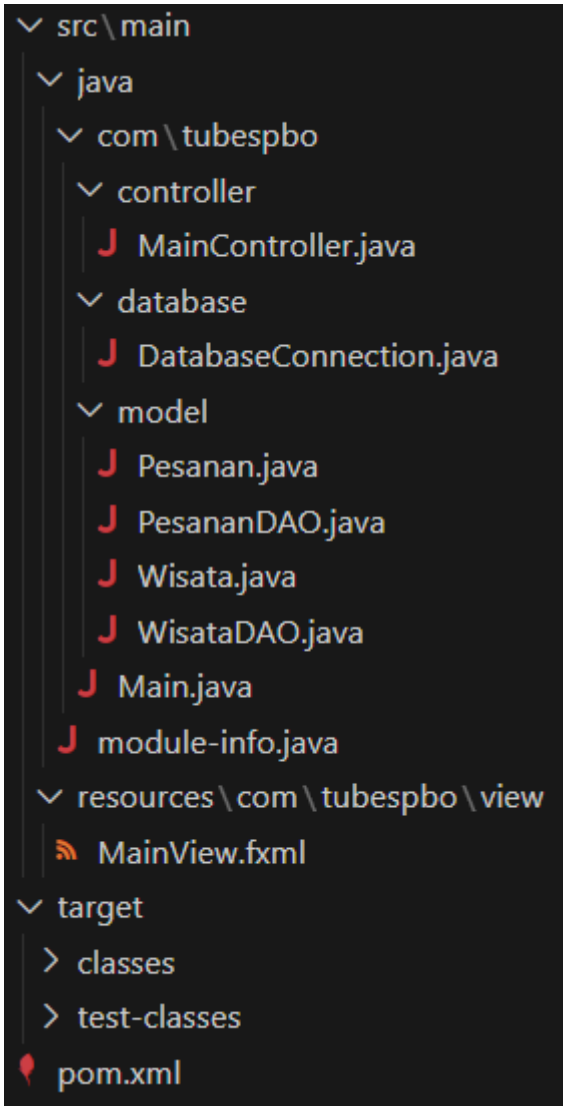
Struktur Folder Proyek



1. **MainController.java**

```
package com.tubespbo.controller;

import javafx.beans.property.SimpleDoubleProperty;
import javafx.beans.property.SimpleIntegerProperty;
import javafx.beans.property.SimpleObjectProperty;
import javafx.beans.property.SimpleStringProperty;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
      import javafx.scene.Parent;
      import javafx.scene.Scene;
      import javafx.scene.control.*;
      import javafx.scene.control.cell.PropertyValueFactory;
      import javafx.stage.Modality;
      import javafx.stage.Stage;
      import com.tubespbo.model.Pesanan;
      import com.tubespbo.model.PesananDAO;
      import com.tubespbo.model.Wisata;
      import com.tubespbo.model.WisataDAO;

      import java.io.IOException;
      import java.io.InputStream;
      import java.sql.SQLException;
      import java.time.LocalDate;
      import java.util.List;

      public class MainController {
          @FXML private ComboBox<Wisata> wisataComboBox;
          @FXML private TextField hargaField;
          @FXML private TextField jumlahField;
          @FXML private DatePicker tanggalPicker;
          @FXML private TableView<Pesanan> pesananTable;

          private WisataDAO wisataDAO;
          private PesananDAO pesananDAO;

          @FXML
          public void initialize() {
              TableColumn<Pesanan, String> namaCol =
   (TableColumn<Pesanan, String>) pesananTable.getColumns().get(0);
              namaCol.setCellValueFactory(cellData -> new
   SimpleStringProperty(cellData.getValue().getWisata().getNama()));

              TableColumn<Pesanan, Double> hargaCol =
   (TableColumn<Pesanan, Double>) pesananTable.getColumns().get(1);
              hargaCol.setCellValueFactory(cellData -> new
   SimpleDoubleProperty(cellData.getValue().getWisata().getHarga()).asObje
   ct());
```

```
44.
45.             TableColumn<Pesanan, Integer> jumlahCol =
    (TableColumn<Pesanan, Integer>) pesananTable.getColumns().get(2);
46.             jumlahCol.setCellValueFactory(cellData -> new
    SimpleIntegerProperty(cellData.getValue().getJumlah()).asObject());
47.
48.             TableColumn<Pesanan, Double> totalHargaCol =
    (TableColumn<Pesanan, Double>) pesananTable.getColumns().get(3);
49.             totalHargaCol.setCellValueFactory(cellData -> new
    SimpleDoubleProperty(cellData.getValue().getTotalHarga()).asObject());
50.
51.             TableColumn<Pesanan, LocalDate> tanggalCol =
    (TableColumn<Pesanan, LocalDate>) pesananTable.getColumns().get(4);
52.             tanggalCol.setCellValueFactory(cellData -> new
    SimpleObjectProperty<>(cellData.getValue().getTanggalKunjungan()));
53.
54.             TableColumn<Pesanan, String> statusCol =
    (TableColumn<Pesanan, String>) pesananTable.getColumns().get(5);
55.             statusCol.setCellValueFactory(cellData -> new
    SimpleStringProperty(cellData.getValue().getStatus()));
56.             try {
57.                 wisataDAO = new WisataDAO();
58.                 pesananDAO = new PesananDAO();
59.
60.                 // Load data wisata ke ComboBox
61.                 List<Wisata> wisataList = wisataDAO.getAllWisata();
62.                 wisataComboBox.getItems().addAll(wisataList);
63.
64.                 // Set listener untuk ComboBox
65.                 wisataComboBox.getSelectionModel().selectedItemProper
    ty().addListener((obs, oldVal, newVal) -> {
66.                     if (newVal != null) {
67.                         hargaField.setText(String.valueOf(newVal.getH
    arga()));
68.                     }
69.                 });
70.
71.                 // Set default tanggal ke hari ini
72.                 tanggalPicker.setValue(LocalDate.now());
73.
74.                 // Load data pesanan ke TableView
75.                 refreshPesananTable();
76.
77.             } catch (SQLException e) {
78.                 showAlert("Database Error", "Gagal memuat data: " +
    e.getMessage());
79.             }
80.         }
```

```
81.
82.         @FXML
83.         private void handleBeli() {
