VERSION 1.0

MEI, 2022

PEMROGRAMAN BERORIENTASI OBJEK

PEMROGRAMAN GUI DENGAN JAVAFX. MODUL 6

DISUSUN OLEH :

- Muhammad Rahadian Arya Saputra
- Riyan Putra Firjatullah

DIAUDIT OLEH :

- Ali Sofyan Kholimi, M.Kom.

PRESENTED BY: TIM LAB-IT

UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN BEROIRENTASI OBJEK

TUJUAN

1. Mahasiswa mampu membangun aplikasi sederhana menggunakan paradigma pemrograman terstruktur.
2. Mahasiswa mampu membangun aplikasi sederhana menggunakan paradigma object oriented.

TARGET MODUL

1. Mahasiswa mampu memahami konsep Encapsulation
2. Mahasiswa mampu memahami & menerapkan Object Oriented Programming

PERSIAPAN SOFTWARE/APLIKASI

1. Compiler java (JDK), JRE.
2. Editor Java (NetBeans, Gel, Eclipse, Jcreator, dll).

KEYWORDS

JavaFx

Table JavaFx

Form JavaFx

PERSIAPAN MATERI

➢ 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 mengantikan 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.

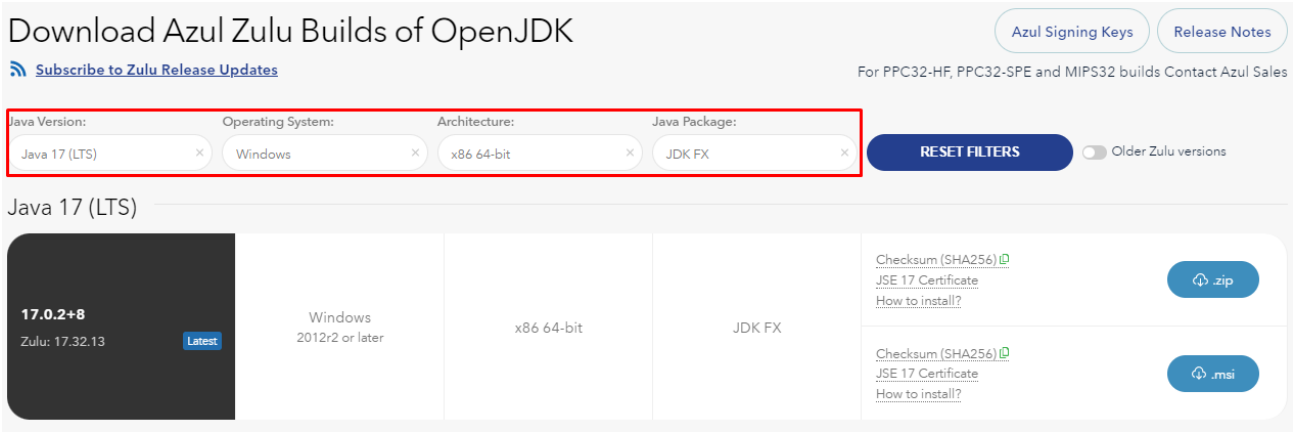
➢ Instalasi

Dalam praktikum kali ini digunakan lingkungan pengembangan Apache Netbeans dan java JDK 17 (Last Version) yang didalamnya termasuk JavaFX 17. Berikut ini adalah langkah-langkah instalasi lingkungan pengembangan Apache Netbeans untuk memprogram JavaFx:

A. **Download File JDK FX**

Download file terlebih dahulu untuk dapat menggunakan JavaFX di:

