**UJIAN AKHIR SEMESTER 3**

**TAHUN 2021**

**LAPORAN**

**PEMPROGRAMAN BERORIENTASI OBJEK GRAPHICAL USER INTERFACE (GUI)**

DOSEN PENGAMPU : Ir. Gede Humaswara Prathama, S.T., M.T.



OLEH :

Nama : Bagus Nararya Nanda Raditya

NIM : (42030064)

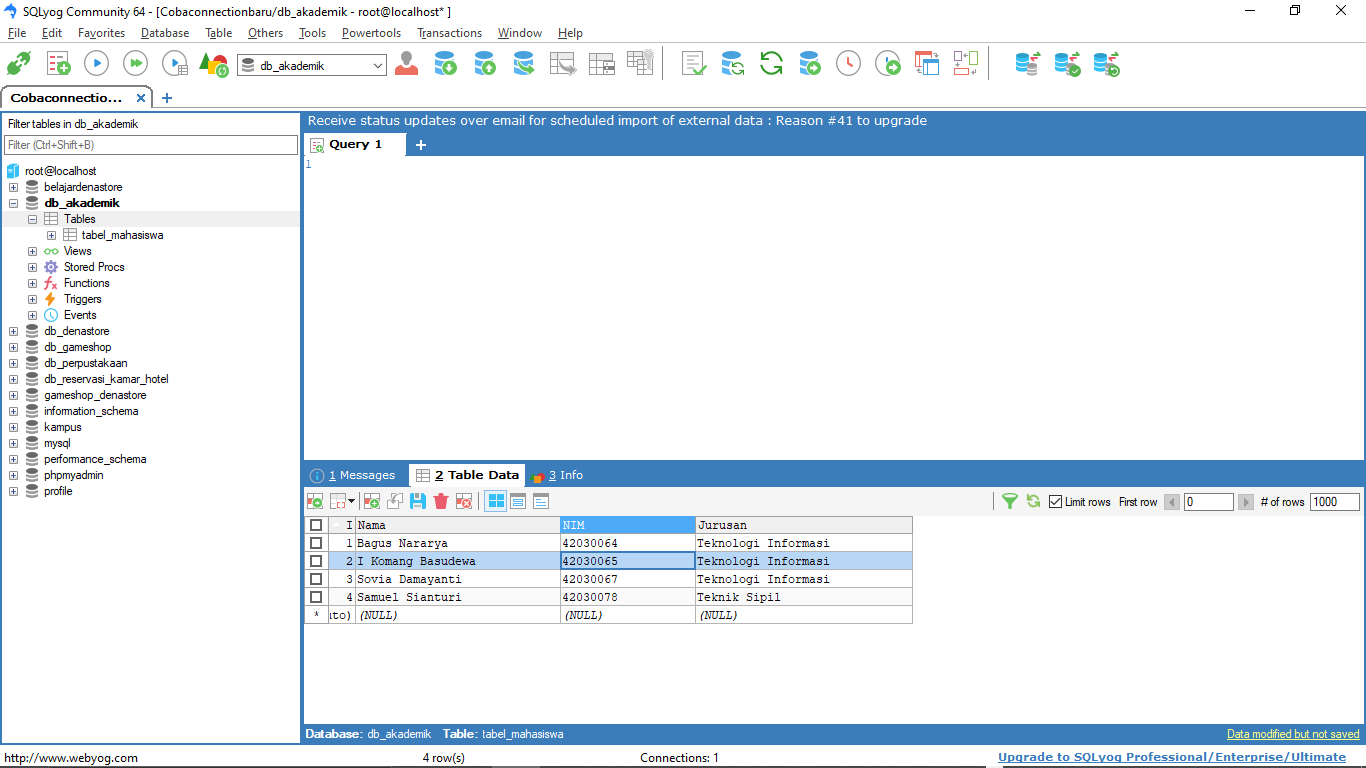
Jurusan : Teknologi Informasi

**FAKULTAS TEKNIK DAN INFORMATIKA**

**UNIVERSITAS PENDIDIKAN NASIONAL (UNDIKNAS)**

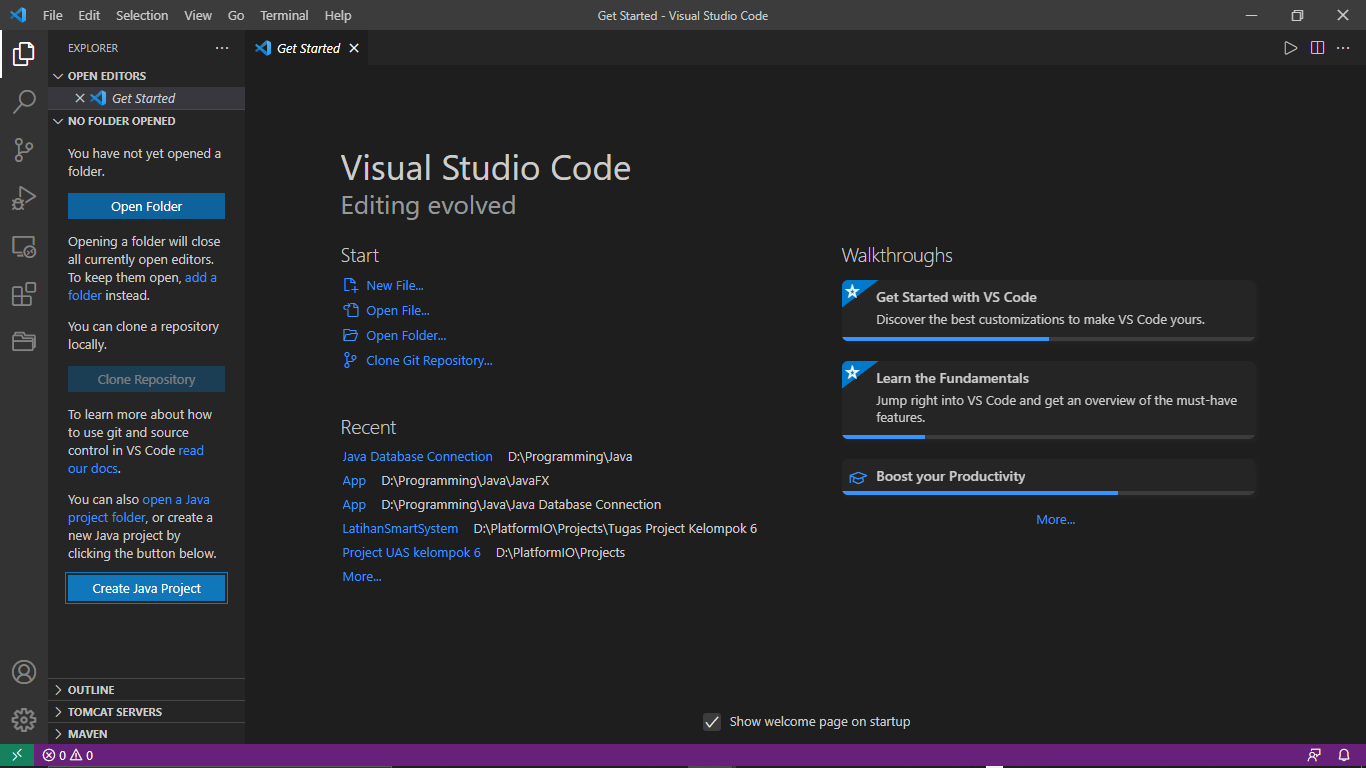
1. Langkah pertama yang harus dilakukan adalah menyiapkan beberapa komponen sebelum membuat *Grafical User Interface.*
2. **Membuat database dan tabel**

Disini saya telah membuat sebuah database yang berisikan 1 buah table seperti pada gambar:

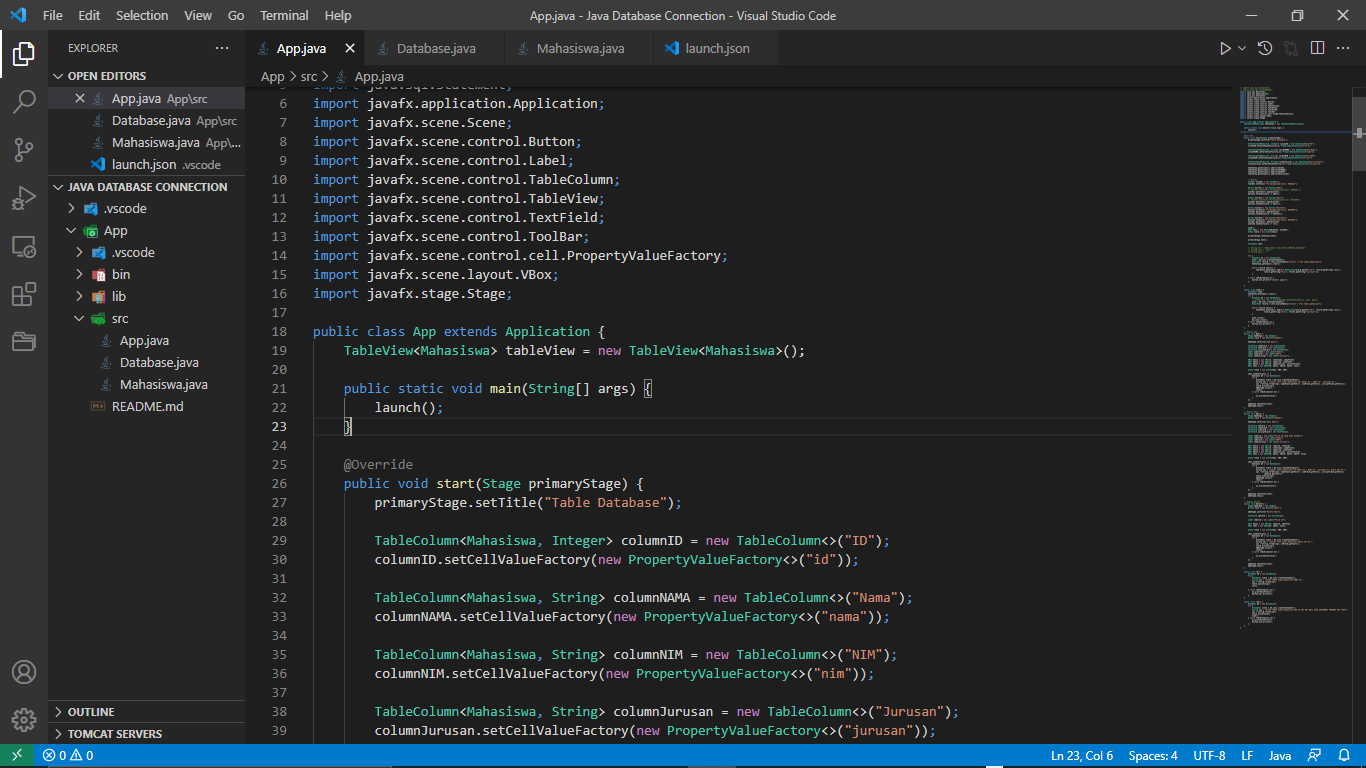


1. **Membuat New Java Project pada VSCODE**

Pastikan anda telah membuat project berbasis java seperti pada gambar berikut:

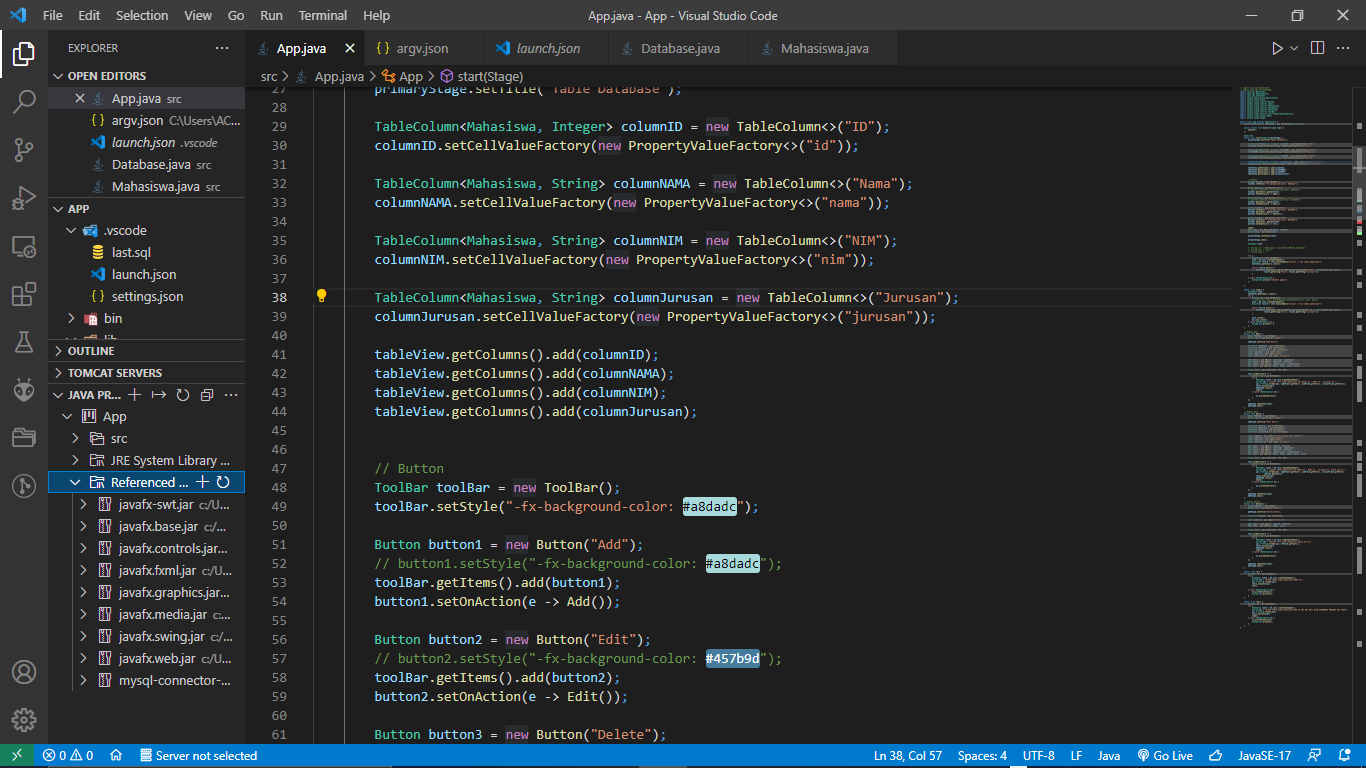


Setelah berhasil membuat Java Project seperti pada gambar:



1. **Menambahkan Referenced Libraries dan Mysql-Connector**

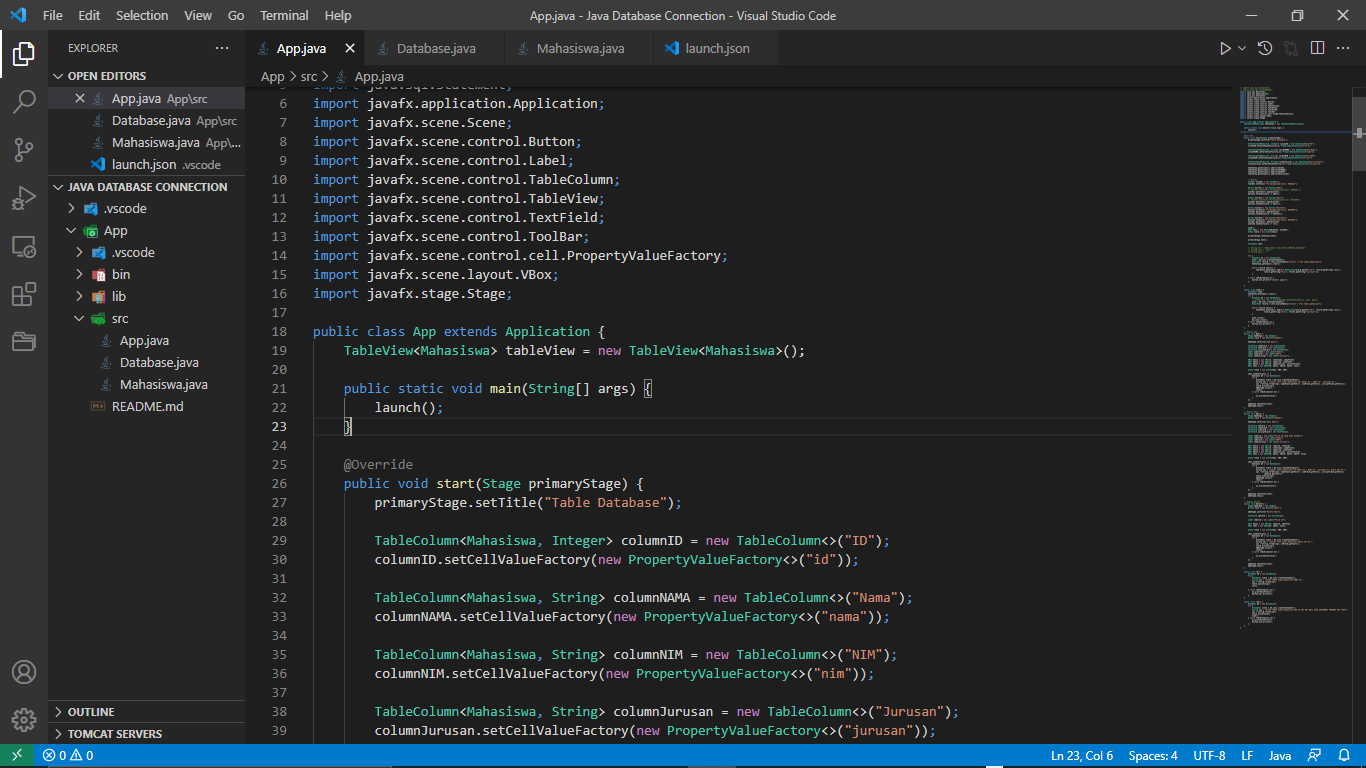
Selanjutnya menambahkan JavaFx dan Mysql-Connector sebagai libraries pada java project seperti pada gambar:



1. Setelah menambahkan Libraries, selanjutnya menambahkan perintah pada Launch JSON sebagai module koneksi path agar java project terhubung dengan javafx :

"vmArgs": "--module-path C:/Users/ACER/Downloads/javafx-sdk-17.0.1/lib --add-modules javafx.controls",

1. Langkah yang kedua adalah Membuat Beberapa File App.java, Database.java, dan Mahasiswa.java seperti pada gambar berikut:



Kemudian membuat beberapa syntax untuk membuat *Grafical User Interface.*

Pada kasus ini saya telah membuat sebuah object Interface berupa Add, Edit, dan Delete pada suatu database db\_akademik dalam tabel\_mahasiswa. Berikut ini merupakan program baris code yang disiapkan untuk membangun sebuah Grafical User Interface akademik diantaranya:

1. File **Database.java**

File ini berfungsi sebagai koneksi dari database ke program java

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

public class Database {

    private final String JDBC\_DRIVER = "com.mysql.cj.jdbc.Driver";

    private final String DB\_URL = "jdbc:mysql://localhost:3306/db\_akademik";

    private final String USER = "root";

    private final String PASS = "";

    Connection conn;

    Statement stmt;

    ResultSet rs;

    public Database() {

        try {

            // register driver yang akan dipakai

            Class.forName(JDBC\_DRIVER);

            // buat koneksi ke database

            conn = DriverManager.getConnection(DB\_URL, USER, PASS);

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

1. File **Mahasiswa.java**

File ini berfungsi sebagai declaration struktur database yang diterapkan di javafx yang nantinya akan digunakan dalam file App.java

public class Mahasiswa {

    private int id;

    private String nama = null;

    private String nim = null;

    private String jurusan = null;

public Mahasiswa(int inputId, String inputNama, String inputNim,

String inputJurusan) {

        this.id = inputId;

        this.nama = inputNama;

        this.nim = inputNim;

        this.jurusan = inputJurusan;

    }

    public int getId() {

        return id;

    }

    public String getNama() {

        return nama;

    }

    public String getNim() {

        return nim;

    }

    public String getJurusan() {

        return jurusan;

    }

}

1. File **App.java**

File ini berfungsi sebagai Running apps program java Project. Berikut merupakan langkah menyusun code diantaranya:

1. Menambahkan Import pada file App.java

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.ToolBar;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

Fungsi dari Import ini adalah sebagai Suatu Perintah Untuk Memasukan suatu Method dalam Bahasa Pemrograman [Java](https://www.java.com/) sehingga perintah tersebut dapat Aktif dan digunakan atau berfungsi.

1. Selanjutnya Menambahkan public class dan run java

public class App extends Application {

    TableView<Mahasiswa> tableView = new TableView<Mahasiswa>();

    public static void main(String[] args) {

        launch();

    }

……

}

digunakan untuk menyatakan bahwa kelas/method/attribute tersebut dapat diakses oleh kelas lain dimanapun.

1. Didalam class App tersebut kemudian tambahkan Code berikut:

@Override

    public void start(Stage primaryStage) {

        primaryStage.setTitle("Table Database");

        TableColumn<Mahasiswa, Integer> columnID = new

TableColumn<>("ID");

        columnID.setCellValueFactory(new

PropertyValueFactory<>("id"));

        TableColumn<Mahasiswa, String> columnNAMA = new

TableColumn<>("Nama");

        columnNAMA.setCellValueFactory(new

PropertyValueFactory<>("nama"));

        TableColumn<Mahasiswa, String> columnNIM = new

TableColumn<>("NIM");

        columnNIM.setCellValueFactory(new

PropertyValueFactory<>("nim"));

        TableColumn<Mahasiswa, String> columnJurusan = new

TableColumn<>("Jurusan");

        columnJurusan.setCellValueFactory(new

PropertyValueFactory<>("jurusan"));

        tableView.getColumns().add(columnID);

        tableView.getColumns().add(columnNAMA);

        tableView.getColumns().add(columnNIM);

        tableView.getColumns().add(columnJurusan);

Code ini berfungsi untuk membuat tampilan database melalui javafx.

1. Lalu menambahkan beberapa button untuk eksekusi database seperti menambahkan data (add), mengubah data(Edit), dan menghapus data(Delete), serta refresh data untuk menampilkan data yang lebih fresh dan rapi.

// Button

ToolBar toolBar = new ToolBar();

toolBar.setStyle("-fx-background-color: #a8dadc");

Button button1 = new Button("Add");

toolBar.getItems().add(button1);

button1.setOnAction(e -> Add());

Button button2 = new Button("Edit");

// button2.setStyle("-fx-background-color: #457b9d");

toolBar.getItems().add(button2);

button2.setOnAction(e -> Edit());

Button button3 = new Button("Delete");

button3.setStyle("-fx-background-color: #e63946");

toolBar.getItems().add(button3);

button3.setOnAction(e -> Delete());

Button button4 = new Button("Refresh");

button4.setStyle("-fx-background-color: #8aea92");

toolBar.getItems().add(button4);

button4.setOnAction(e -> re());

load();

VBox vbox = new VBox(tableView, toolBar);

Scene scene = new Scene(vbox);

primaryStage.setScene(scene);

primaryStage.show();

Statement stmt;

1. Kemudian Menambahkan code koneksi database untuk memastikan koneksi database berhasil . Serta Method Public void load sebagai Tableview untuk menampilkan database yang berhasil terkoneksi

try {

            Database db = new Database();

            stmt = db.conn.createStatement();

            ResultSet record = stmt.executeQuery("select \* from

tabel\_mahasiswa");

            tableView.getItems().clear();

            while (record.next()) {

                tableView.getItems().add(new

Mahasiswa(record.getInt("id"),

record.getString("nama"),

            record.getString("nim"),

record.getString("jurusan")));

            }

        } catch (SQLException e) {

            System.out.println("koneksi gagal");

        }

    }

    public void load() {

        Statement stmt;

        tableView.getItems().clear();

        try {

            Database db = new Database();

            stmt = db.conn.createStatement();

