

LAPORAN TUGAS
PEMROGRAMAN BERBASIS OBJEK
PERTEMUAN KELIMA 19 SEPTEMBER 2023



UIN SUNAN AMPEL
S U R A B A Y A

DOSEN PEMBIMBING

Bayu Adhi Nugroho, Ph.D.

(197905182014031001)

DISUSUN OLEH

Mochamad Roiyan Rintiarno

(09020622033)

UIN SUNAN AMPEL SURABAYA

TAHUN 2023

1. Tugas

→ Membuat CRUD sederhana menggunakan java dan PostgreSQL (Graphical User Interface)

2. Pembahasan dan Isi

- a. Persiapan dan Langkah-langkah
 - 1. Menyiapkan file Source Code

Terdapat beberapa kelas yang nantinya akan dibuat, yakni kelas Output sebagai file utama dan kelas Book.

a. Kelas Book

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package Tampilan;
6
7  /**
8   *
9   * @author milea
10  */
11  public class Book {
12      private int id;
13      private String name;
14      private String pengarang;
15      private String jenis;
16
17      /**
18       * @return the id
19       */
20      public int getId() {
21          return id;
22      }
23
24      /**
25       * @param id the id to set
26       */
27      public void setId(int id) {
28          this.id = id;
29      }
30
31      /**
32       * @return the name
33       */
34      public String getName() {
35          return name;
36      }
37
38      /**
39       * @param name the name to set
40       */
41      public void setName(String name) {
42          this.name = name;
43      }
44
45      /**
46       * @return the pengarang
47       */
48      public String getPengarang() {
49          return pengarang;
50      }
51
52      /**
53       * @param pengarang the pengarang to set
54       */
55      public void setPengarang(String pengarang) {
56          this.pengarang = pengarang;
57      }
58
59      /**
60       * @return the jenis
61       */
62      public String getJenis() {
63          return jenis;
64      }
65
66      /**
67       * @param jenis the jenis to set
68       */
69      public void setJenis(String jenis) {
70          this.jenis = jenis;
71      }
72  }
73
```


b. Kelas Output

```
1 package Tampilan;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Statement;
9 import java.util.ArrayList;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12 import javax.swing.JOptionPane;
13 import javax.swing.table.DefaultTableModel;
14
15 /**
16  *
17  * @author milea
18  */
19 public class Output extends javax.swing.JFrame {
20
21     private ArrayList<Book> dataBuku;
22
23     /**
24      * Creates new form Output
25      */
26     public Output() {
27         this.refresh();
28     }
29
30     private void refresh(){
31         try {
32             dataBuku = new ArrayList<>();
33             initComponents();
34             Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost:5432/perpustakaan", "postgres", "postgres");
35             tampil(conn);
36         } catch (SQLException ex) {
37             Logger.getLogger(Output.class.getName()).log(Level.SEVERE, null, ex);
38         }
39     }
40
41     public void peringatan(String pesan) {
42         JOptionPane.showMessageDialog(rootPane, pesan);
43     }
44
45     private int masukkanData(Connection conn, int id, String name, String pengarang, String jenis) throws SQLException {
46         PreparedStatement pst = conn.prepareStatement("INSERT INTO books (id, name, pengarang, jenis) VALUES(?, ?, ?, ?)");
47         pst.setInt(1, id);
48         pst.setString(2, name);
49         pst.setString(3, pengarang);
50         pst.setString(4, jenis);
51         return pst.executeUpdate();
52     }
53
54     private int updateData(Connection conn, int id, String name, String pengarang, String jenis) throws SQLException {
55         PreparedStatement pst = conn.prepareStatement("UPDATE books SET name=?, pengarang=?, jenis=? WHERE id=?");
56         pst.setInt(1, id);
57         pst.setString(2, name);
58         pst.setString(3, pengarang);
59         pst.setString(4, jenis);
60         return pst.executeUpdate();
61     }
62
63     private int deleteData(Connection conn, int id) throws SQLException {
64         PreparedStatement pst = conn.prepareStatement("DELETE FROM books WHERE id=?");
65         pst.setInt(1, id);
66         return pst.executeUpdate();
67     }
68
69     private void resetField(){
70         inputID.setText("");
71         inputJudul.setText("");
72         inputJenisBuku.setText("");
73         inputPengarang.setText("");
74     }
75 }
```