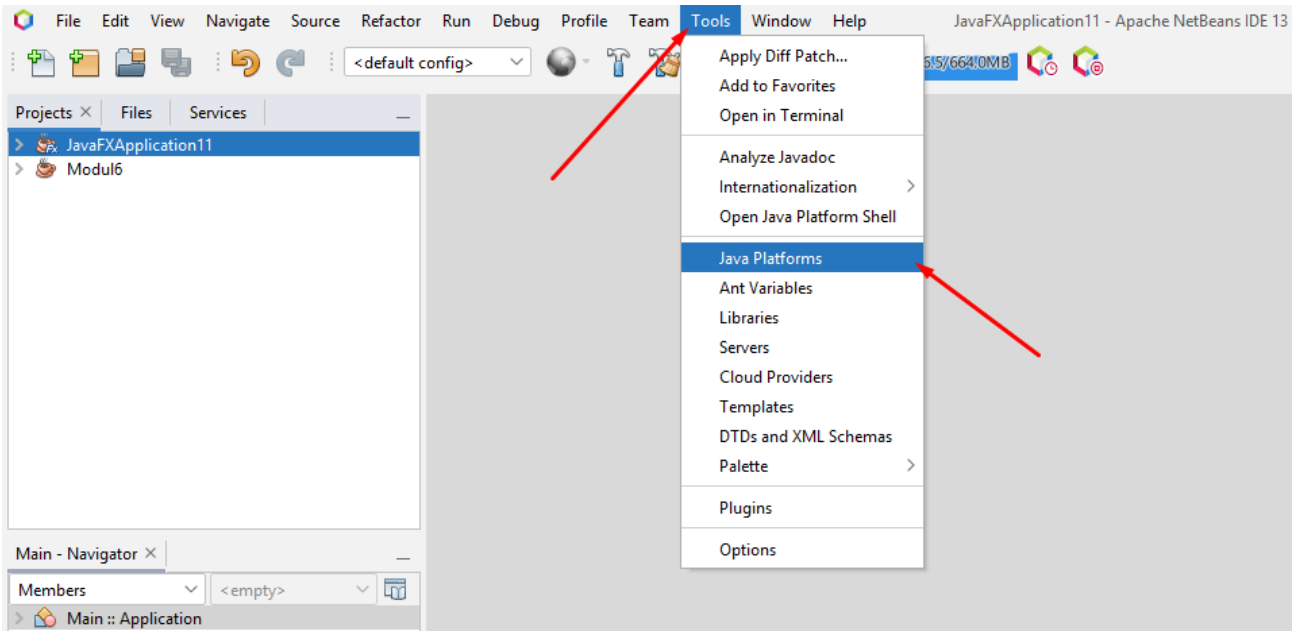
https://www.azul.com/downloads/ Kemudian scroll hingga akhir dan ubah pada bagian Java Version, Operating Systems, Architecture, dan Java Package seperti pada gambar dibawah dengan format .zip.



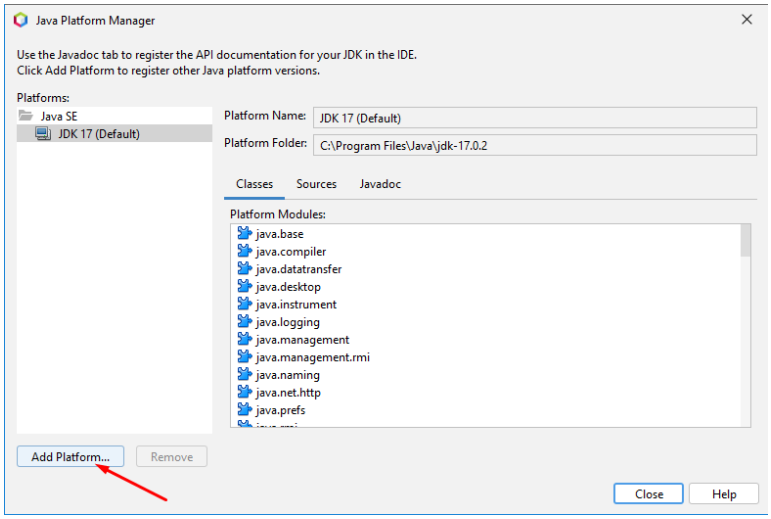
Kemudian setelah selesai proses downloading, ekstrak file .zip tersebut di folder yang kalian inginkan.

B. **Konfigurasi Apache Netbeans**

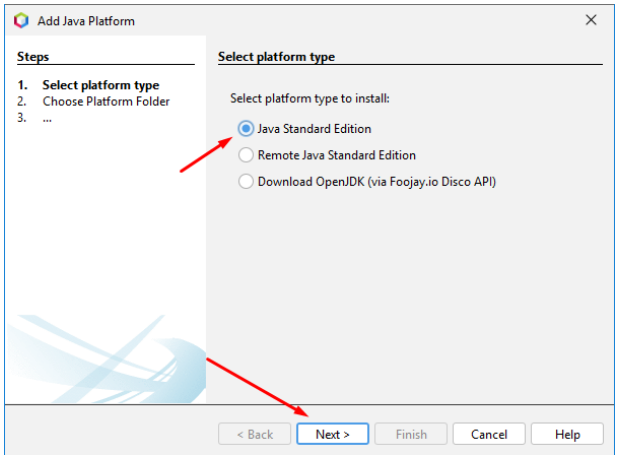
Buka IDE Apache Netbeans kalian, kemudian pilih Tools → Java Platforms.



