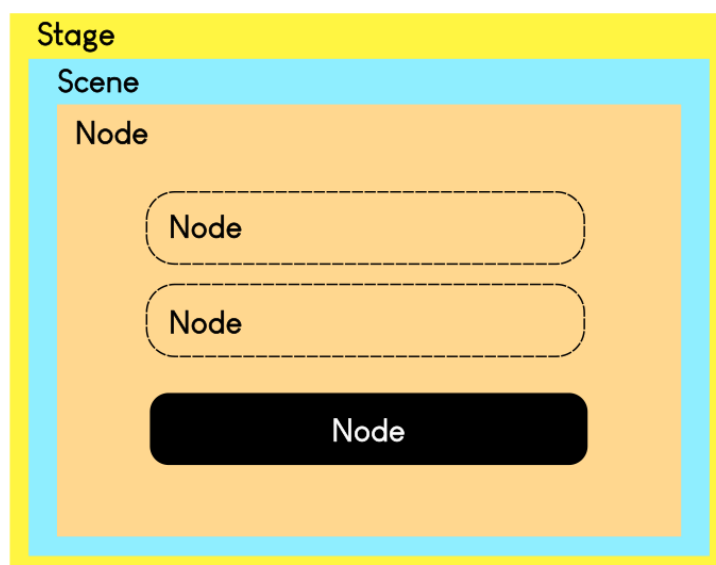
```
@Override
public void start(Stage stage) throws Exception {

}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    launch(args);
}
```

### Membuat GUI Sederhana

1. Untuk melakukan coding yang berkaitan dengan GUI anda dapat melakukannya di dalam method **start**
2. Selalu ingat bahwa struktur dari JavaFX adalah seperti berikut



3. Pertama-tama buatlah sebuah object **button** yang nantinya akan dimasukkan ke dalam **scene**

```
public void start(Stage stage) throws Exception {
    Button btn = new Button("Click Me!");
}
```

4. Kemudian buat sebuah objek **StackPane** untuk menampung node yang akan ditampilkan. Kemudian tambahkan **button** yang telah dibuat sebelumnya.

```
public void start(Stage stage) throws Exception {
    Button btn = new Button("Click Me!");

    StackPane root = new StackPane();
    root.getChildren().add(btn);
}
```

5. Kemudian buat sebuah **Scene** yang akan menampung node yang telah dibuat

```
public void start(Stage stage) throws Exception {
    Button btn = new Button("Click Me!");

    StackPane root = new StackPane();
    root.getChildren().add(btn);

    Scene scene = new Scene(root);
}
```

6. Terakhir atur stage yang sudah ada pada parameter. Anda dapat mengatur judul, ukuran, dan isi dari stage sesuai keinginan. Diakhiri dengan method **show** yang akan menampilkan stage yang sudah dipersiapkan.

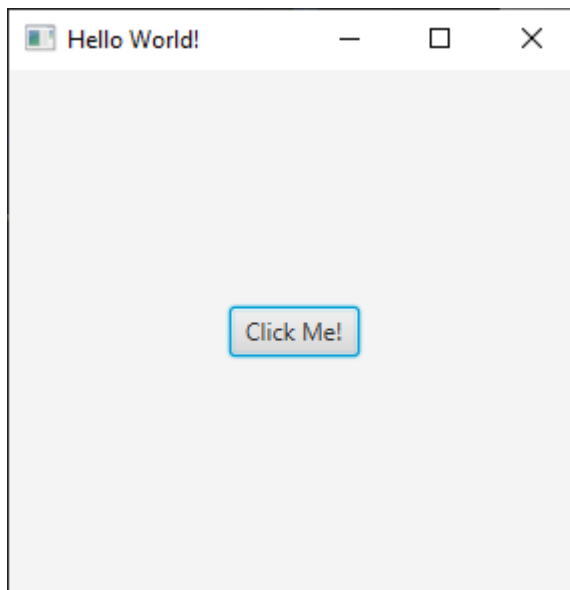
```
public void start(Stage stage) throws Exception {
    Button btn = new Button("Click Me!");

    StackPane root = new StackPane();
    root.getChildren().add(btn);

    Scene scene = new Scene(root);

    stage.setHeight(300);
    stage.setWidth(300);
    stage.setScene(scene);
    stage.setTitle("Hello World!");
    stage.show();
}
```

