LAPORAN PRAKTIKUM

BAHASA PEMROGRAMAN 2

MODUL 7

Dosen pengampu : Yulyanto, S.Kom., M.TI.

Disusun oleh :

Nama : Muhammad Rizal Nurfirdaus

NIM : 20230810088

Jadwal : Rabu, 14:40 – 16:15

Kelas : TINFC-2023-04

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS KUNINGAN

2025

DAFTAR ISI

BAB I

PRETEST

1. Jelaskan tentang :

   Statement

   ResultSet

Kerjakan pda selembar kertas dengan waktu 5 menit dan dikumpulkan melalui Asisten lab.

a. **Statement**
   Statement adalah antarmuka di dalam Java (package java.sql) yang digunakan untuk mengeksekusi pernyataan SQL statis terhadap database. Statement dibuat dari objek Connection dan digunakan untuk menjalankan query seperti SELECT, INSERT, UPDATE, atau DELETE.

   Contoh:

   ```
   Statement stmt = conn.createStatement();

   ResultSet rs = stmt.executeQuery("SELECT * FROM users");
   ```

b. **ResultSet**
   ResultSet adalah antarmuka yang digunakan untuk menyimpan hasil dari query SELECT. Objek ini digunakan untuk menelusuri data baris demi baris menggunakan method seperti next(), dan mengambil nilai kolom dengan getString(), getInt(), dll.
   Contoh:

   ```
   while (rs.next()) {
   System.out.println(rs.getString("username"));
   }
   ```