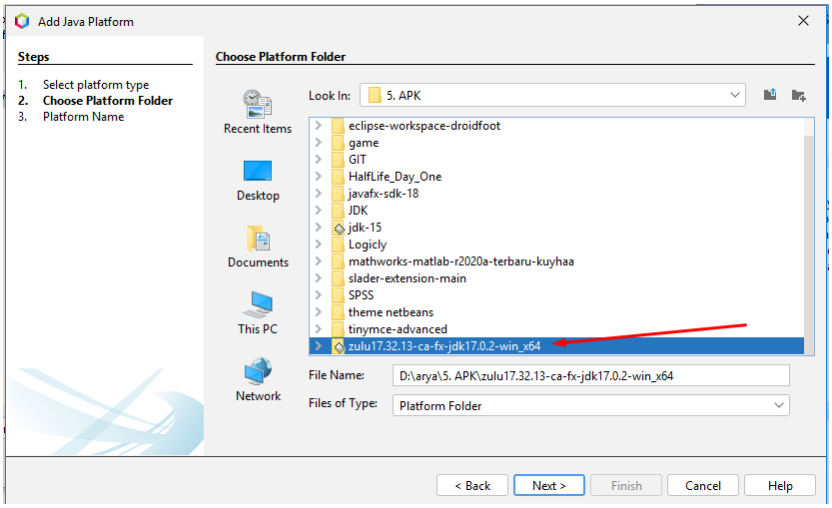
Seteleh Java Platform Manager terbuka, pilih Add Platform dibagian kiri bawah.



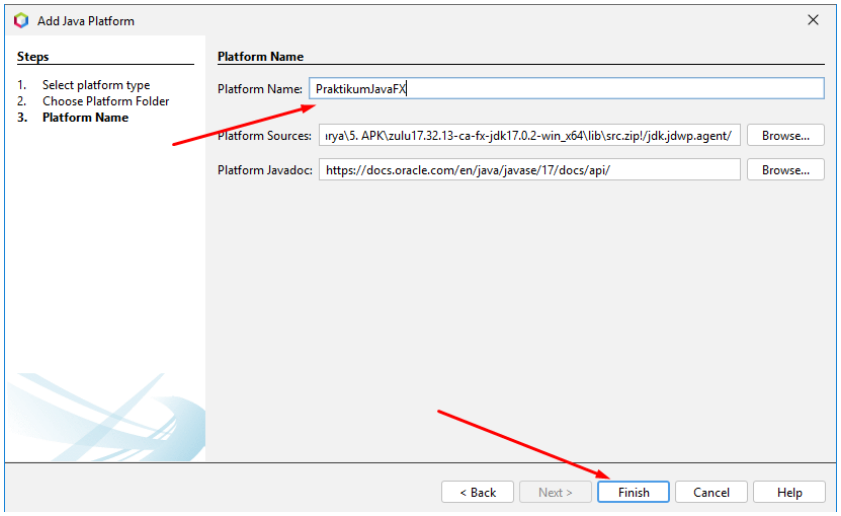
Setelah itu pilih Java Standard Edition dan tekan Next.