7. Jika anda mencoba menjalankan project tersebut, maka akan muncul tampilan seperti berikut



### Membuat Interaksi Sederhana

1. Sekarang anda diminta untuk membuat interaksi apabila **button** diklik
2. Pertama-tama ubah **StackPane** yang digunakan sebelumnya menjadi

#### **VBox**

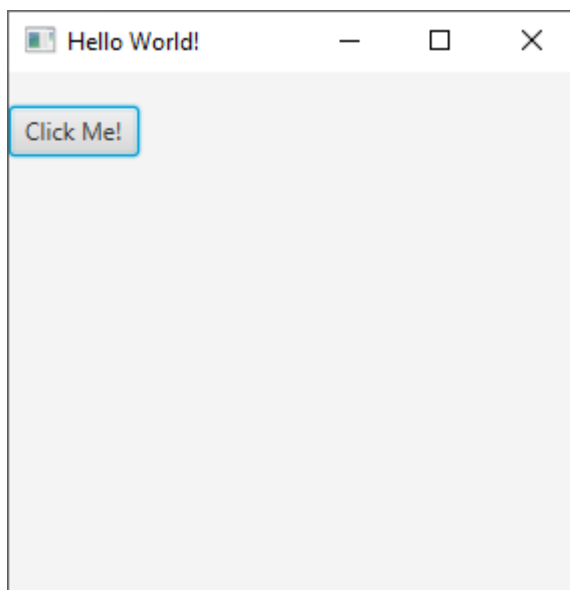
```
public void start(Stage stage) throws Exception {  
    Button btn = new Button("Click Me!");  
  
    VBox root = new VBox();  
    root.getChildren().add(btn);  
  
    Scene scene = new Scene(root);  
  
    stage.setHeight(300);  
    stage.setWidth(300);  
    stage.setScene(scene);  
    stage.setTitle("Hello World!");  
    stage.show();  
}
```

3. Kemudian buat sebuah objek **Label** dengan isi awalnya string kosong. Kemudian tambahkan label ini ke dalam VBox dengan posisi di atas

## Button

```
public void start(Stage stage) throws Exception {  
  
    Button btn = new Button("Click Me!");  
    Label txt = new Label("");  
  
    VBox root = new VBox();  
    root.getChildren().add(txt);  
    root.getChildren().add(btn);  
  
    Scene scene = new Scene(root);  
  
    stage.setHeight(300);  
    stage.setWidth(300);  
    stage.setScene(scene);  
    stage.setTitle("Hello World!");  
    stage.show();  
}
```

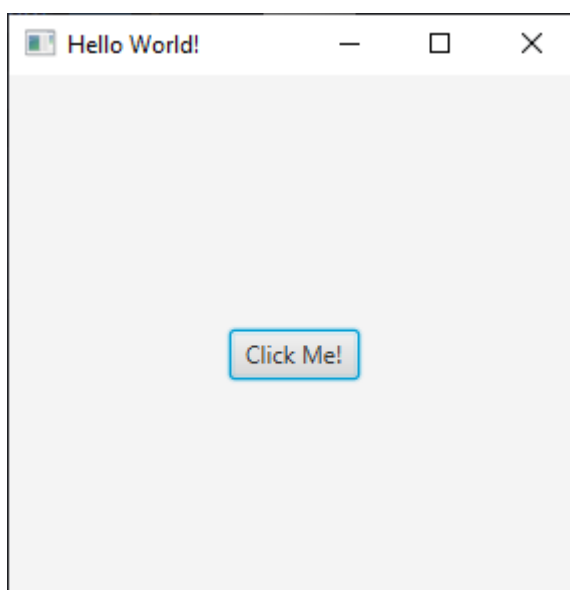
4. Jika dijalankan maka tampilannya akan seperti berikut



5. Untuk membuat tampilannya berada di tengah anda dapat menggunakan method **setAlignment** yang ada pada **VBox**

```
public void start(Stage stage) throws Exception {  
  
    Button btn = new Button("Click Me!");  
    Label txt = new Label("");  
  
    VBox root = new VBox();  
    root.setAlignment(Pos.CENTER);  
    root.getChildren().add(txt);  
    root.getChildren().add(btn);  
  
    Scene scene = new Scene(root);  
  
    stage.setHeight(300);  
    stage.setWidth(300);  
    stage.setScene(scene);  
    stage.setTitle("Hello World!");  
    stage.show();  
}
```