BAB II

PRAKTIKUM

1. Event Load Data ke Form
2. EDIT DATA
3. DELETE DATA

```
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.PreparedStatement;
import javax.swing.ImageIcon;

/**
 *
 * @author Muhammad Rizal Nur F (dimodifikasi berdasarkan permintaan)
 */
public class FUser extends javax.swing.JFrame {

    Statement st;
    ResultSet rs;
    Koneksi koneksi;

    public FUser() {
```

```
        koneksi = new Koneksi();
        initComponents();
        load_data();
        IDOtomatis();
        level();
        bersih(); // Panggil bersih di akhir untuk set state tombol awal
        IIdUser.setEditable(false);
    }

    private void load_data() {
        Object header[] = {"ID USER", "LEVEL", "NAMA USER", "JENIS
KELAMIN",
            "NO HANDPHONE", "USERNAME", "PASSWORD"};
        DefaultTableModel data = new DefaultTableModel(null, header);
        TUser.setModel(data);

        String sql = "SELECT tbl_user.id_user, tbl_level.level, tbl_user.nama_user, "
            + "tbl_user.jk, tbl_user.NOPE, tbl_user.username, tbl_user.password "
            + "FROM tbl_user INNER JOIN tbl_level ON tbl_user.id_level =
tbl_level.id_level ORDER BY tbl_user.id_user ASC;";

        try {
            if (koneksi.con == null) {
                JOptionPane.showMessageDialog(null, "Koneksi ke database gagal!");
                return;
            }
            st = koneksi.con.createStatement();
            rs = st.executeQuery(sql);
            while (rs.next()) {
                String k1 = rs.getString(1);
```

```
            String k2 = rs.getString(2);
            String k3 = rs.getString(3);
            String k4 = rs.getString(4);
            String k5 = rs.getString(5);
            String k6 = rs.getString(6);
            String k7 = rs.getString(7); // Password
            String k[] = {k1, k2, k3, k4, k5, k6, k7};
            data.addRow(k);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error saat load data: " +
e.getMessage());
        e.printStackTrace();
    }
  }

  private void IDOtomatis() {
    try {
        if (koneksi.con == null) {
            // JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");
// Sudah ada di load_data
            return;
        }
        st = koneksi.con.createStatement();
        String sql = "SELECT * FROM tbl_user order by id_user desc";
        rs = st.executeQuery(sql);
        if (rs.next()) {
            String idUser = rs.getString("id_user").substring(4);
            String AN = "" + (Integer.parseInt(idUser) + 1);
            String Nol = "";
```

```
            if (AN.length() == 1) {
                Nol = "00";
            } else if (AN.length() == 2) {
                Nol = "0";
            } else if (AN.length() == 3) {
                Nol = "";
            }
            IIdUser.setText("USER" + Nol + AN);
        } else {
            IIdUser.setText("USER001");
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error ID Otomatis: " +
e.getMessage());
        e.printStackTrace();
    }
}

private void level() {
    if (CLevel.getItemCount() > 0) {
        CLevel.setSelectedIndex(0);
    }
}

private void bersih() {
    IDOtomatis(); // Generate ID baru dulu
    if (CLevel.getItemCount() > 0) {
        CLevel.setSelectedIndex(0);
    }
```

```
        INamaUser.setText("");
        Gjk.clearSelection();
        INope.setText("");
        IUser.setText("");
        IPassword.setText("");
        INamaUser.requestFocus();

        // Reset button states
        BSimpan.setEnabled(true);
        BEdit.setEnabled(false);
        BHapus.setEnabled(false);
        IIdUser.setEditable(false); // ID User tidak boleh diedit, tapi ID Otomatis
bisa dijalankan
    }

    private int getIdLevelByName(String levelName) {
        // Fungsi ini sudah baik menggunakan PreparedStatement
        try {
            if (koneksi.con == null) return -1;
            String sql = "SELECT id_level FROM tbl_level WHERE level = ?";
            PreparedStatement ps = koneksi.con.prepareStatement(sql);
            ps.setString(1, levelName);
            ResultSet rsLevel = ps.executeQuery();
            if (rsLevel.next()) {
                return rsLevel.getInt("id_level");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error mendapatkan ID Level: " +
e.getMessage());
            e.printStackTrace();
```

```
        }
        // Fallback jika tidak ketemu (sebaiknya tidak terjadi jika data konsisten)
        // Atau bisa juga handle jika query tidak return apa2, misalnya return 0 atau -
1
        // dan berikan pesan error yang lebih spesifik.
        // Untuk sekarang, jika query gagal, kita biarkan pesan error dari catch
muncul.
        // Untuk contoh fallback, jika nama level di DB berbeda:
        // if (levelName.equalsIgnoreCase("ADMINISTRATOR")) return 1;
        return 0; // Indikasi level tidak ditemukan atau error
    }

    // --- Method untuk Edit Data (sesuai gambar) ---
    private void Edit() {
        try {
            if (koneksi.con == null) {
                JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");
                return;
            }

            // Mengambil Data Jenis Kelamin
            String jk = "";
            if (JkL.isSelected()) {
                jk = "L";
            } else if (JkP.isSelected()) {
                jk = "P";
            }

            // Mengambil ID_LEVEL dari ComboBox
```

```
        // Cara sesuai gambar (direct query, rentan SQL Injection jika nama level
aneh)
        // Cara yang lebih baik adalah menggunakan getIdLevelByName() yang
sudah ada
        // Namun, demi mengikuti gambar, kita gunakan query langsung:
        String id_level_str = "";
        st = koneksi.con.createStatement(); // Pastikan st diinisialisasi
        String query_level = "SELECT ID_LEVEL FROM tbl_level WHERE
LEVEL='" + CLevel.getSelectedItem().toString() + "'";
        ResultSet rs_level = st.executeQuery(query_level);
        if (rs_level.next()) {
            id_level_str = rs_level.getString("ID_LEVEL");
        } else {
            JOptionPane.showMessageDialog(this, "Level tidak ditemukan di
database!", "Error", JOptionPane.ERROR_MESSAGE);
            return; // Hentikan jika level tidak valid
        }

        // Validasi dasar (seperti di BSimpan)
        if (INamaUser.getText().isEmpty() || jk.isEmpty() ||