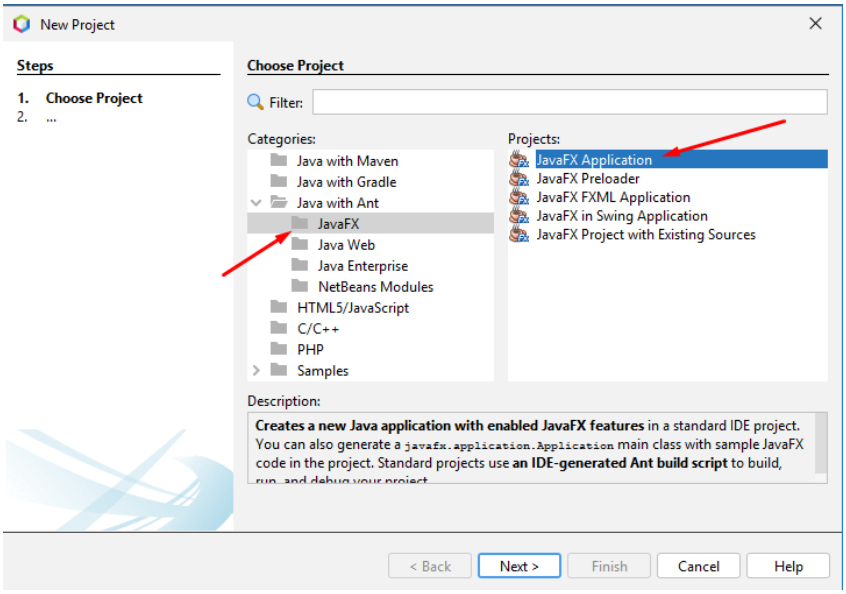
Arahkan pada folder tempat kalian extrack file yang kalian download sebelumnya.

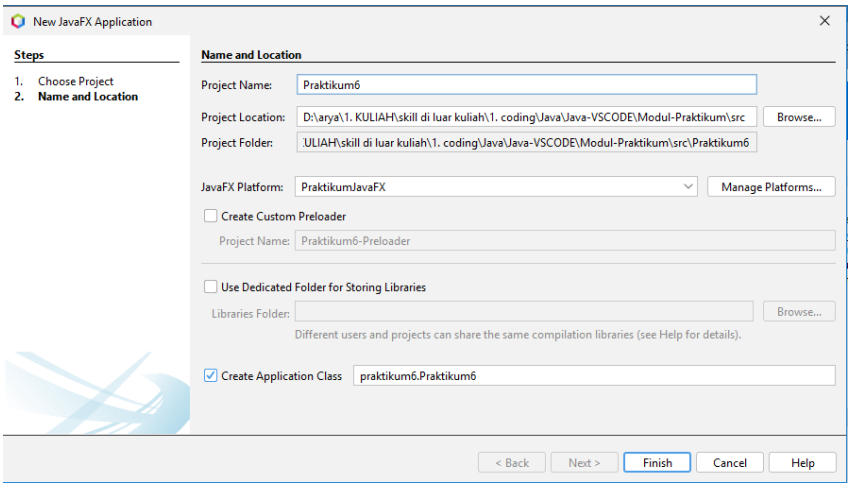


Kemudian beri nama Platform Name sesuai keinginan kalian, lalu tekan finish.

