

UJIAN AKHIR SEMESTER

PEMROGRAMAN BERORIENTASI OBJEK GRAPHICAL USER INTERFACE (GUI)

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

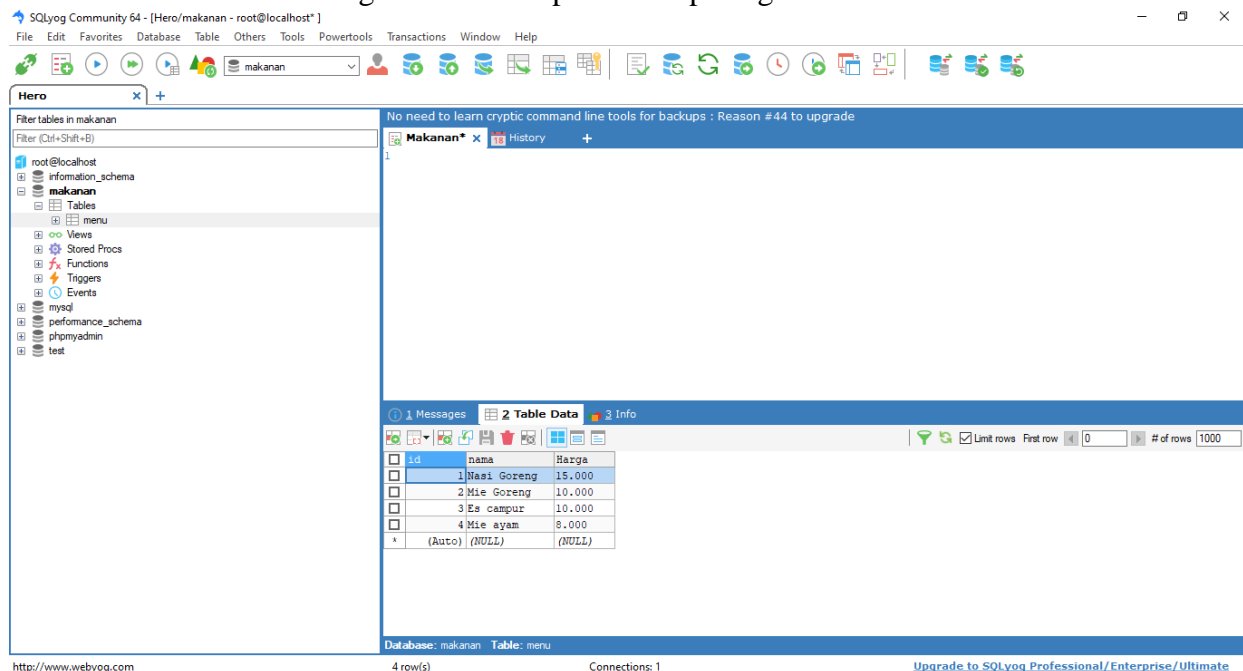


OLEH :

Nama : Agus Yusuf Noviyanto
NIM : (42030043)

**JURUSAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PENDIDIKAN NASIONAL (UNDIKNAS)
2021**

1. Langkah pertama yang harus dilakukan yaitu membuat database, disini saya menggunakan database terminal dengan 1 tabel. Dapat dilihat pada gambar dibawah.



2. Langkah selanjutnya yaitu install mysql-connector-java dan javafx pada google. Ini digunakan untuk menghubungkan database pada java dan javafx (untuk libraries javafx).
3. Selanjutnya beralih ke vscode (java) untuk mengimport data:
 - Pertama yang harus dilakukan yaitu membuat 3 file. File App.java, Database.java, dan makanan.java, kemudian masukan kode pada file App.java dengan query:

```
import javafx.application.Application;
import javafx.scene.Scene;
import java.sql.SQLException;
import java.sql.Statement;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import java.sql.ResultSet;
import javafx.scene.control.*;
import javafx.scene.control.Button;
import java.io.FileNotFoundException;

public class App extends Application {
    TableView<makanan> tableView = new TableView<makanan>();

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

    @Override
    public void start(Stage primaryStage) throws FileNotFoundException {
        primaryStage.setTitle("UAS OOP");
        TableColumn<makanan, String> columnID = new TableColumn<>("No");
        columnID.setCellValueFactory(new PropertyValueFactory<>("Id"));
    }
}
```

```

TableColumn<makanan, String> columnNama = new TableColumn<>("Menu");
columnNama.setCellValueFactory(new PropertyValueFactory<>("Nama"));

TableColumn<makanan, String> columnHarga = new TableColumn<>("Harga");
columnHarga.setCellValueFactory(new PropertyValueFactory<>("Harga"));

tableView.getColumns().add(columnID);
tableView.getColumns().add(columnNama);
tableView.getColumns().add(columnHarga);

ToolBar toolBar = new ToolBar();

Button button1 = new Button("Add Menu");
toolBar.getItems().add(button1);
button1.setOnAction(e -> add());

Button button2 = new Button("Delete");
toolBar.getItems().add(button2);
button2.setOnAction(e -> delete());

Button button3 = new Button("EDIT");
toolBar.getItems().add(button3);
button3.setOnAction(e -> edit());

Button button4 = new Button("Refresh");
toolBar.getItems().add(button4);
button4.setOnAction(e -> re());

VBox vbox = new VBox(tableView, toolBar);

Scene scene = new Scene(vbox);

primaryStage.setScene(scene);

primaryStage.show();
load()