            ResultSet record = stmt.executeQuery("select \* from

tabel\_mahasiswa");

            while (record.next()) {

                tableView.getItems().add(new

Mahasiswa(record.getInt("id"),

record.getString("nama"),

                  record.getString("nim"),

record.getString("jurusan")));

            }

            stmt.close();

            db.conn.close();

        } catch (SQLException e) {

            System.out.println("");

        }

    }

1. Kemudian Menambahkan setiap method function add, edit, delete dan refresh seperti pada code berikut :

// Bagian add

    public void Add() {

        Stage addStage = new Stage();

        Button save = new Button("Simpan");

        addStage.setTitle("Add Data");

        TextField namaField = new TextField();

        TextField nimField = new TextField();

        TextField jurusanField = new TextField();

        Label labelnama = new Label("Nama");

        Label labelnim = new Label("NIM");

        Label labeljurusan = new Label("Jurusan");

        VBox hbox1 = new VBox(5, labelnama, namaField);

        VBox hbox2 = new VBox(5, labelnim, nimField);

        VBox hbox3 = new VBox(5, labeljurusan, jurusanField);

        VBox vbox = new VBox(20, hbox1, hbox2, hbox3, save);

        Scene scene = new Scene(vbox, 500, 500);

        save.setOnAction(e -> {

            Database db = new Database();

            try {

                Statement state = db.conn.createStatement();

                String sql = "insert into tabel\_mahasiswa set

Nama='%s', NIM='%s', Jurusan='%s'";

                sql = String.format(sql, namaField.getText(), nimField.getText(), jurusanField.getText());

                state.execute(sql);

                addStage.close();

                load();

            } catch (SQLException e1) {

                e1.printStackTrace();

            }

        });

        addStage.setScene(scene);

        addStage.show();

    }

    // Bagian Edit

    public void Edit() {

        Stage addStage = new Stage();

        Button save = new Button("Simpan");

        addStage.setTitle("Edit Data");

        TextField idField = new TextField();

        TextField namaField = new TextField();

        TextField nimField = new TextField();

        TextField jurusanField = new TextField();

        Label labelid = new Label("Pilih ID yang akan diubah");

        Label labelnama = new Label("Nama");

        Label labelnim = new Label("NIM");

        Label labeljurusan = new Label("Jurusan");

        VBox hbox1 = new VBox(5, labelid, idField);

        VBox hbox2 = new VBox(5, labelnama, namaField);

        VBox hbox3 = new VBox(5, labelnim, nimField);

        VBox hbox4 = new VBox(5, labeljurusan, jurusanField);

        VBox vbox = new VBox(20, hbox1, hbox2, hbox3, hbox4,save);

        Scene scene = new Scene(vbox, 500, 500);

        save.setOnAction(e -> {

            Database db = new Database();

            try {

                Statement state = db.conn.createStatement();

                String sql = "update tabel\_mahasiswa set Nama='%s', NIM='%s', Jurusan='%s' where ID='%s'";

                sql = String.format(sql, namaField.getText(), nimField.getText(), jurusanField.getText(),

                        idField.getText());

                state.execute(sql);

                addStage.close();

                load();

            } catch (SQLException e1) {

                e1.printStackTrace();

            }

        });

        addStage.setScene(scene);

        addStage.show();

    }

    // Bagian Delete

    public void Delete() {

        Stage addStage = new Stage();

        Button save = new Button("Hapus");

        addStage.setTitle("Delete Data");

        TextField idField = new TextField();

        Label labelid = new Label("Pilih Id");

        VBox hbox1 = new VBox(5, labelid, idField);

        VBox vbox = new VBox(20, hbox1, save);

        Scene scene = new Scene(vbox, 500, 500);

        save.setOnAction(e -> {

            Database db = new Database();

            try {

                Statement state = db.conn.createStatement();

                String sql = "delete from tabel\_mahasiswa where id='%s'";

                sql = String.format(sql, idField.getText());

                state.execute(sql);

                addStage.close();

                load();

            } catch (SQLException e1) {

                e1.printStackTrace();

            }

        });

        addStage.setScene(scene);

        addStage.show();

    }

    public void re() {

        Database db = new Database();

        try {

            Statement state = db.conn.createStatement();

            String sql = "ALTER TABLE tabel\_mahasiswa DROP id";

            sql = String.format(sql);

            state.execute(sql);

            re2();

        } catch (SQLException e1) {

            e1.printStackTrace();

            System.out.println();

        }

    }

    public void re2() {

        Database db = new Database();

        try {

            Statement state = db.conn.createStatement();

            String sql = "ALTER TABLE tabel\_mahasiswa ADD id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY FIRST";

            sql = String.format(sql);

            state.execute(sql);

            load();

        } catch (SQLException e1) {

            e1.printStackTrace();

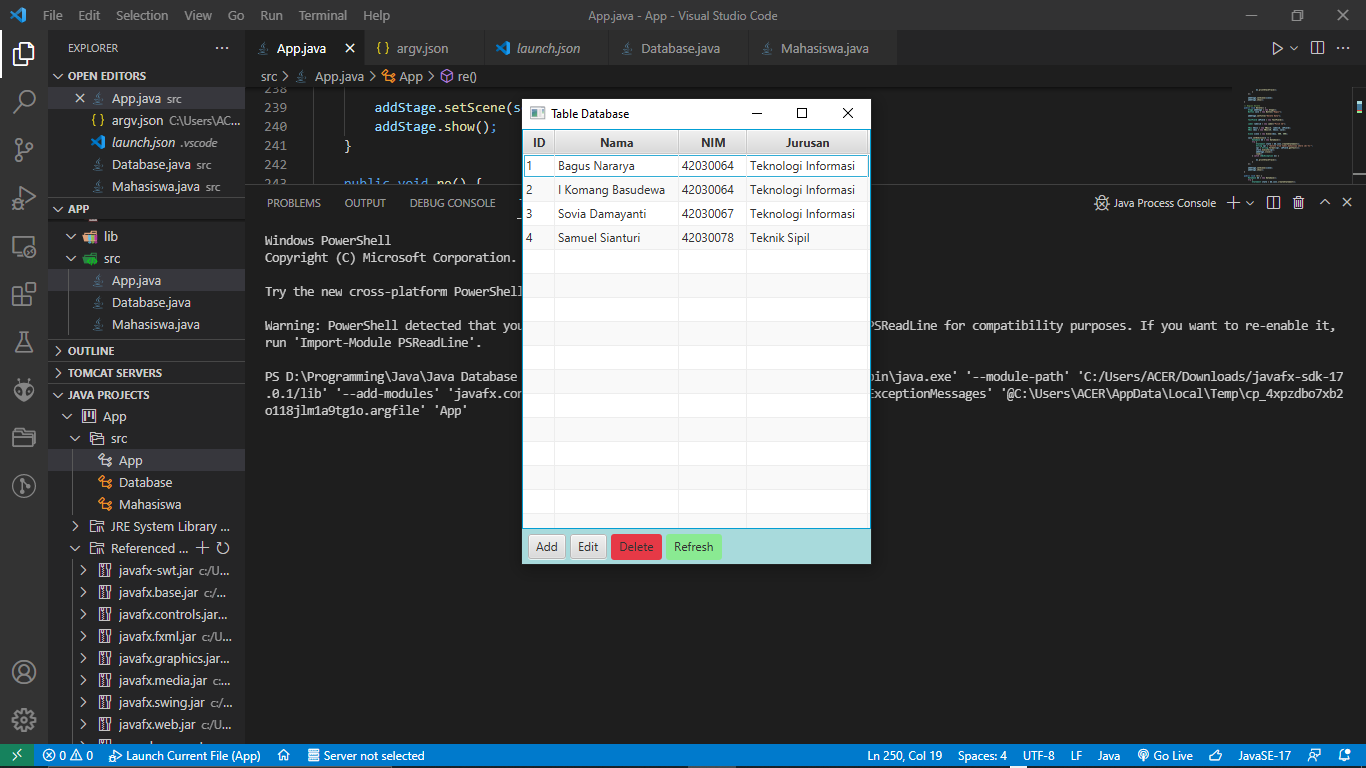
            System.out.println();