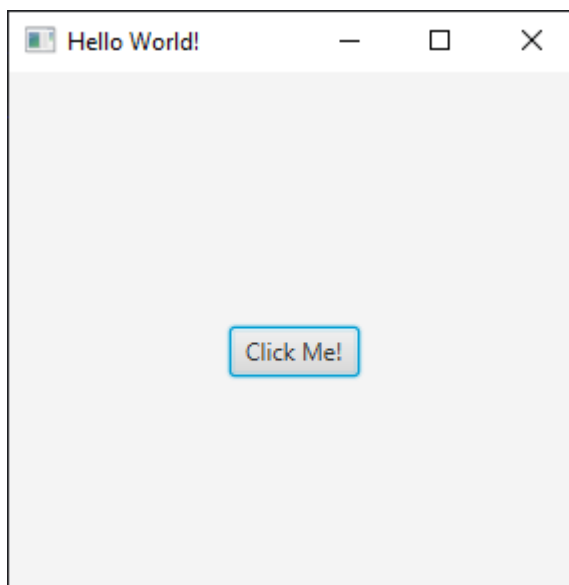
Tampilan :



6. Buatlah sebuah **setOnAction** ketika **Button** diklik maka isi dari Label akan menjadi **"Hello World!"**

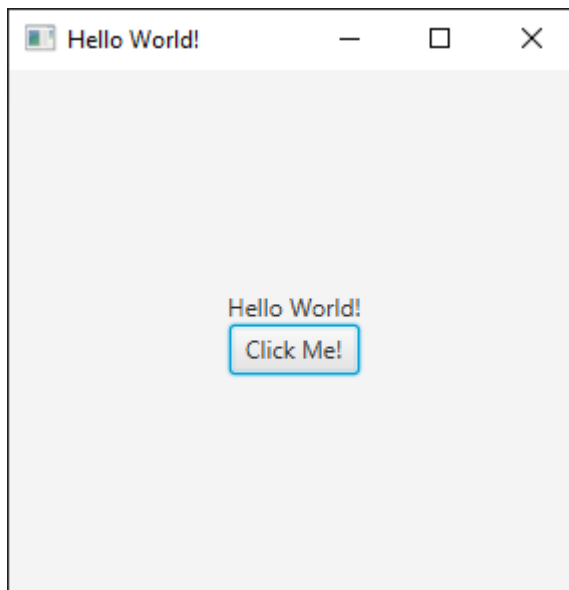
```
public void start(Stage stage) throws Exception {  
  
    Button btn = new Button("Click Me!");  
    Label txt = new Label("");  
  
    btn.setOnAction(new EventHandler<ActionEvent>() {  
  
        @Override  
        public void handle(ActionEvent event) {  
            txt.setText("Hello World!");  
        }  
  
    });  
  
    VBox root = new VBox();  
    root.setAlignment(Pos.CENTER);  
    root.getChildren().add(txt);  
    root.getChildren().add(btn);  
  
    Scene scene = new Scene(root);  
  
    stage.setHeight(300);  
    stage.setWidth(300);  
    stage.setScene(scene);  
    stage.setTitle("Hello World!");  
    stage.show();  
}
```

