

UJIAN AKHIR SEMESTER

PEMROGRAMAN BERORIENTASI OBJEK GRAPHICAL USER INTERFACE (GUI)

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



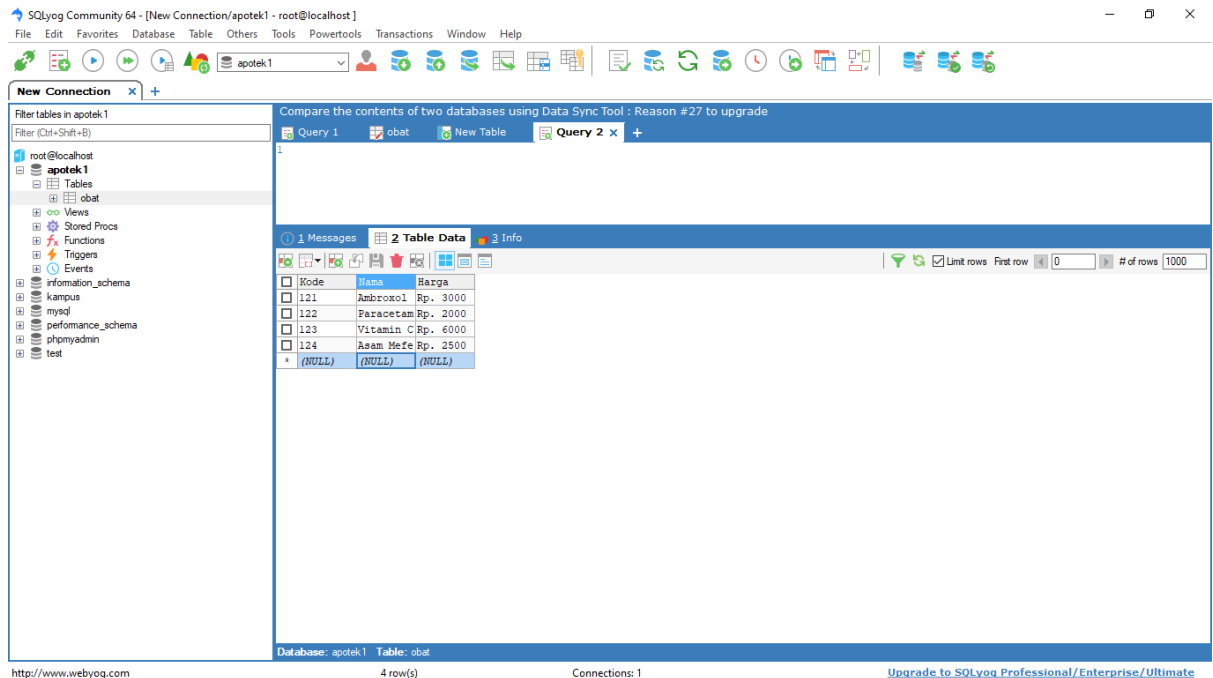
Oleh:

Nama : Gusti Ayu Mas Tara Cinantya
NIM : (42030056)

**JURUSAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PENDIDIKAN NASIONAL (UNDIKNAS)
2022**

PEMROGRAMAN BERORIENTASI OBJEK GRAPHICAL USER INTERFACE (GUI)

1. Pertama, membuat database terlebih dahulu, disini saya menggunakan mysql untuk membuat database. Nama database saja “apotek1” dan table “obat”



2. Kedua, menginstall mysql-connector-java dan javafx yang dapat di download di Google. Ini digunakan untuk menghubungkan database (mysql) pada java dan javafx (untuk library javafx).
3. Selanjutnya, buka vscode (java) lalu import data:
 - a. Pertama membuat 3 file yaitu App.java, Database.java dan Apotek.java. kemudian masukkan 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<Apotek> tableView = new TableView<Apotek>();
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws FileNotFoundException {
        primaryStage.setTitle("Data Obat");
        TableColumn<Apotek, String> columnKode = new TableColumn<>("Kode");
        columnKode.setCellValueFactory(new PropertyValueFactory<>("Kode"));

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

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

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

       ToolBar toolBar = new ToolBar();

        Button button1 = new Button("Add Data");
        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 App.java yang akan di eksekusi atau run. Dalam App.java akan berisi kode-kode untuk menampilkan data dari database, menampilkan dalam bentuk table, menambah data (add), mengedit data dan menghapus data.

b. Selanjutnya pada file Database.java dengan memasukkan 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/apotek1";  
    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();  
        }  
    }  
}
```

c. Selanjutnya membuat class untuk tabel Apotek.java dengan memasukkan kode sebagai berikut:

```
public class Apotek {  
  
    private String Kode = null;  
    private String Nama = null;  
    private String Harga = null;  
  
    public Apotek(String Kode, String Nama, String Harga) {
```

```

        this.Kode = Kode;
        this>Nama = Nama;
        this.Harga = Harga;
    }

    public String getKode() {
        return Kode;
    }

    public String getNama() {
        return Nama;
    }

    public String getHarga() {
        return Harga;
    }
}

```

- d. Pada file App.java berisikan kode kode untuk menampilkan data dari database, menampilkan dalam bentuk table, menambahkan data (add), mengedit data serta menghapus data.