INope.getText().isEmpty() || IUser.getText().isEmpty() || new
String(IPassword.getPassword()).isEmpty()) {
            JOptionPane.showMessageDialog(this, "Data Harus Diisi !!",
"Peringatan", JOptionPane.WARNING_MESSAGE);
            return;
        }

        // Query Input ke Tabel USer
        // PERHATIAN: Penggunaan Statement dengan konkatenasi string sangat
rentan SQL Injection!
        String sql_edit = "UPDATE tbl_user SET id_level='" + id_level_str + "'
```

```
                + ", nama_user='" + INamaUser.getText() + "'"
                + ", jk='" + jk + "'"
                + ", nope='" + INope.getText() + "'"
                + ", username='" + IUser.getText() + "'"
                + ", password='" + new String(IPassword.getPassword()) + "'"
                + " WHERE id_user='" + IIdUser.getText() + "'";

            st.executeUpdate(sql_edit);
            JOptionPane.showMessageDialog(null, "Data berhasil Di Update");

            load_data();
            bersih(); // Ini akan mengaktifkan BSimpan dan menonaktifkan BEdit, BHapus
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error saat update data: " + e.getMessage());
            e.printStackTrace();
        }
    }

    // --- Method untuk Hapus Data (sesuai gambar) ---
    private void Hapus() {
        try {
            if (koneksi.con == null) {
                JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");
                return;
            }
            // Konfirmasi sebelum hapus
            int konfirmasiHapus = JOptionPane.showConfirmDialog(this,
                    "Apakah Anda yakin ingin menghapus data ini?",
```

```
                "Konfirmasi Hapus",
                JOptionPane.YES_NO_OPTION,
                JOptionPane.WARNING_MESSAGE);

        if (konfirmasiHapus == JOptionPane.YES_OPTION) {
            st = koneksi.con.createStatement(); // Pastikan st diinisialisasi
            // PERHATIAN: Penggunaan Statement dengan konkatenasi string sangat rentan SQL Injection!
            String sql_delete = "DELETE FROM tbl_user WHERE id_user='" + IIdUser.getText() + "'";
            st.executeUpdate(sql_delete);
            JOptionPane.showMessageDialog(null, "Data berhasil Di Hapus");

            load_data();
            bersih(); // Ini akan mengaktifkan BSimpan dan menonaktifkan BEdit, BHapus
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error saat hapus data: " + e.getMessage());
        e.printStackTrace();
    }
}


/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
```

```
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        Gjk = new javax.swing.ButtonGroup();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        IIdUser = new javax.swing.JTextField();
        INamaUser = new javax.swing.JTextField();
        INope = new javax.swing.JTextField();
        IUser = new javax.swing.JTextField();
        CLevel = new javax.swing.JComboBox<>();
        JkL = new javax.swing.JRadioButton();
        JkP = new javax.swing.JRadioButton();
        IPassword = new javax.swing.JPasswordField();
        jLabel9 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        TUser = new javax.swing.JTable();
        jLabel10 = new javax.swing.JLabel();
        BSimpan = new javax.swing.JButton();
        BEdit = new javax.swing.JButton();
        BHapus = new javax.swing.JButton();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Form User");

        jLabel1.setFont(new java.awt.Font("Segoe UI", 3, 12)); // NOI18N
        jLabel1.setText("FORM INPUT DATA");

        jLabel2.setText("ID USER");

        jLabel3.setText("LEVEL");

        jLabel4.setText("NAMA USER");

        jLabel5.setText("JENIS KELAMIN");

        jLabel6.setText("NO HP");

        jLabel7.setText("USERNAME");

        jLabel8.setText("PASSWORD");

        CLevel.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"ADMIN", "GUDANG", "KASIR" }));

        Gjk.add(JkL);
        JkL.setText("L");
        JkL.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                JkLActionPerformed(evt);
            }
```

```
});

Gjk.add(JkP);
JkP.setText("P");

jLabel9.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
jLabel9.setText("DATA USER APLIKASI KASIR");

TUser.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
TUser.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        TUserMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(TUser);

jLabel10.setFont(new java.awt.Font("Segoe UI", 3, 12)); // NOI18N
jLabel10.setText("DATA USER");
```

```
        BSimpan.setText("SIMPAN");
        BSimpan.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BSimpanActionPerformed(evt);
            }
        });

        BEdit.setText("EDIT");
        BEdit.setEnabled(false);
        BEdit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BEditActionPerformed(evt);
            }
        });

        BHapus.setText("HAPUS");
        BHapus.setEnabled(false);
        BHapus.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BHapusActionPerformed(evt);
            }
        });

        try {
            BSimpan.setIcon(new
ImageIcon(getClass().getResource("/icons/save_icon.png")));
            BEdit.setIcon(new
ImageIcon(getClass().getResource("/icons/edit_icon.png")));
            BHapus.setIcon(new
ImageIcon(getClass().getResource("/icons/delete_icon.png")));
```

```
        } catch (NullPointerException e) {
            System.err.println("Warning: Icon file not found. Check paths in
initComponents().");
        }

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(42, 42, 42)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(135, 135, 135)
                        .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jLabel10)
                        .addGap(236, 236, 236))
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                            .addComponent(jLabel3)
                            .addComponent(jLabel4)
                            .addComponent(jLabel5)
                            .addComponent(jLabel6)
```

```
                .addComponent(jLabel7)
                .addComponent(jLabel8)
                .addComponent(jLabel2))
            .addGap(25, 25, 25)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                .addGroup(layout.createSequentialGroup()