7. Jika dijalankan maka akan tampil seperti berikut  
Sebelum tombol diklik:



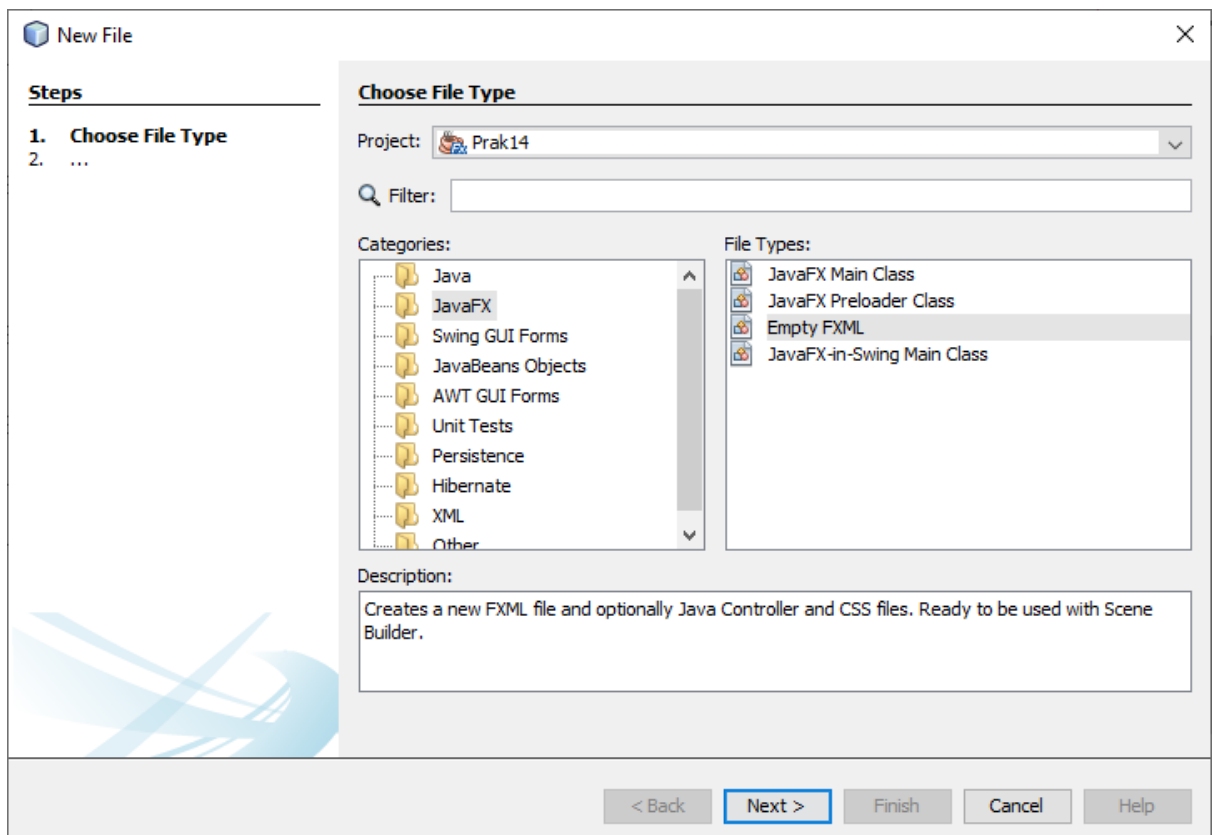


Sesudah Tombol Diklik

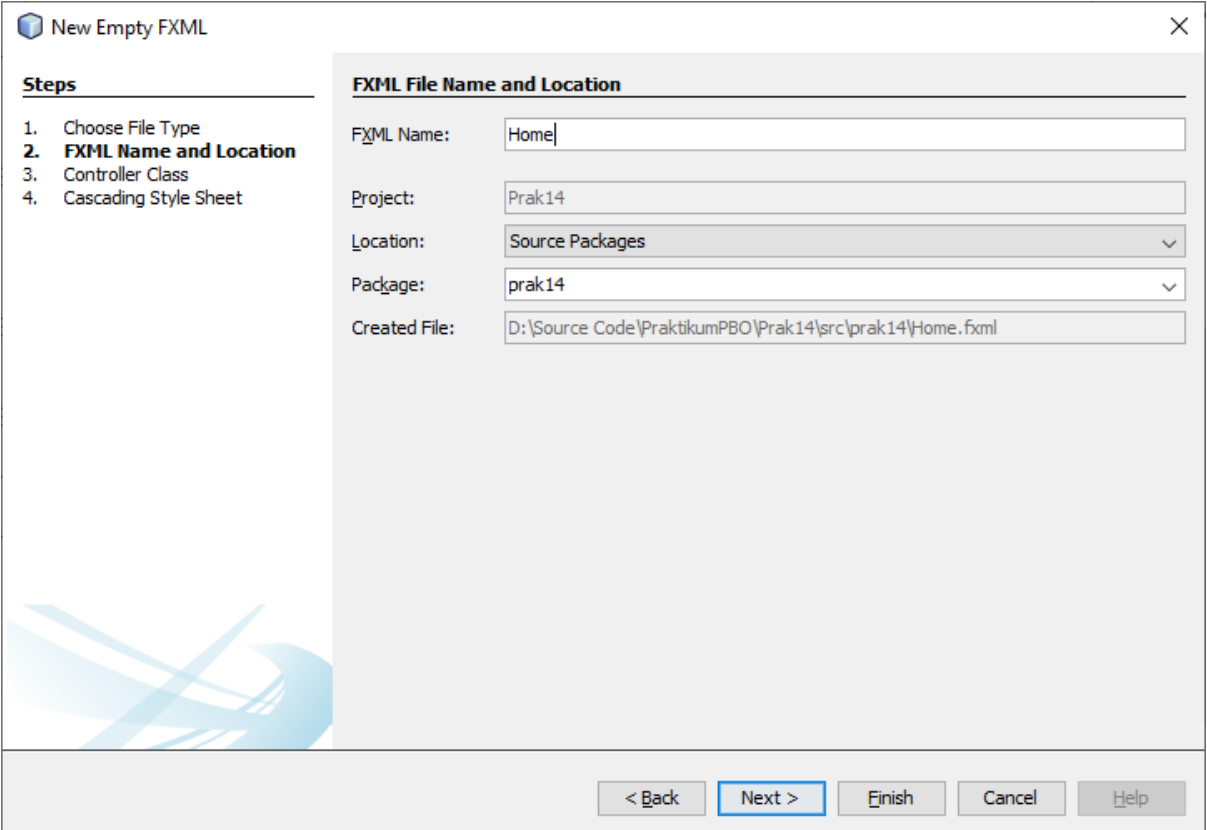


## Membuat GUI Menggunakan Scene Builder

1. Buatlah sebuah file FXML baru pada project anda. Dengan cara **klik kanan > new > other**. Kemudian pilih **JavaFX** dan cari **Empty FXML**



2. Beri nama file tersebut **Home**



New Empty FXML

**Steps**

1. Choose File Type
- 2. FXML Name and Location**
3. Controller Class
4. Cascading Style Sheet

**FXML File Name and Location**

FXML Name:

Project:

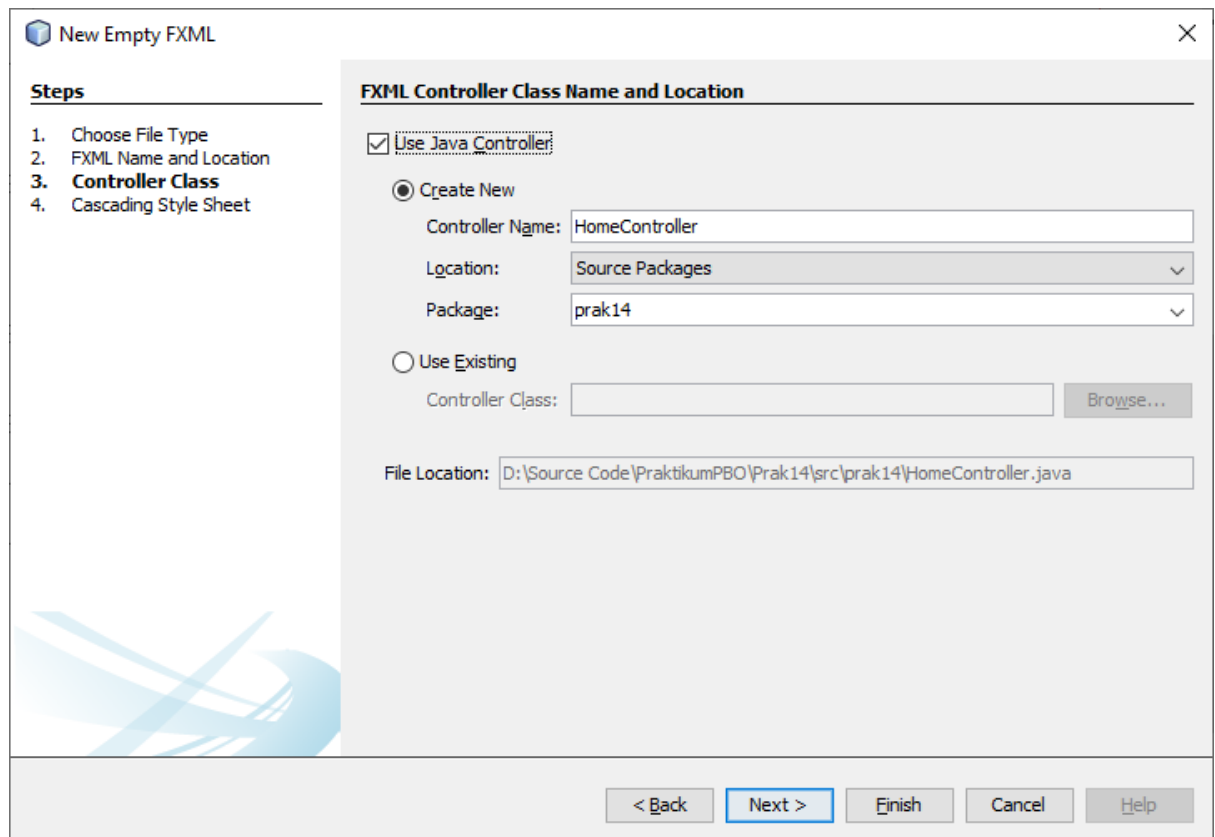
Location:

Package:

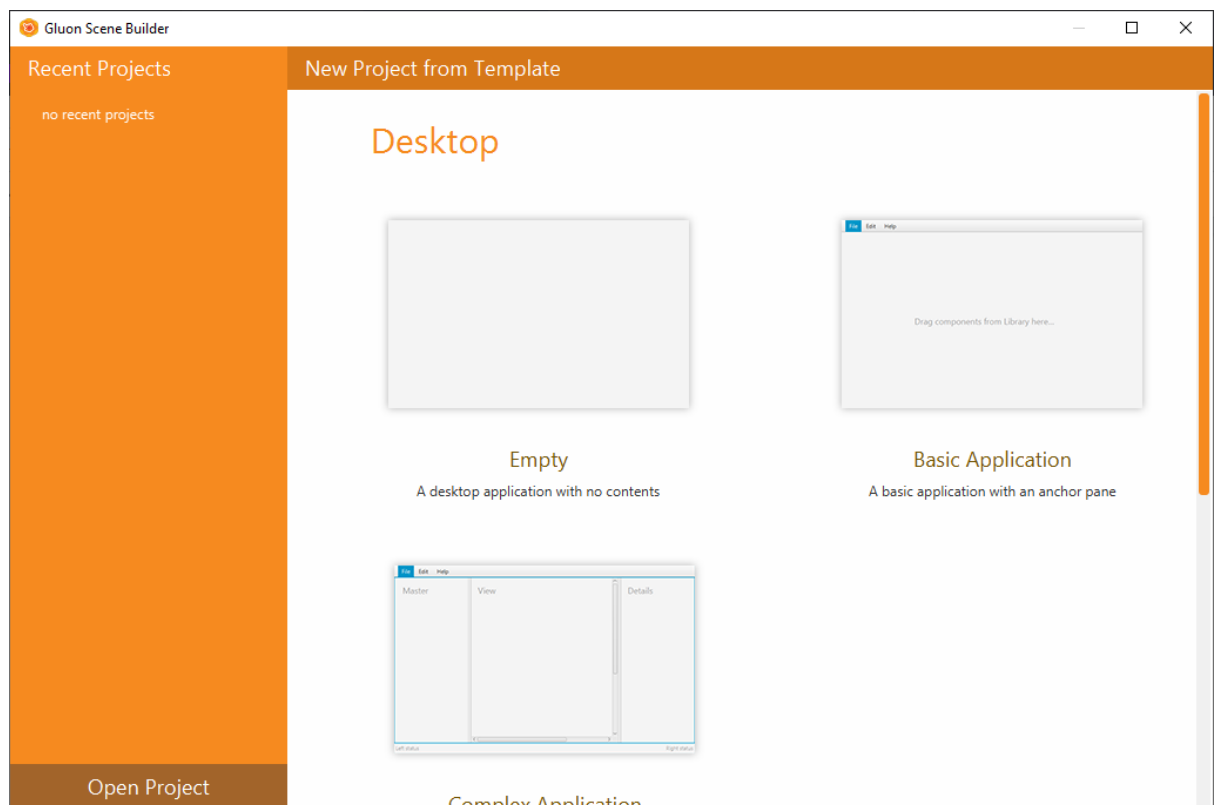
Created File:

< Back   Next >   Finish   Cancel   Help

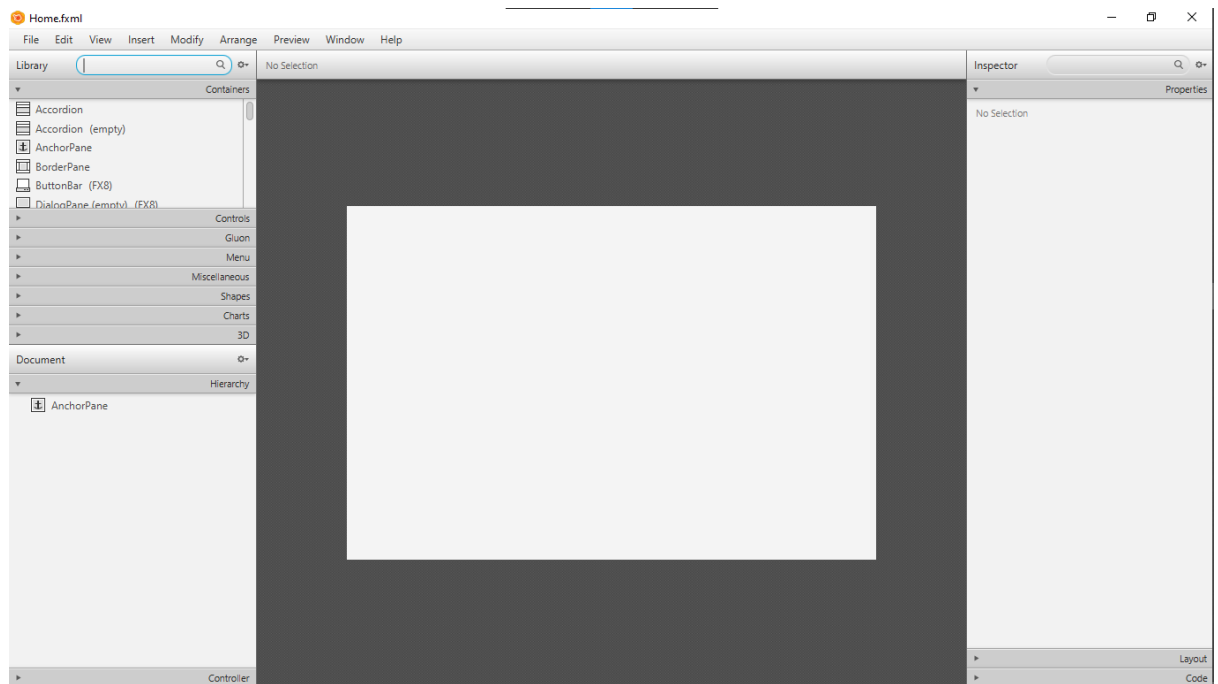
- Centang kolom use Java Controller dan biarkan yang lainnya secara default



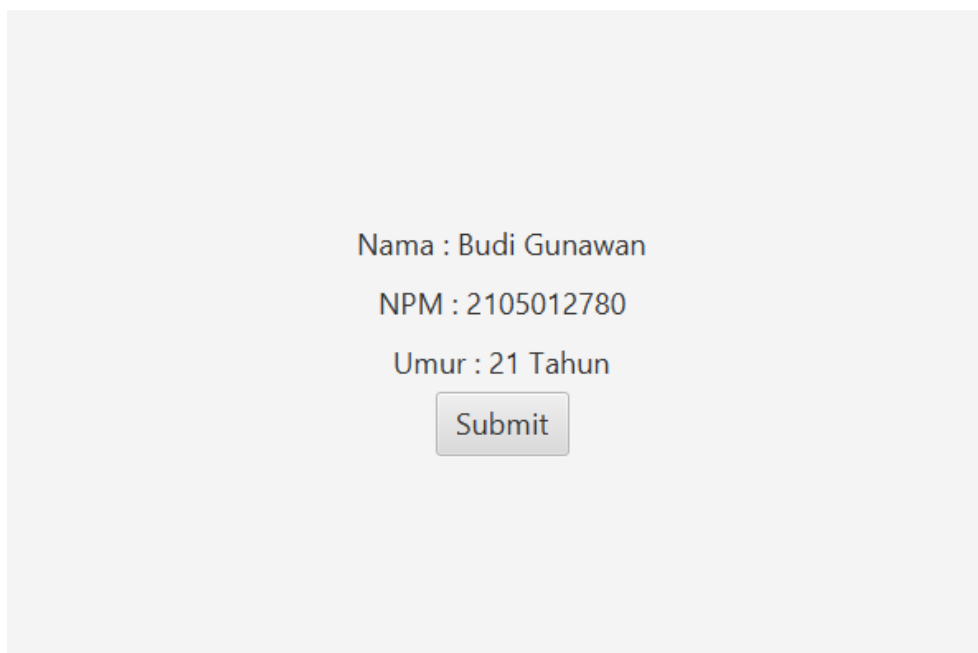
- Buka Software **Scene Builder** yang telah diinstal sebelumnya



5. Klik Open Project, kemudian cari file FXML yang baru dibuat pada project anda.
6. Jika berhasil membuka maka tampilan awal akan terlihat seperti ini