84.             try {
85.                 Wisata selectedWisata =
   wisataComboBox.getSelectionModel().getSelectedItem();
86.                 if (selectedWisata == null) {
87.                     showAlert("Peringatan", "Pilih tempat wisata
   terlebih dahulu");
88.                     return;
89.                 }
90.
91.                 int jumlah;
92.                 try {
93.                     jumlah = Integer.parseInt(jumlahField.getText());
94.                     if (jumlah <= 0) {
95.                         showAlert("Peringatan", "Jumlah tiket harus
   lebih dari 0");
96.                         return;
97.                     }
98.                 } catch (NumberFormatException e) {
99.                     showAlert("Peringatan", "Jumlah tiket harus
   berupa angka");
100.                    return;
101.                }
102.
103.                LocalDate tanggal = tanggalPicker.getValue();
104.                if (tanggal == null) {
105.                    showAlert("Peringatan", "Pilih tanggal
   kunjungan");
106.                    return;
107.                }
108.
109.                Pesanan pesanan = new Pesanan(0, selectedWisata,
   jumlah, tanggal, "pending");
110.                pesananDAO.addPesanan(pesanan);
111.
112.                refreshPesananTable();
113.                clearForm();
114.
115.            } catch (SQLException e) {
116.                showAlert("Database Error", "Gagal menyimpan pesanan:
   " + e.getMessage());
117.            }
118.        }
119.
120.        @FXML
121.        private void handleEdit() {
```

```
122.             Pesanan selectedPesanan =
   pesananTable.getSelectionModel().getSelectedItem();
123.             if (selectedPesanan == null) {
124.                 showAlert("Peringatan", "Pilih pesanan yang akan
   diubah statusnya");
125.                 return;
126.             }
127.
128.             if ("lunas".equals(selectedPesanan.getStatus())) {
129.                 showAlert("Peringatan", "Pesanan sudah lunas");
130.                 return;
131.             }
132.
133.             try {
134.                 pesananDAO.updateStatusPesanan(selectedPesanan.getId(
   ), "lunas");
135.                 refreshPesananTable();
136.             } catch (SQLException e) {
137.                 showAlert("Database Error", "Gagal mengupdate status:
   " + e.getMessage());
138.             }
139.         }
140.
141.         @FXML
142.         private void handleHapus() {
143.             Pesanan selectedPesanan =
   pesananTable.getSelectionModel().getSelectedItem();
144.             if (selectedPesanan == null) {
145.                 showAlert("Peringatan", "Pilih pesanan yang akan
   dihapus");
146.                 return;
147.             }
148.
149.             try {
150.                 pesananDAO.deletePesanan(selectedPesanan.getId());
151.                 refreshPesananTable();
152.             } catch (SQLException e) {
153.                 showAlert("Database Error", "Gagal menghapus pesanan:
   " + e.getMessage());
154.             }
155.         }
156.
157.         @FXML
158.         private void handleRefresh() {
159.             refreshPesananTable();
160.         }
161.
162.         private void refreshPesananTable() {
```

```
            try {
                List<Pesanan> pesananList =
   pesananDAO.getAllPesanan();
                pesananTable.getItems().setAll(pesananList);
            } catch (SQLException e) {
                showAlert("Database Error", "Gagal memuat data
   pesanan: " + e.getMessage());
            }
        }

        private void clearForm() {
            wisataComboBox.getSelectionModel().clearSelection();
            hargaField.clear();
            jumlahField.clear();
            tanggalPicker.setValue(LocalDate.now());
        }

        private void showAlert(String title, String message) {
            Alert alert = new Alert(Alert.AlertType.WARNING);
            alert.setTitle(title);
            alert.setHeaderText(null);
            alert.setContentText(message);
            alert.showAndWait();
        }
    }
```