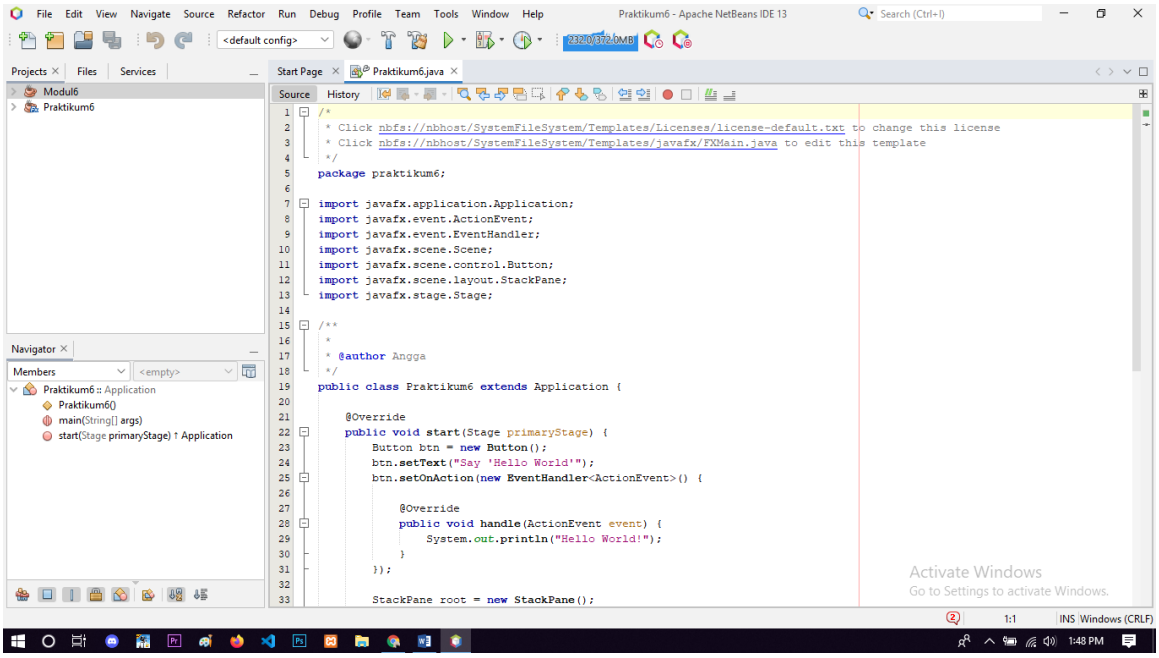


Buat project baru dan pilih JavaFX → JavaFX Application.





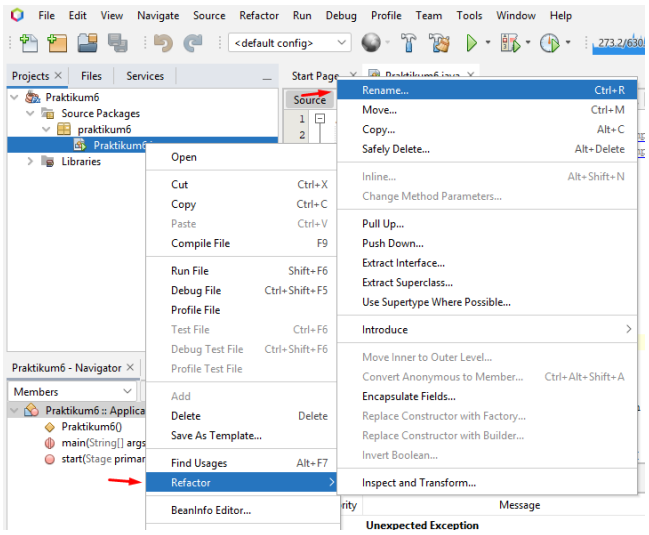
Maka akan menampilkan tampilan berikut:

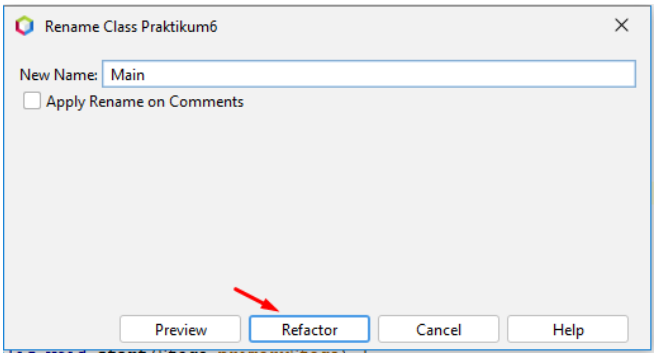



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/javafx/FXMain.java to edit this template
 */
package praktikum6;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

/**
 *
 * @author Angga
 */
public class Praktikum6 extends Application {

    @Override
    public void start(Stage primaryStage) {
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