7. Ubah Isi dari tampilan tersebut menjadi seperti berikut



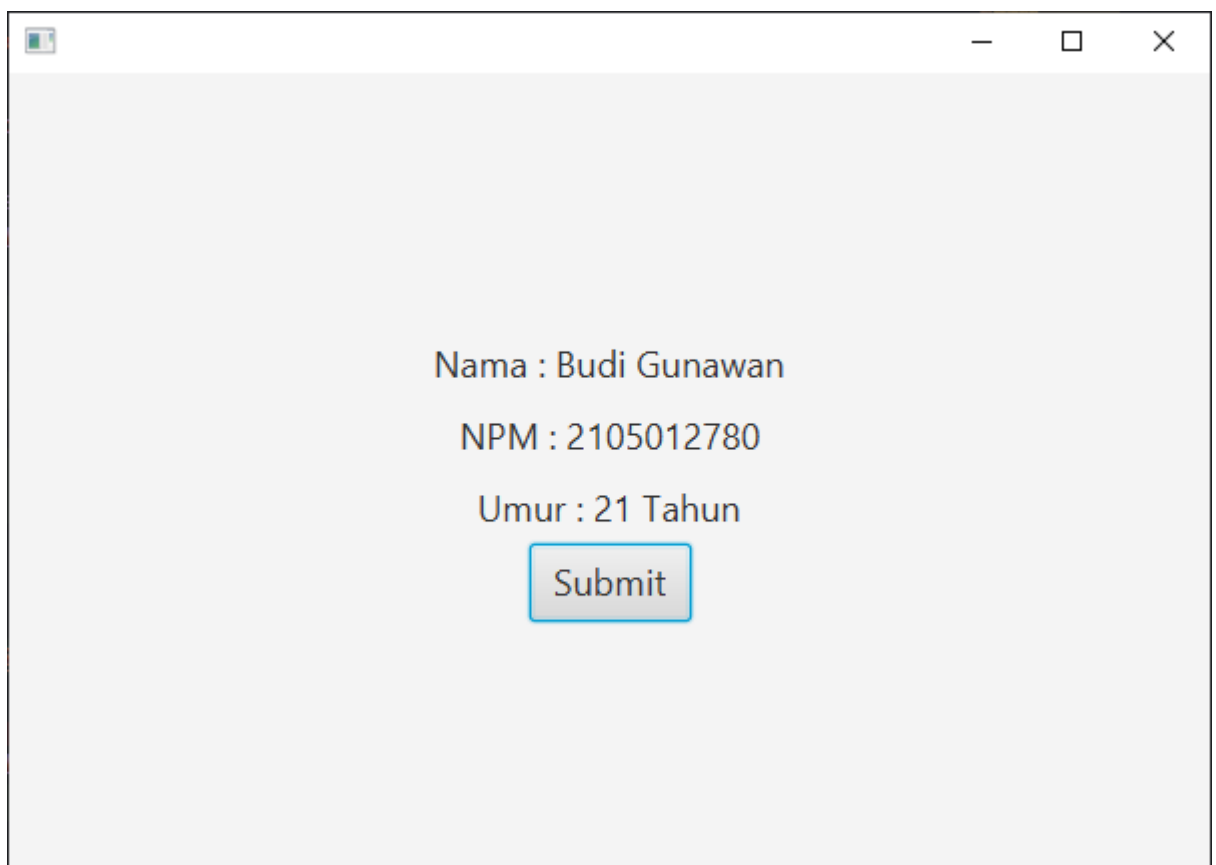
8. Save hasil yang telah diubah kemudian kembali ke NetBeans

## Menghubungkan Program dengan FXML

1. Pada Class utama, ubahlah isi dari method **start** agar dapat menampilkan UI yang telah dibuat menggunakan Scene Builder.
2. Anda dapat menggunakan Class **FXMLLoader** untuk mengambil resource dari FXML

```
public void start(Stage stage) throws Exception {  
  
    Parent root = FXMLLoader.load(getClass().getResource("Home.fxml"));  
  
    Scene scene = new Scene(root);  
  
    stage.setScene(scene);  
    stage.show();  
}
```

3. Maka ketika dijalankan akan tampil seperti berikut



## Tugas

1. Lanjutkan program yang sudah dibuat.
2. Ubah nama stage menjadi **Nama-NPM**
3. Kumpulkan Screenshoot hasil runningnya ke vclass!