        }

    }

}

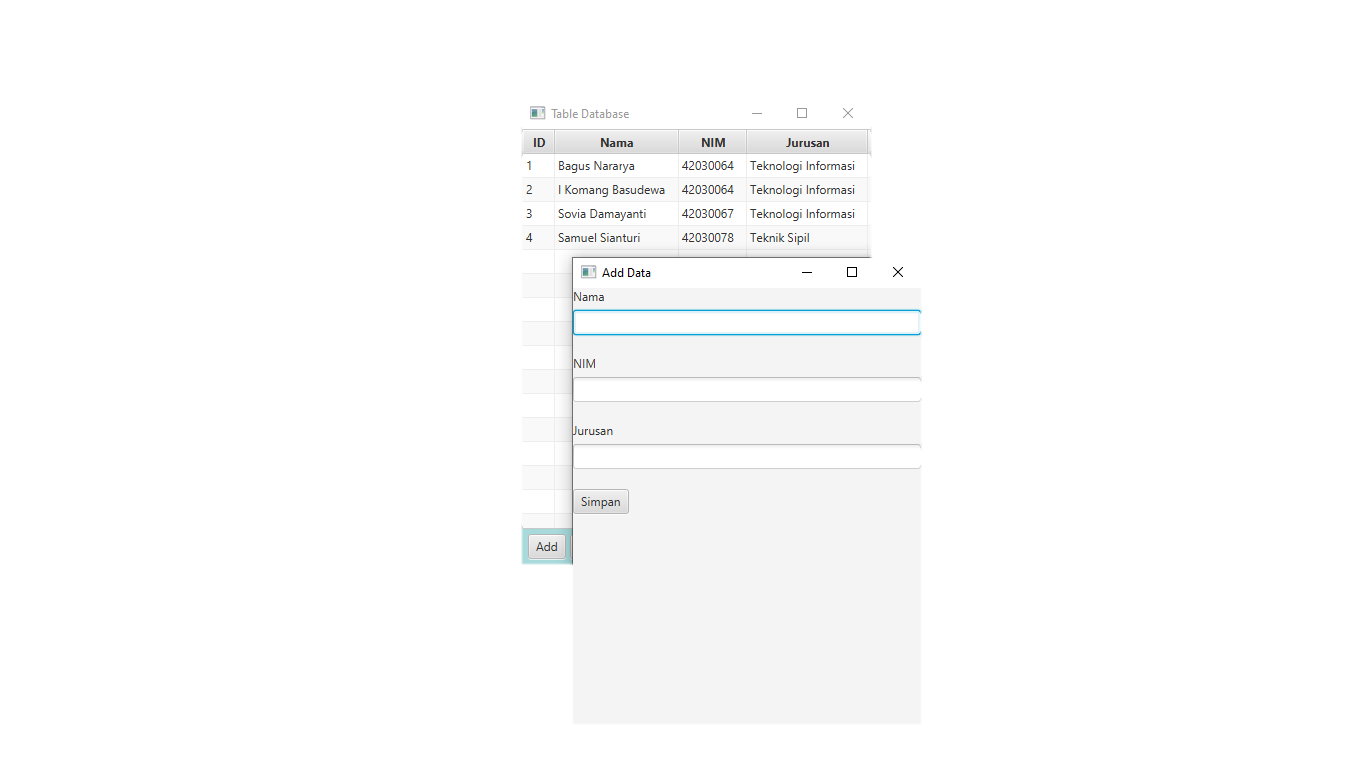
1. Setelah semua Syntax berhasil diterapkan maka dapat dilakukan eksekusi java project atau run java. Berikut adalah hasilnya :



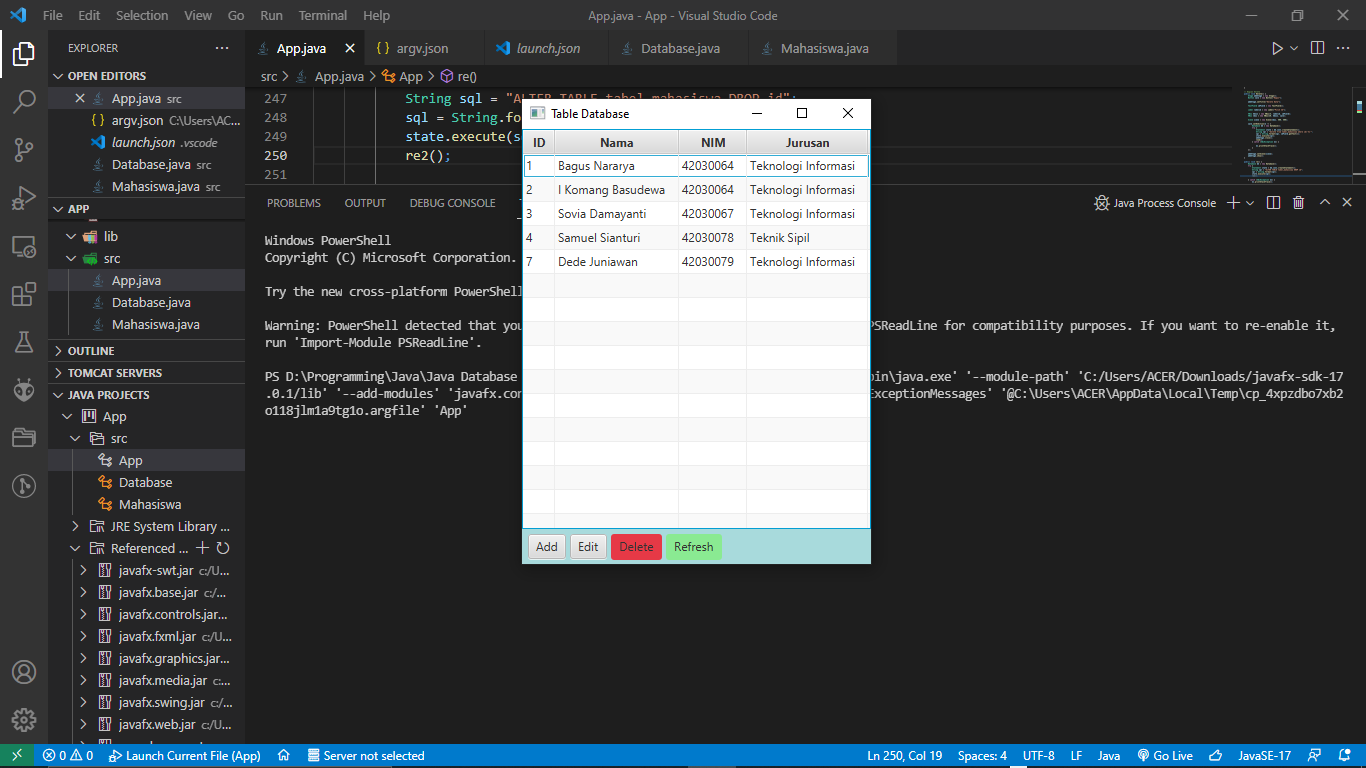
Pada menu di bagian bawah anda dapat menambahkan data, mengedit data, menghapus data, maupun menyusun secara fresh atau rapi dengan button refresh. Berikut hasilnya :

1. Add

Klik button Add, lalu akan muncul Add data. Anda dapat menambahkan data melalu textfield yang tersedia