```
1 private void tampil(Connection conn) {
2     dataBuku.clear();
3     try {
4         String sql = "select * from books ORDER BY id";
5         Statement st = conn.createStatement();
6         ResultSet rs = st.executeQuery(sql);
7         while (rs.next()) {
8             Book data = new Book();
9             data.setId(rs.getInt(1));
10            data.setName(String.valueOf(rs.getObject(2)));
11            data.setPengarang(String.valueOf(rs.getObject(3)));
12            data.setJenis(String.valueOf(rs.getObject(4)));
13            dataBuku.add(data);
14        }
15        DefaultTableModel model = (DefaultTableModel) tabelBuku.getModel();
16        model.setRowCount(0);
17        for (Book data : dataBuku) {
18            Object[] baris = new Object[4];
19            baris[0] = data.getId();
20            baris[1] = data.getName();
21            baris[2] = data.getPengarang();
22            baris[3] = data.getJenis();
23            model.addRow(baris);
24        }
25    } catch (SQLException ex) {
26        Logger.getLogger(Output.class.getName()).log(Level.SEVERE, null, ex);
27    }
28 }
```

[illegible]

```

1 private void inputIDActionPerformed(java.awt.event.ActionEvent evt) {
2     // TODO add your handling code here:
3 }
4
5 private void inputPengarangActionPerformed(java.awt.event.ActionEvent evt) {
6     // TODO add your handling code here:
7 }
8
9 private void inputJudulActionPerformed(java.awt.event.ActionEvent evt) {
10    // TODO add your handling code here:
11 }
12
13 private void inputJenisBukuActionPerformed(java.awt.event.ActionEvent evt) {
14    // TODO add your handling code here:
15 }
16
17 private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
18    // TODO add your handling code here:
19    int id = Integer.parseInt(inputID.getText().trim());
20    String name = inputJudul.getText();
21    String pengarang = inputPengarang.getText();
22    String jenis = inputJenisBuku.getText();
23    if (!name.isEmpty() && !pengarang.isEmpty() && !jenis.isEmpty() && id > 0) {
24        try {
25            // TODO add your handling code here:
26            Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost:5432/perpustakaan", "postgres", "postgres");
27            int k = masukkanData(conn, id, name, pengarang, jenis);
28            if (k > 0) {
29                tampil(conn);
30                this.peringatan("Simpan Berhasil");
31            } else {
32                this.peringatan("Simpan Gagal");
33            }
34        } catch (SQLException ex) {
35            Logger.getLogger(Output.class.getName()).log(Level.SEVERE, null, ex);
36        }
37    } else {
38        this.peringatan("Input Data Buku Gagal");
39    }
40    this.resetField();
41 }

```

```

1 private void btnRefreshActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_btnRefreshActionPerformed
2     // TODO add your handling code here:
3     this.resetField();
4     this.refresh();
5 } //GEN-LAST:event_btnRefreshActionPerformed
6
7 private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_btnUpdateActionPerformed
8     // TODO add your handling code here:
9     int id = Integer.parseInt(inputID.getText().trim());
10    String name = inputJudul.getText();
11    String pengarang = inputPengarang.getText();
12    String jenis = inputJenisBuku.getText();
13    if (!name.isEmpty() && !pengarang.isEmpty() && !jenis.isEmpty() && id > 0) {
14        try {
15            // TODO add your handling code here:
16            Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost:5432/perpustakaan", "postgres", "postgres");
17            int k = updateData(conn, id, name, pengarang, jenis);
18            if (k > 0) {
19                tampil(conn);
20                this.peringatan("Update Berhasil");
21            } else {
22                this.peringatan("Update Gagal");
23            }
24        } catch (SQLException ex) {
25            Logger.getLogger(Output.class.getName()).log(Level.SEVERE, null, ex);
26        }
27    } else {
28        this.peringatan("Update Data Mahasiswa Gagal");
29    }
30    this.resetField();
31 }

```



```

1 private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_btnDeleteActionPerformed
2     // TODO add your handling code here:
3     int id = Integer.parseInt(inputID.getText().trim());
4     if (id > 0) {
5         try {
6             // TODO add your handling code here:
7             Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost:5432/perpustakaan", "postgres", "postgres");
8             int k = deleteData(conn, id);
9             if (k > 0) {
10                 tampil(conn);
11                 this.peringatan("Hapus Berhasil");
12             } else {
13                 this.peringatan("Hapus Gagal");
14             }
15         } catch (SQLException ex) {
16             Logger.getLogger(Output.class.getName()).log(Level.SEVERE, null, ex);
17         }
18     } else {
19         this.peringatan("Delete Data Buku Gagal");
20     }
21     this.resetField();
22 }

```