2. DatabaseConnection.java

```
package com.tubespbo.database;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
        private static final String URL =
   "jdbc:mariadb://localhost:3306/tubes_pbo";
        private static final String USER = "root";
        private static final String PASSWORD = "";

        public static Connection getConnection() throws SQLException
   {
            return DriverManager.getConnection(URL, USER, PASSWORD);
        }
    }
```

3. **Pesanan.java**

```java
package com.tubespbo.model;

import java.time.LocalDate;

public class Pesanan {
    private int id;
        private Wisata wisata;
        private int jumlah;
        private LocalDate tanggalKunjungan;
        private String status;

        public Pesanan(int id, Wisata wisata, int jumlah, LocalDate tanggalKunjungan, String status) {
            this.id = id;
            this.wisata = wisata;
            this.jumlah = jumlah;
            this.tanggalKunjungan = tanggalKunjungan;
            this.status = status;


        }

        public double getTotalHarga() {
            return wisata.getHarga() * jumlah;
        }

        // Getters and Setters
        public int getId() { return id; }
        public Wisata getWisata() { return wisata; }
        public int getJumlah() { return jumlah; }
        public LocalDate getTanggalKunjungan() { return tanggalKunjungan; }
        public String getStatus() { return status; }

        public void setId(int id) { this.id = id; }
        public void setWisata(Wisata wisata) { this.wisata = wisata; }
        public void setJumlah(int jumlah) { this.jumlah = jumlah; }
        public void setTanggalKunjungan(LocalDate tanggalKunjungan) { this.tanggalKunjungan = tanggalKunjungan; }
        public void setStatus(String status) { this.status = status; }
    }
```