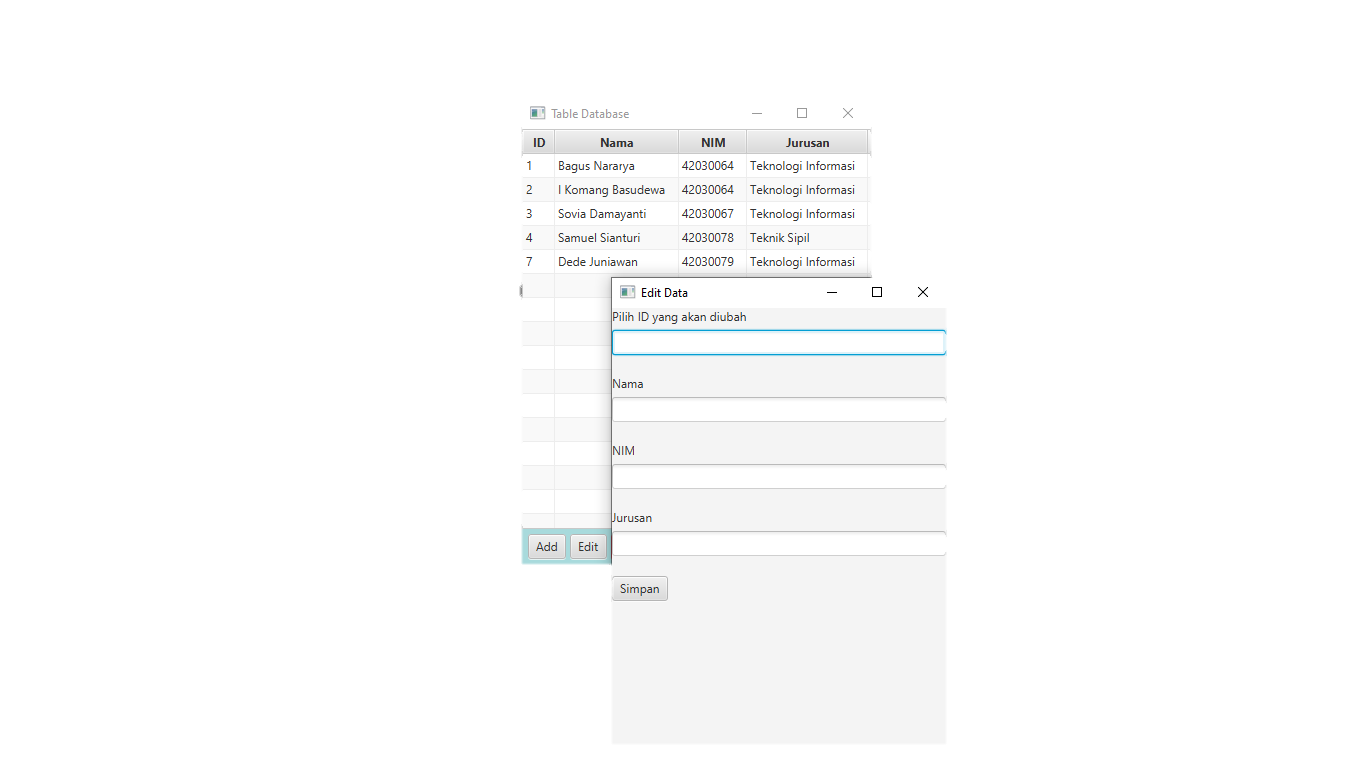
Jika sudah mengisi setiap teksfield, klik simpan lalu lihat hasilnya:



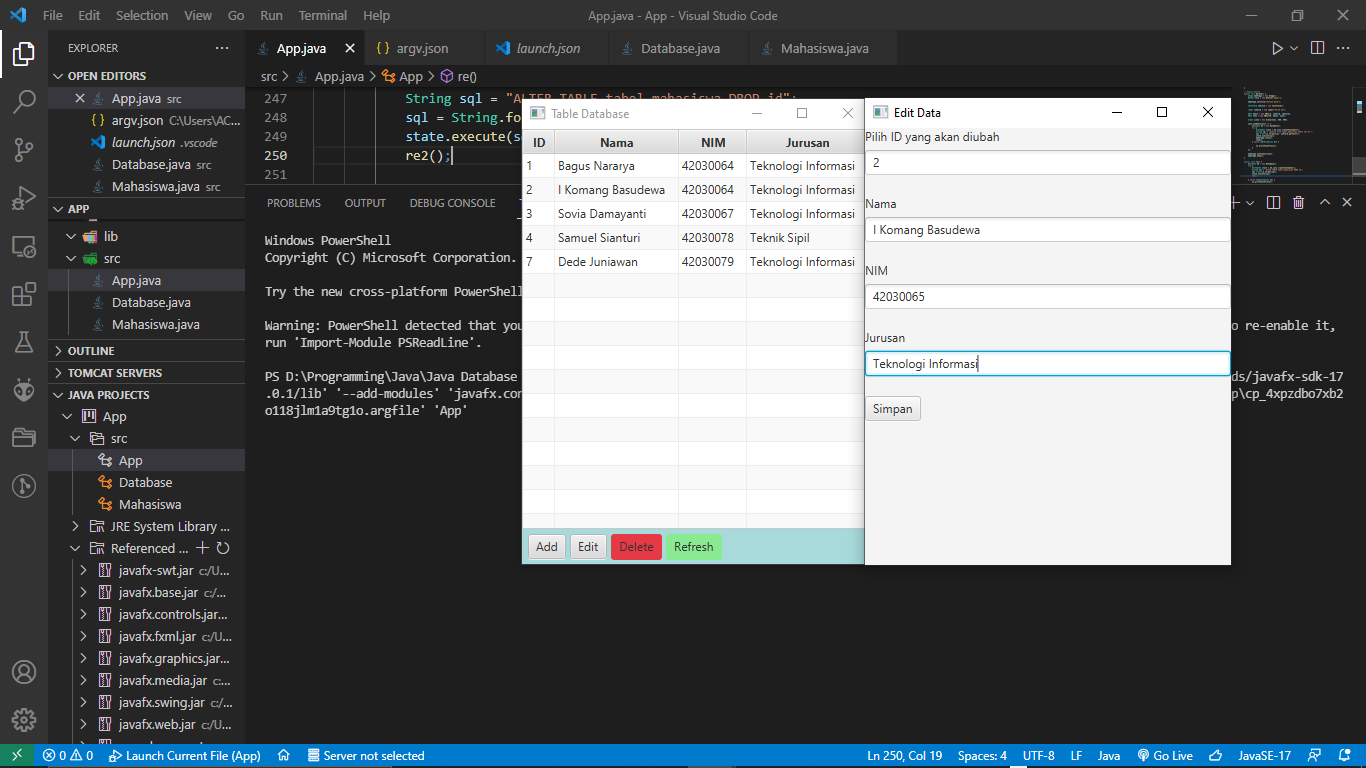
1 row data berhasil ditambahkan.