```

1 public static void main(String args[]) {
2     /* Set the Nimbus look and feel */
3     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
4     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
5      * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
6      */
7     try {
8         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
9             if ("Nimbus".equals(info.getName())) {
10                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
11                 break;
12             }
13         }
14     } catch (ClassNotFoundException ex) {
15         java.util.logging.Logger.getLogger(Output.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
16     } catch (InstantiationException ex) {
17         java.util.logging.Logger.getLogger(Output.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
18     } catch (IllegalAccessException ex) {
19         java.util.logging.Logger.getLogger(Output.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
20     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
21         java.util.logging.Logger.getLogger(Output.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
22     }
23     //</editor-fold>
24
25     /* Create and display the form */
26     java.awt.EventQueue.invokeLater(new Runnable() {
27         public void run() {
28             new Output().setVisible(true);
29         }
30     });
31 }
32
33 // Variables declaration - do not modify//GEN-BEGIN:variables
34 private javax.swing.JButton btnDelete;
35 private javax.swing.JButton btnInsert;
36 private javax.swing.JButton btnRefresh;
37 private javax.swing.JButton btnUpdate;
38 private javax.swing.JTextField inputID;
39 private javax.swing.JTextField inputJenisBuku;
40 private javax.swing.JTextField inputJudul;
41 private javax.swing.JTextField inputPengarang;
42 private javax.swing.JScrollPane jScrollPane1;
43 private javax.swing.JLabel label1;
44 private javax.swing.JLabel labelID;
45 private javax.swing.JLabel labelJenisBuku;
46 private javax.swing.JLabel labelJudul;
47 private javax.swing.JLabel labelPengarang;
48 private javax.swing.JTable tabelBuku;
49 // End of variables declaration//GEN-END:variables
50 }

```

2. Output

a. Tampilan Awal

Daftar Buku

ID Buku

Pengarang

Judul

Jenis Buku

ID	Judul	Pengarang	Jenis
2	bandung lautan air	mpu gandrang	fiksi
4	Mereka tak pernah me...	Tipe X	nostalgia
5	Bila kau tak disampingku	Sheila on 7	Lirik lagu
10	Dan	Sheila	Lagu
11	Nemen	GuyonWaton	Lagu Jawa

b. Ketika menambahkan data

Daftar Buku

ID Buku

Pengarang

Judul

Jenis Buku

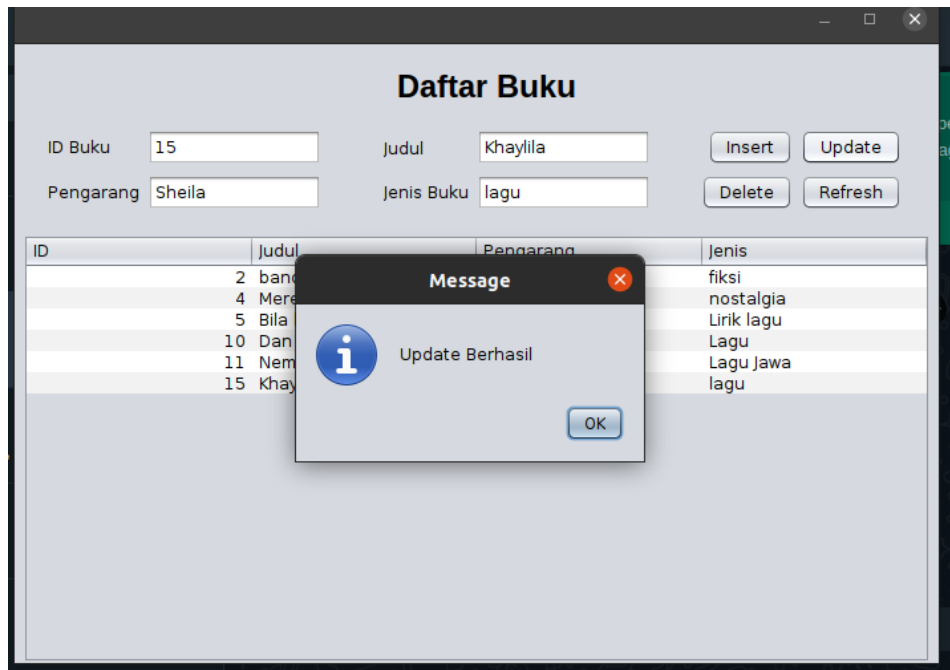
ID	Judul	Pengarang	Jenis
2	band		fiksi
4	Mere		nostalgia
5	Bila		Lirik lagu
10	Dan		Lagu
11	Nem		Lagu Jawa
15	Tand		apaya

Message

i

Simpan Berhasil

c. Ketika mengubah data



d. Ketika menghapus data