4. **PesananDAO.java**

```java
package com.tubespbo.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
      import java.sql.SQLException;
      import java.time.LocalDate;
      import java.util.ArrayList;
      import java.util.List;

      import com.tubespbo.database.DatabaseConnection;

      public class PesananDAO {
          public List<Pesanan> getAllPesanan() throws SQLException {
              List<Pesanan> pesananList = new ArrayList<>();
              String sql = "SELECT p.*, w.nama, w.harga FROM pesanan p JOIN wisata w ON p.wisata_id = w.id WHERE w.nama IS NOT NULL";

              try (Connection conn = DatabaseConnection.getConnection();
                   PreparedStatement stmt = conn.prepareStatement(sql);
                   ResultSet rs = stmt.executeQuery()) {

                  while (rs.next()) {
                      Wisata wisata = new Wisata(
                          rs.getInt("wisata_id"),
                          rs.getString("nama"),
                          rs.getDouble("harga")
                      );

                      Pesanan pesanan = new Pesanan(
                          rs.getInt("id"),
                          wisata,
                          rs.getInt("jumlah"),
                          rs.getDate("tanggal_kunjungan").toLocalDate(),
                          rs.getString("status")
                      );
                      pesananList.add(pesanan);
                  }
              }
              return pesananList;
          }

          public void addPesanan(Pesanan pesanan) throws SQLException {
              String sql = "INSERT INTO pesanan (wisata_id, jumlah, tanggal_kunjungan, status) VALUES (?, ?, ?, ?)";
```

```
48.
49.             try (Connection conn =
    DatabaseConnection.getConnection();
50.                  PreparedStatement stmt = conn.prepareStatement(sql))
    {
51.
52.                 stmt.setInt(1, pesanan.getWisata().getId());
53.                 stmt.setInt(2, pesanan.getJumlah());
54.                 stmt.setDate(3,
    java.sql.Date.valueOf(pesanan.getTanggalKunjungan()));
55.                 stmt.setString(4, pesanan.getStatus());
56.                 stmt.executeUpdate();
57.             }
58.         }
59.
60.         public void updateStatusPesanan(int id, String status) throws
    SQLException {
61.             String sql = "UPDATE pesanan SET status = ? WHERE id =
    ?";
62.
63.             try (Connection conn =
    DatabaseConnection.getConnection();
64.                  PreparedStatement stmt = conn.prepareStatement(sql))
    {
65.
66.                 stmt.setString(1, status);
67.                 stmt.setInt(2, id);
68.                 stmt.executeUpdate();
69.             }
70.         }
71.
72.         public void deletePesanan(int id) throws SQLException {
73.             String sql = "DELETE FROM pesanan WHERE id = ?";
74.
75.             try (Connection conn =
    DatabaseConnection.getConnection();
76.                  PreparedStatement stmt = conn.prepareStatement(sql))
    {
77.
78.                 stmt.setInt(1, id);
79.                 stmt.executeUpdate();
80.             }
81.         }
82.     }
```

5. **Wisata.java**

```java
package com.tubespbo.model;

public class Wisata {
    private int id;
          private String nama;
          private double harga;

          public Wisata(int id, String nama, double harga) {
              this.id = id;
              this.nama = nama;
              this.harga = harga;
          }

          // Getters and Setters
          public int getId() { return id; }
          public String getNama() { return nama; }
          public double getHarga() { return harga; }

          public void setId(int id) { this.id = id; }
          public void setNama(String nama) { this.nama = nama; }
          public void setHarga(double harga) { this.harga = harga; }

          @Override
          public String toString() {
              return nama + " (Rp" + harga + ")";
          }
      }
```

6. **WisataDAO.java**

```
package com.tubespbo.model;

import java.sql.Connection;
      import java.sql.PreparedStatement;
      import java.sql.ResultSet;
      import java.sql.SQLException;
      import java.util.ArrayList;
      import java.util.List;

      import com.tubespbo.database.DatabaseConnection;

      public class WisataDAO {
          public List<Wisata> getAllWisata() throws SQLException {
              List<Wisata> wisataList = new ArrayList<>();
              String sql = "SELECT * FROM wisata";

              try (Connection conn =
   DatabaseConnection.getConnection();
                   PreparedStatement stmt = conn.prepareStatement(sql);
                   ResultSet rs = stmt.executeQuery()) {

                  while (rs.next()) {
                      Wisata wisata = new Wisata(
                          rs.getInt("id"),
                          rs.getString("nama"),
                          rs.getDouble("harga")
                      );
                      wisataList.add(wisata);
                  }
              }
              return wisataList;
          }

          public Wisata getWisataById(int id) throws SQLException {
              String sql = "SELECT * FROM wisata WHERE id = ?";

              try (Connection conn =
   DatabaseConnection.getConnection();
                   PreparedStatement stmt = conn.prepareStatement(sql))
   {

                  stmt.setInt(1, id);
                  try (ResultSet rs = stmt.executeQuery()) {
                      if (rs.next()) {
                          return new Wisata(
                              rs.getInt("id"),
                              rs.getString("nama"),
```

```
                            rs.getDouble("harga")
                        );
                    }
                }
                return null;
            }
        }
```