                  .addComponent(BSimpan)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                  .addComponent(BEdit)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                  .addComponent(BHapus)
                  .addGap(0, 0, Short.MAX_VALUE))
                .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
                    .addGroup(layout.createSequentialGroup()
                      .addComponent(JkL)
                      .addGap(18, 18, 18)
                      .addComponent(JkP))
                    .addComponent(CLevel,
javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(IIdUser,
javax.swing.GroupLayout.DEFAULT_SIZE, 206, Short.MAX_VALUE)
                    .addComponent(INope)
                    .addComponent(IUser)
                    .addComponent(IPassword)
```

```
                .addComponent(INamaUser))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
127, Short.MAX_VALUE)

                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 630,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(0, 34, Short.MAX_VALUE))))))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

            .addComponent(jLabel9)

            .addGap(350, 350, 350))

    );

    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGap(20, 20, 20)

            .addComponent(jLabel9)

            .addGap(32, 32, 32)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)

                .addComponent(jLabel1)

                .addComponent(jLabel10))

            .addGap(9, 9, 9)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

                .addGroup(layout.createSequentialGroup()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRA
ILING)
                .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(jLabel2)
                    .addComponent(IIdUser,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(jLabel3)
                    .addComponent(CLevel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(jLabel4)
                    .addComponent(INamaUser,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(jLabel5)
                    .addComponent(JkL)
```

```
                .addComponent(JkP))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel6)
                .addGap(6, 6, 6))
            .addComponent(INope,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
            .addComponent(jLabel7)
            .addComponent(IUser,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 141,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
        .addComponent(jLabel8)
        .addComponent(IPassword,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(32, 32, 32)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addComponent(BSimpan)
        .addComponent(BEdit)
```

```
                .addComponent(BHapus))
            .addContainerGap(117, Short.MAX_VALUE))
        );

        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>//GEN-END:initComponents

    private void JkLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_JkLActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_JkLActionPerformed

    private void BSimpanActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BSimpanActionPerformed
        if (koneksi.con == null) {
            JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");
            return;
        }

        String idUser = IIdUser.getText();
        String levelName = CLevel.getSelectedItem().toString();
        String namaUser = INamaUser.getText();
        String jk = "";
        if (JkL.isSelected()) {
            jk = "L";
        } else if (JkP.isSelected()) {
            jk = "P";
        }
        String nope = INope.getText();
```

```
        String username = IUser.getText();
        String password = new String(IPassword.getPassword());

        if (namaUser.isEmpty() || jk.isEmpty() || nope.isEmpty() ||
username.isEmpty() || password.isEmpty() || CLevel.getSelectedIndex() == -1 ||
(CLevel.getSelectedItem() != null &&
CLevel.getSelectedItem().toString().equals("- Pilih Level -")) ) {
            JOptionPane.showMessageDialog(this, "Data Harus Diisi !!",
"Peringatan", JOptionPane.WARNING_MESSAGE);
            return;
        }

        int konfirmasi = JOptionPane.showConfirmDialog(this,
                "Apakah Data Sudah Benar? SIMPAN?",
                "Simpan Data",
                JOptionPane.YES_NO_OPTION,
                JOptionPane.QUESTION_MESSAGE);

        if (konfirmasi == JOptionPane.YES_OPTION) {
            int idLevel = getIdLevelByName(levelName); // Gunakan fungsi yang
sudah ada
            if (idLevel <= 0) { // getIdLevelByName akan return 0 atau -1 jika
error/tidak ketemu
                JOptionPane.showMessageDialog(this, "Level tidak valid atau tidak
ditemukan di database.", "Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            try {
                String checkSql = "SELECT COUNT(*) FROM tbl_user WHERE
username = ? AND id_user != ?";
                PreparedStatement checkPs = koneksi.con.prepareStatement(checkSql);
```

```
            checkPs.setString(1, username);
            checkPs.setString(2, idUser);
            ResultSet checkRs = checkPs.executeQuery();
            if (checkRs.next() && checkRs.getInt(1) > 0) {
                JOptionPane.showMessageDialog(this, "Username '" + username + "' sudah digunakan!", "Error", JOptionPane.ERROR_MESSAGE);
                IUser.requestFocus();
                return;
            }

            // Gunakan PreparedStatement untuk keamanan
            String sql = "INSERT INTO tbl_user (id_user, id_level, nama_user, jk, NOPE, username, password) VALUES (?, ?, ?, ?, ?, ?, ?)";
            PreparedStatement ps = koneksi.con.prepareStatement(sql);
            ps.setString(1, idUser);
            ps.setInt(2, idLevel);
            ps.setString(3, namaUser);
            ps.setString(4, jk);
            ps.setString(5, nope);
            ps.setString(6, username);
            ps.setString(7, password);

            int rowsInserted = ps.executeUpdate();
            if (rowsInserted > 0) {
                JOptionPane.showMessageDialog(this, "Data Barang Berhasil Dimasukan");
                load_data();
                bersih();
            }
        } catch (SQLException e) {
```