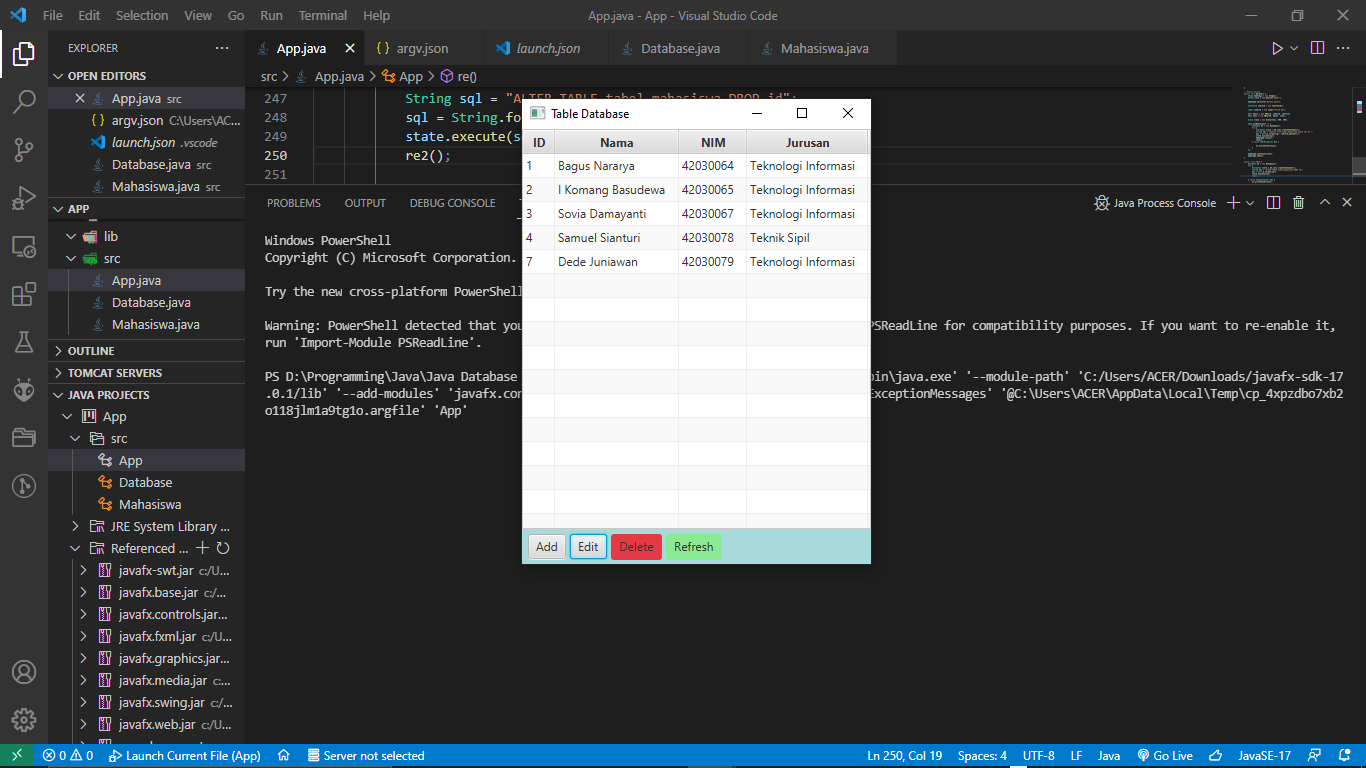
1. Edit

Bagian ini akan mengedit 1 baris data dari ID yang dipilih, langkahnya dengan klik edit, lalu akan muncul field Edit Data.



Terlihat bahwa pada ID ke 2 dan 1 memiliki NIM yang sama, disini saya akan coba merubah ID ke 2 mengganti NIM “42030064” menjadi “42030065” berikut Prosesnya:

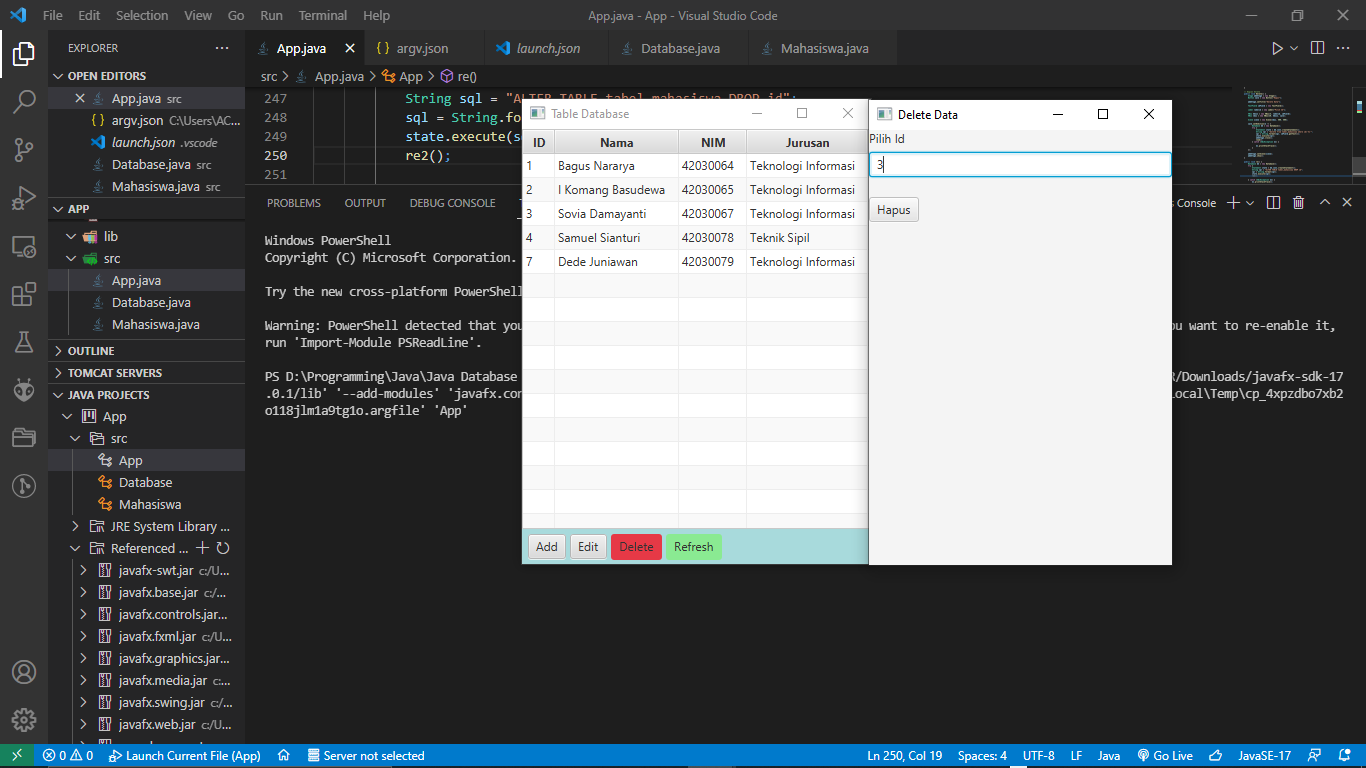




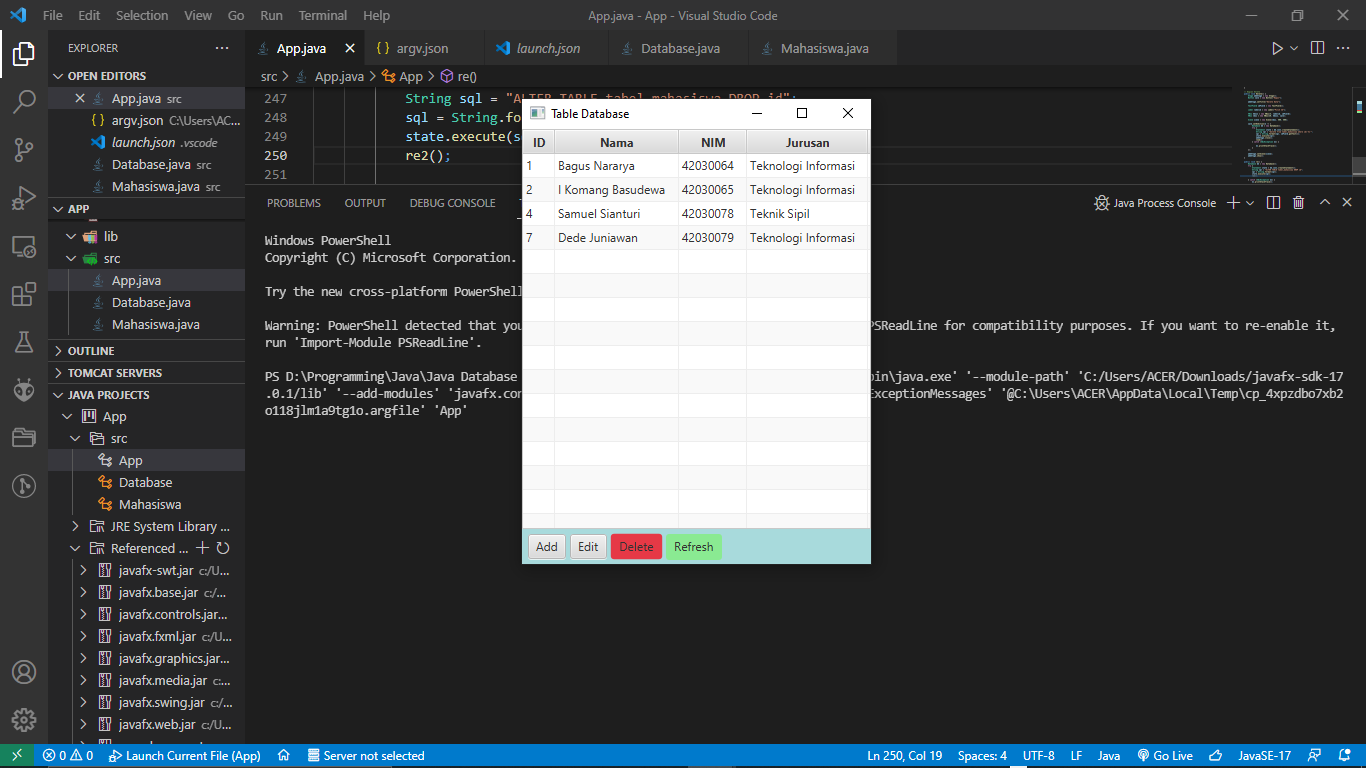
1 baris berhasil di edit.