7. **Main.java**

```
package com.tubespbo;

    import javafx.application.Application;
    import javafx.fxml.FXMLLoader;
    import javafx.scene.Parent;
    import javafx.scene.Scene;
    import javafx.stage.Stage;

    public class Main extends Application {
        @Override
        public void start(Stage primaryStage) {
            try {
                Parent root =
FXMLLoader.load(getClass().getResource("/com/tubespbo/view/MainView.fxm
l"));
                Scene scene = new Scene(root);

                primaryStage.setTitle("Aplikasi Pemesanan Tiket
Wisata");
                primaryStage.setScene(scene);
                primaryStage.setResizable(false);
                primaryStage.show();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

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

8. **MainView.fxml**

```
<?xml version="1.0" encoding="UTF-8"?>

      <?import javafx.scene.control.*?>
      <?import javafx.scene.layout.*?>
      <?import javafx.scene.text.*?>
      <?import javafx.geometry.*?>
      <?import javafx.scene.control.cell.PropertyValueFactory?>

      <VBox xmlns="http://javafx.com/javafx/8.0.171"
   xmlns:fx="http://javafx.com/fxml/1"
   fx:controller="com.tubespbo.controller.MainController">
          <padding>
              <Insets top="10" right="10" bottom="10" left="10"/>
          </padding>
          <spacing>10</spacing>

          <Text text="Aplikasi Pemesanan Tiket Wisata" style="-fx-font-
   size: 20; -fx-font-weight: bold;"/>

          <HBox spacing="10" alignment="CENTER_LEFT">
              <Button text="Beli" onAction="#handleBeli" style="-fx-
   font-weight: bold;"/>
              <Button text="Bayar Tiket" onAction="#handleEdit"/>
              <Button text="Hapus" onAction="#handleHapus"/>
              <Button text="Refresh" onAction="#handleRefresh"/>
          </HBox>

          <Separator/>

          <HBox spacing="10">
              <VBox spacing="5">
                  <Label text="Nama Tempat Wisata:"/>
                  <ComboBox fx:id="wisataComboBox" prefWidth="200"/>
              </VBox>

              <VBox spacing="5">
                  <Label text="Harga Tiket:"/>
                  <TextField fx:id="hargaField" editable="false"
   prefWidth="100"/>
              </VBox>

              <VBox spacing="5">
                  <Label text="Jumlah Tiket:"/>
                  <TextField fx:id="jumlahField" prefWidth="50"/>
              </VBox>

              <VBox spacing="5">
```

```
                <Label text="Tanggal Kunjungan:"/>
                <DatePicker fx:id="tanggalPicker" prefWidth="150"/>
            </VBox>
        </HBox>

        <Separator/>

        <TableView fx:id="pesananTable" prefHeight="200">
            <columns>
                <TableColumn text="Nama Wisata" prefWidth="200">
                </TableColumn>
                <TableColumn text="Harga" prefWidth="100">
                </TableColumn>
                <TableColumn text="Jumlah" prefWidth="50">
                </TableColumn>
                <TableColumn text="Total Harga" prefWidth="120">
                </TableColumn>
                <TableColumn text="Tanggal Kunjungan"
  prefWidth="150">
                </TableColumn>
                <TableColumn text="Status" prefWidth="100">
                </TableColumn>
            </columns>
        </TableView>
    </VBox>
```