        StackPane root = new StackPane();
```

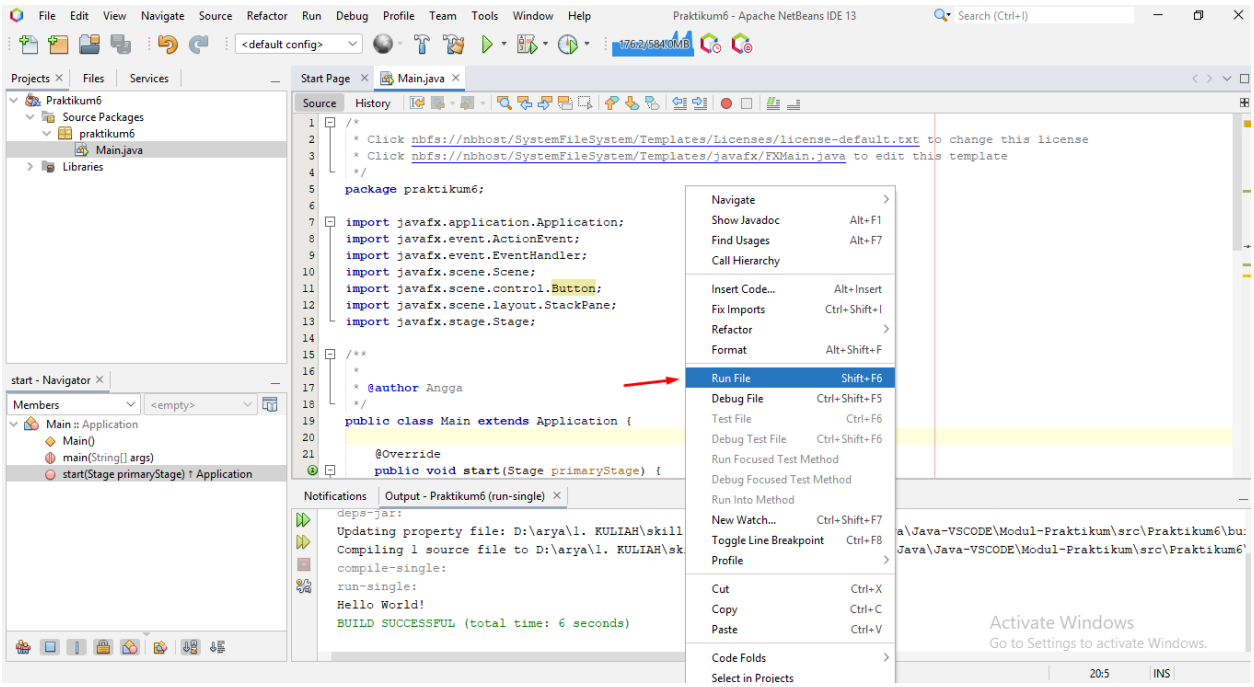
Jika pada library import javafx.application.Application; error. Maka lakukan hal dibawah ini.

Pada class utama kalian Refactor → Rename dan kalian ubah menjadi class Main

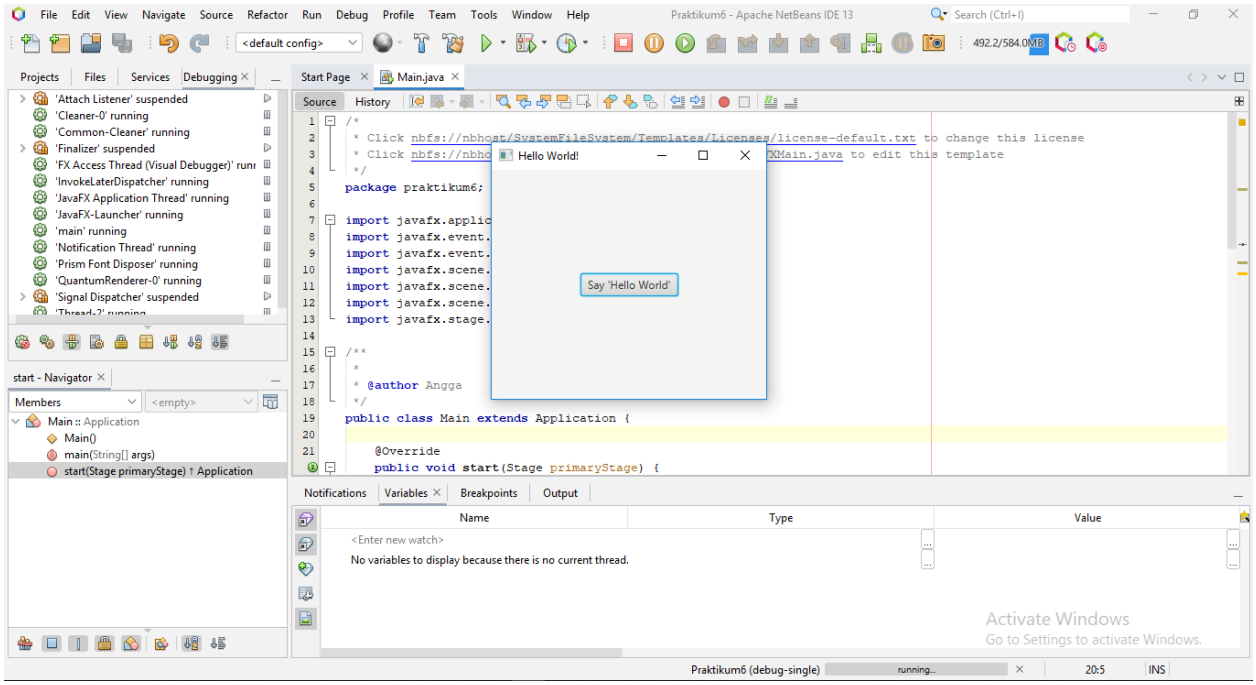




Setelah itu kalian tekan kanan pada mouse/mouse pad kalian, dan pilih Run File.



Maka program akan menampilkan GUI (graphical user interface) seperti pada gambar dibawah ini:



KEGIATAN PERCOBAAN

PERCOBAAN 1

Membuat Form JavaFx

Membuat form adalah dasar dalam membangun aplikasi. Baik tidaknya aplikasi form akan menetukan 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();

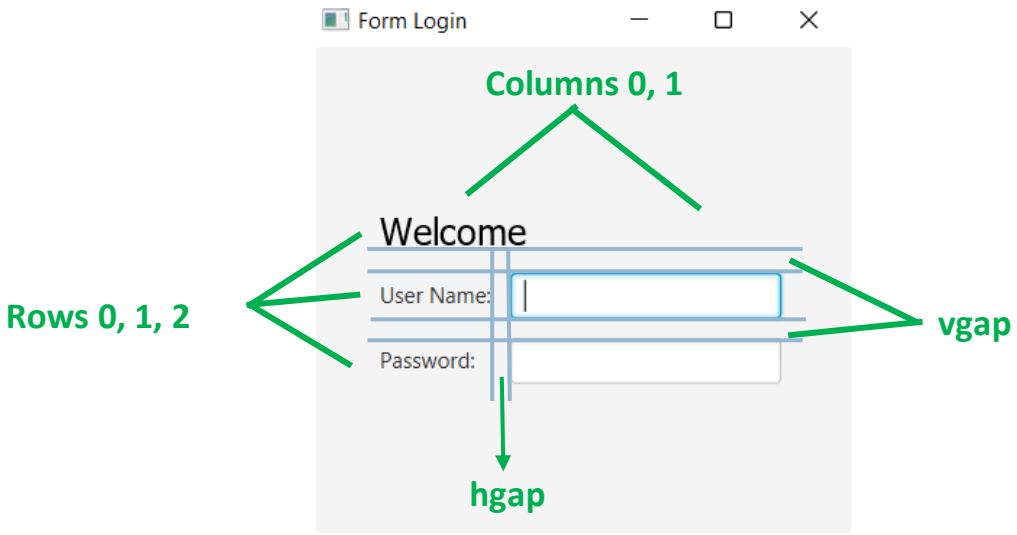
```
GridPane grid = new GridPane();
grid.setAlignment(Pos.CENTER);
grid.setHgap(10);
grid.setVgap(10);
grid.setPadding(new Insets(25, 25, 25, 25));

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);
```

```
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);
```

Tampilan program login akan Nampak 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");
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);
```

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

- Membuat tampilan awal tabel

```
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 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);
    }

}
```

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

```
final ObservableList<Mahasiswa> data = FXCollections.observableArrayList(
        new Mahasiswa("Syahrul", "202010370311299", "syahrul@gmail.com" ),
        new Mahasiswa("Azka", "202010370311311", "azka@gmail.com" )
);
```

Langkah selanjutnya adalah mengasosiasikan data ke dalam colom, 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);
```

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 tambahka 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.

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

Lalu coba jalankan dan coba masukan inputannya.

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

LEMBAR KERJA

KEGIATAN 1

Kerjakan salah satu soal dibawah ini tanpa menggunakan **Scene Builder** :

- **Soal A (Easy)**

  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 diklik 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.

- **Soal B (Hard)**

  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 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.

*Jika ada source code yang identik, maka akan dilakukan pengurangan nilai.

RUBRIK PENILAIAN

Aspek Penilaian	Poin
Kegiatan 1 Soal A	40
Kegiatan 1 Soal B	60
Pemahaman	40
Total	**80 | 100**