1. Delete

Klik Button merah untuk Delete lalu akan muncul tampilan Delete Data seperti pada gambar:



Disini saya akan menghapus salah 1 baris database tersebut yaitu pada ID ke 3, berikut adalah hasilnya:

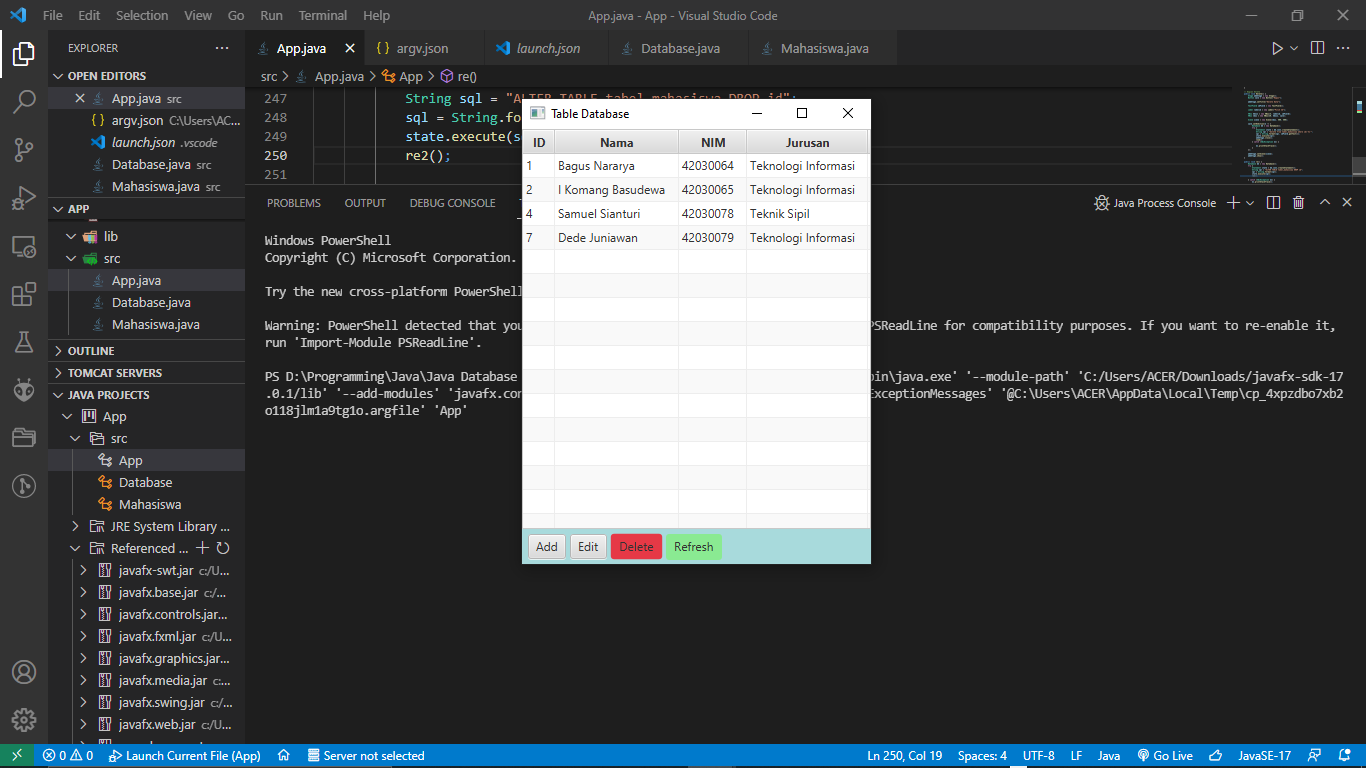


Terlihat bahwa ID ke 3 berhasil dihapus.

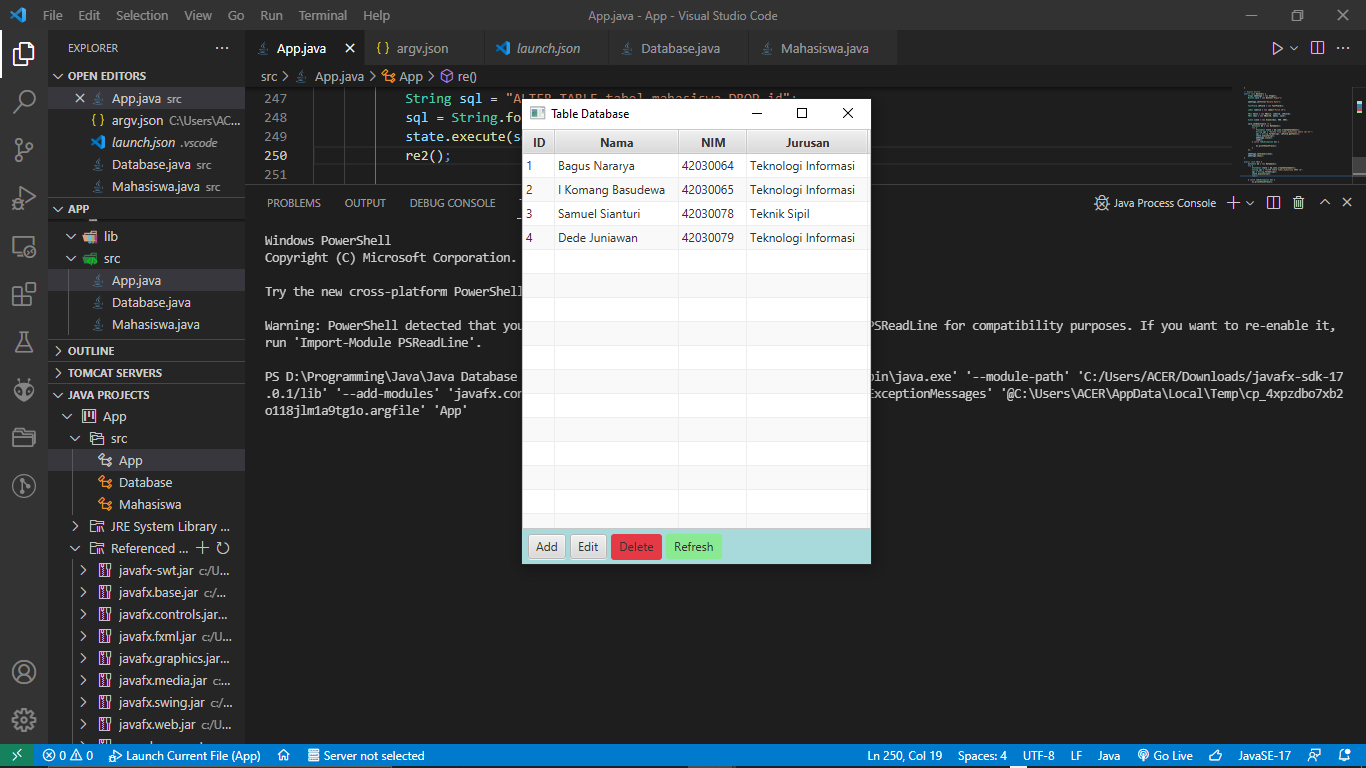
1. Refresh

Bagian ini adalah button refresh yang berfungsi merefresh data menjadi lebih kompleks dan lebih rapi.

Klik Button Refresh lalu, data yang tadinnya tidak terstruktur seperti pada gambar ini:



Akan tampil menjadi seperti pada gambar berikut:



Data berhasil di Refresh.