```

Statement stmt;
try {
    Database db = new Database();
    stmt = db.conn.createStatement();
    ResultSet rs = stmt.executeQuery("select * from obat");
    tableView.getItems().clear();
    // tampilkan hasil query
    while (rs.next()) {
        tableView.getItems().add(new Apotek(rs.getString("Kode"),
rs.getString("Nama"), rs.getString("Harga")));
    }

    stmt.close();
    db.conn.close();
} catch (SQLException e) {
    e.printStackTrace();
}
}

```

- e. Selanjutnya menambahkan method untuk memanggil table kembali:

```

public void load() {
    Statement stmt;
    tableView.getItems().clear();
    try {
        Database db = new Database();

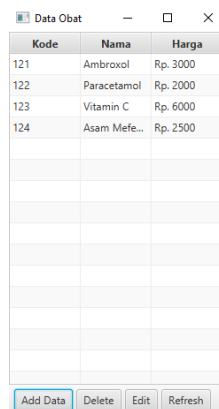
```

```

        stmt = db.conn.createStatement();
        ResultSet rs = stmt.executeQuery("select * from obat");
        while (rs.next()) {
            tableView.getItems().addAll(new
Apotek(rs.getString("Kode"), rs.getString("Nama"), rs.getString("Harga")));
        }
        stmt.close();
        db.conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

f. Selanjutnya mengeksekusi/run App.java dan akan menampilkan:



Kode	Nama	Harga
121	Ambroxol	Rp. 3000
122	Paracetamol	Rp. 2000
123	Vitamin C	Rp. 6000
124	Asam Mefe...	Rp. 2500

Add Data Delete Edit Refresh

g. Selanjutnya adalah menambahkan data (Add Data) dengan kode dan tampilan sebagai berikut:

```

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

    addStage.setTitle("Add Data");

    TextField KodeField = new TextField();
    TextField NamaField = new TextField();
    TextField HargaField = new TextField();

    Label labelKode = new Label("Kode");
    Label labelNama = new Label("Nama");
    Label labelHarga = new Label("Harga");

    VBox hbox1 = new VBox(5, labelKode, KodeField);
    VBox hbox2 = new VBox(5, labelNama, NamaField);
    VBox hbox3 = new VBox(5, labelHarga, HargaField);
    VBox vbox = new VBox(20, hbox1, hbox2, hbox3, save);

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

```



```

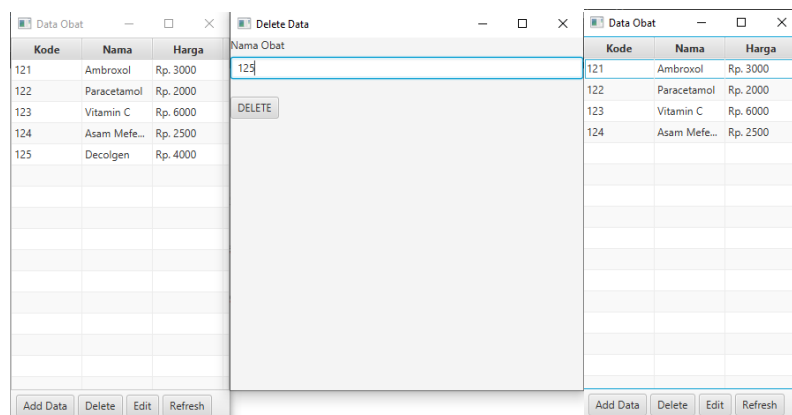
Scene scene = new Scene(vbox, 400, 400);

save.setOnAction(e -> {
    Database db = new Database();
    try {
        Statement state = db.conn.createStatement();
        String sql = "delete from obat 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();
}

```



i. Terakhir adalah mengedit data dengan kode dan tampilan sebagai berikut:

```

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

    addStage.setTitle("Edit Data");

    TextField KodeField = new TextField();
    TextField HargaField = new TextField();
    Label labelKode = new Label("Kode");
    Label labelHarga = new Label("Harga");

    VBox hbox1 = new VBox(5, labelKode, KodeField);
    VBox hbox2 = new VBox(5, labelHarga, HargaField);

```



```

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

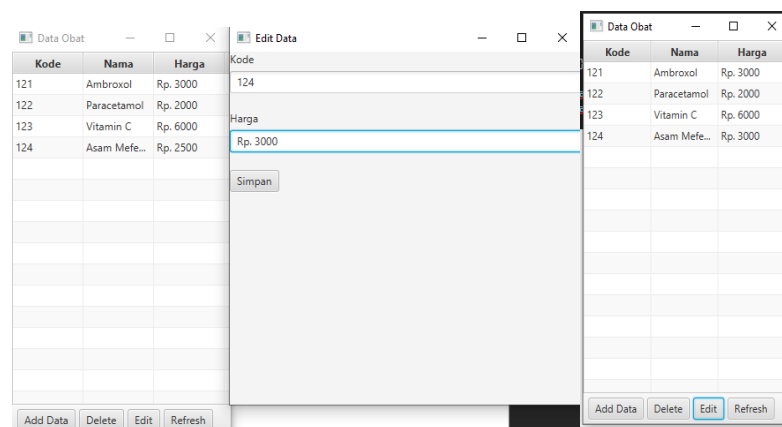
Scene scene = new Scene(vbox, 400, 400);

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

        e1.printStackTrace();
    }
});

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

```



- j. Selanjutnya adalah method untuk merefresh apa yang kita edit pada data, misalnya harga. Disana saya mengubah harga Asam mefenamat dari Rp. 2500 menjadi Rp. 3000. Jika sudah selesai mengedit data (simpan) dan belum berubah, dapat dilakukan refresh sehingga harga akan berubah.

```

public void re() {
    Database db = new Database();
    try {
        Statement state = db.conn.createStatement();
        String sql = "ALTER TABLE obat DROP Kode";

```

```

        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 obat ADD Kode INT NOT NULL PRIMARY
KEY FIRST";
        sql = String.format(sql);
        state.execute(sql);
        load();
    } catch (SQLException e1) {
        e1.printStackTrace();
        System.out.println();
    }
}
}

```