```

Jika ingin di eksekusi maka file inilah (App.java) yang akan di run/eksekusi. Dalam App.java akan berisi kode-kode untuk menampilkan data dari database, menampilkan dalam bentuk tabel, menambahkan (add), mengedit, serta menghapus.

Lalu langkah selanjutnya beralih ke file Database.java dengan memasukan kode sebagai berikut:

```
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/makanan";
    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();
        }
    }
}
```

- Selanjutnya membuat class untuk table makanan.java dengan memasukan kode sebagai berikut:

```
4. public class makanan {
5.
6.     private String nama = null;
7.     private int id;
8.     private String harga = null;
9.
10.    public makanan(int id, String nama, String harga) {
11.        this.nama = nama;
12.        this.id = id;
13.        this.harga = harga;
14.    }
15.
16.    public String getNama() {
17.        return nama;
18.    }
```

```

19.
20.     public int getId() {
21.         return id;
22.     }
23.
24.     public String getHarga() {
25.         return harga;
26.     }
27. }

```

28. Pada file App.java saya membuat kode untuk menampilkan data dari database, menampilkan dalam bentuk tabel, menambahkan (add), mengedit, serta menghapus.
- Pertama, kode untuk menampilkan data dari database dan menampilkan dalam bentuk tabel sebagai berikut:

```

29. Statement stmt;
30.     try {
31.         Database db = new Database();
32.         stmt = db.conn.createStatement();
33.         ResultSet rs = stmt.executeQuery("select * from menu");
34.         tableView.getItems().clear();
35.         // tampilkan hasil query
36.         while (rs.next()) {
37.             tableView.getItems().add(new makanan(rs.getInt("Id"),
rs.getString("Nama"), rs.getString("Harga")));
38.         }
39.
40.         stmt.close();
41.         db.conn.close();
42.     } catch (SQLException e) {
43.         e.printStackTrace();
44.     }
45. }

```

Pada program ini saya menambahkan method untuk memanggil table kembali

```

: public void load() {
    Statement stmt;
    tableView.getItems().clear();
    try {
        Database db = new Database();
        stmt = db.conn.createStatement();
        ResultSet rs = stmt.executeQuery("select * from menu");
        while (rs.next()) {
            tableView.getItems().addAll(new makanan(rs.getInt("Id"),
rs.getString("Nama"), rs.getString("Harga")));
        }
        stmt.close();
        db.conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```
}
}
```

- Selanjutnya file App.java ini dapat dilangsung di run/eksekusi, nanti akan ditampilkan tabel seperti pada gambar dibawah.

No	Menu	Harga
1	Nasi Goreng	15.000
2	Mie Goreng	10.000
3	Es campur	10.000
4	Mie ayam	8.000
5	Bakso	15.000

Add Menu Delete EDIT Refresh

- Kedua, untuk menambahkan (add) kode seperti berikut:

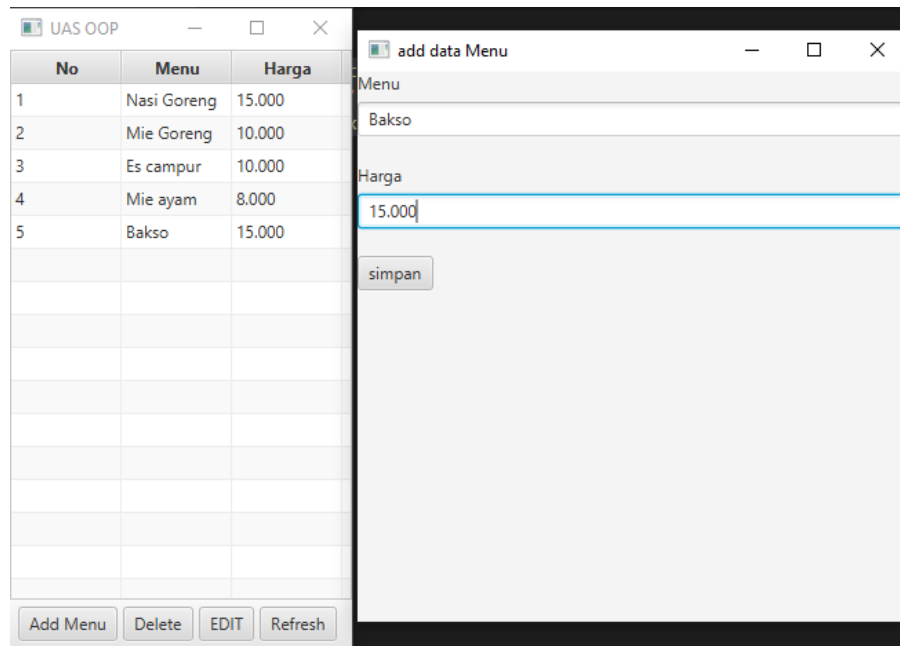
```
46. public void add() {
47.     Stage addStage = new Stage();
48.     Button save = new Button("simpan");
49.
50.     addStage.setTitle("add data Menu");
51.
52.     TextField namaField = new TextField();
53.     TextField hargaField = new TextField();
54.     Label labelNama = new Label("Menu");
55.     Label labelharga = new Label("Harga");
56.
57.     VBox hbox1 = new VBox(5, labelNama, namaField);
58.     VBox hbox2 = new VBox(5, labelharga, hargaField);
59.     VBox vbox = new VBox(20, hbox1, hbox2, save);
60.
61.     Scene scene = new Scene(vbox, 400, 400);
62.
63.     save.setOnAction(e -> {
64.         Database db = new Database();
65.         try {
66.             Statement state = db.conn.createStatement();
67.             String sql = "insert into menu SET nama='%s', harga='%s'";
```

```

68.         sql = String.format(sql, namaField.getText(),
        hargaField.getText());
69.         state.execute(sql);
70.         addStage.close();
71.         load();
72.     } catch (SQLException e1) {
73.
74.         e1.printStackTrace();
75.     }
76. });
77.
78. addStage.setScene(scene);
79. addStage.show();
80. }

```

- Hasilnya dapat dilihat pada gambar:



Selanjutnya, yaitu menghapus data dengan kode berikut:

```

public void delete() {
    Stage addStage = new Stage();
    Button save = new Button("DELETE");

    addStage.setTitle("Delete Data");

    TextField noField = new TextField();
    Label labelNo = new Label("Menu");

    VBox hbox1 = new VBox(5, labelNo, noField);
    VBox vbox = new VBox(20, hbox1, save);

    Scene scene = new Scene(vbox, 400, 400);

    save.setOnAction(e -> {

```

```

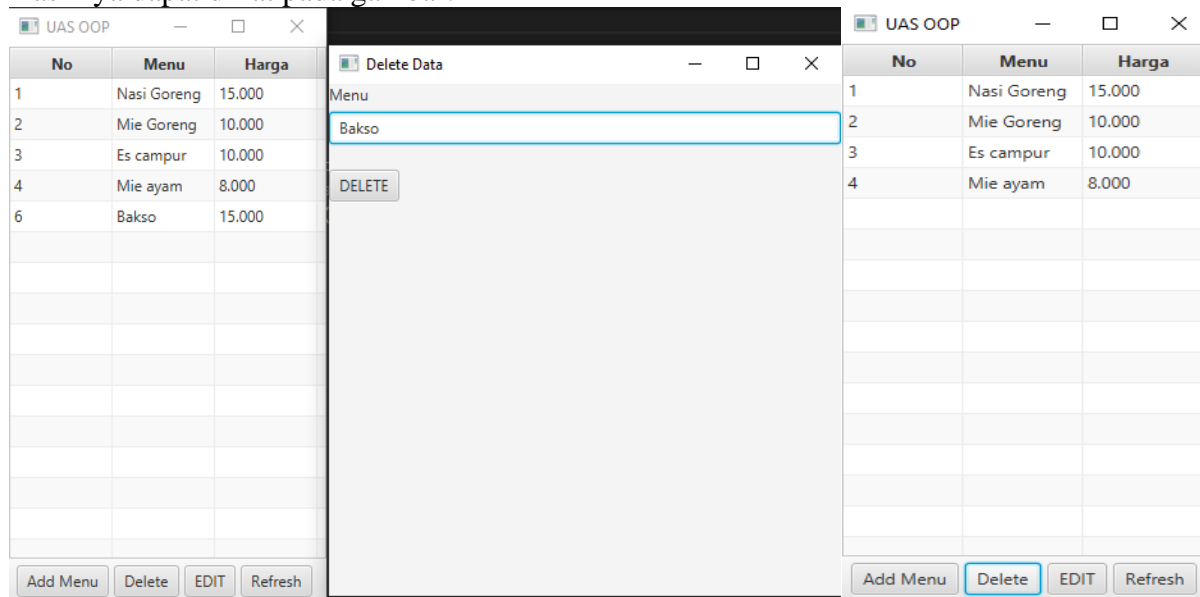
        Database db = new Database();
        try {
            Statement state = db.conn.createStatement();
            String sql = "delete from menu WHERE nama='%s'";
            sql = String.format(sql, noField.getText());
            state.execute(sql);
            addStage.close();
            load();
        } catch (SQLException e1) {

            e1.printStackTrace();
            System.out.println();
        }
    });

    addStage.setScene(scene);
    addStage.show();
}

```

Hasilnya dapat dilihat pada gambar:



langkah terakhir kode untuk mengedit data sebagai berikut:

```

public void edit() {
    Stage addStage = new Stage();
    Button save = new Button("Simpan");

    addStage.setTitle("Edit Menu");

    TextField namaField = new TextField();
    TextField hargaField = new TextField();
    Label labelNama = new Label("Menu");
    Label labelharga = new Label("Harga");

    VBox hbox1 = new VBox(5, labelNama, namaField);
    VBox hbox2 = new VBox(5, labelharga, hargaField);
    VBox vbox = new VBox(20, hbox1, hbox2, save);
}

```



```

Scene scene = new Scene(vbox, 400, 400);

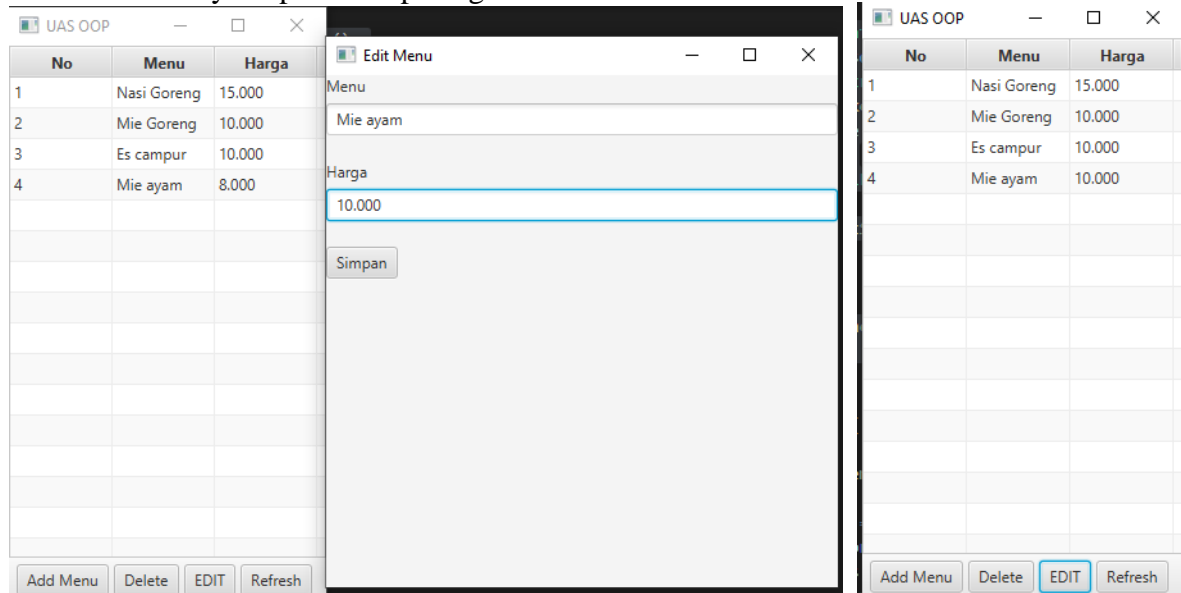
save.setOnAction(e -> {
    Database db = new Database();
    try {
        Statement state = db.conn.createStatement();
        String sql = "UPDATE menu SET Harga = '%s' WHERE nama='%s'";
        sql = String.format(sql, hargaField.getText(),
namaField.getText());
        state.execute(sql);
        addStage.close();
        load();
    } catch (SQLException e1) {

        e1.printStackTrace();
    }
});

addStage.setScene(scene);
addStage.show();
}

```

- Hasilnya dapat dilihat pada gambar:



```
public void re() {
    Database db = new Database();
    try {
        Statement state = db.conn.createStatement();
        String sql = "ALTER TABLE menu 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 menu 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();
    }
}
```

[illegible]