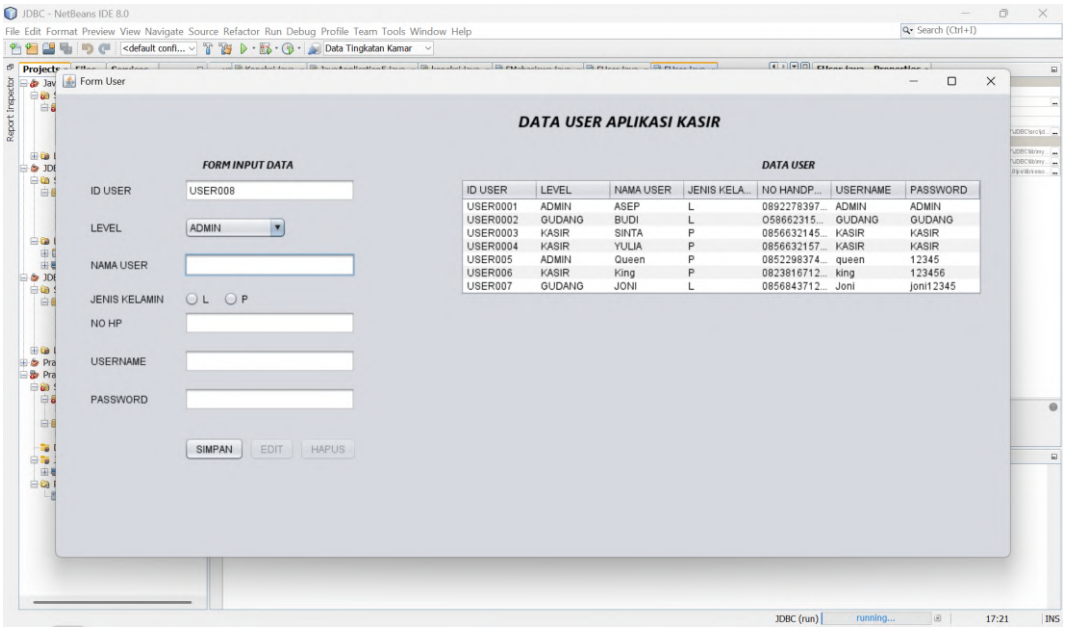
```
            if (e.getSQLState() != null && e.getSQLState().startsWith("23")) { // Cek null untuk SQLState
                JOptionPane.showMessageDialog(this, "ID User '" + idUser + "' sudah ada atau terjadi kesalahan integritas data.\nDetail: " + e.getMessage(), "Error Simpan", JOptionPane.ERROR_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(this, "Error saat menyimpan data: " + e.getMessage(), "Error Simpan", JOptionPane.ERROR_MESSAGE);
            }
            e.printStackTrace();
        }
    }
    }//GEN-LAST:event_BSimpanActionPerformed

    private void BEditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BEditActionPerformed
        Edit(); // Panggil method Edit
    }//GEN-LAST:event_BEditActionPerformed

    private void BHapusActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BHapusActionPerformed
        Hapus(); // Panggil method Hapus
    }//GEN-LAST:event_BHapusActionPerformed

    private void TUserMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_TUserMouseClicked
        int bar = TUser.getSelectedRow();
        if (bar != -1) { // Pastikan ada baris yang terpilih
            String a = TUser.getValueAt(bar, 0).toString(); // ID USER
            String b = TUser.getValueAt(bar, 1).toString(); // LEVEL
            String c = TUser.getValueAt(bar, 2).toString(); // NAMA USER
            String d = TUser.getValueAt(bar, 3).toString(); // JENIS KELAMIN
```

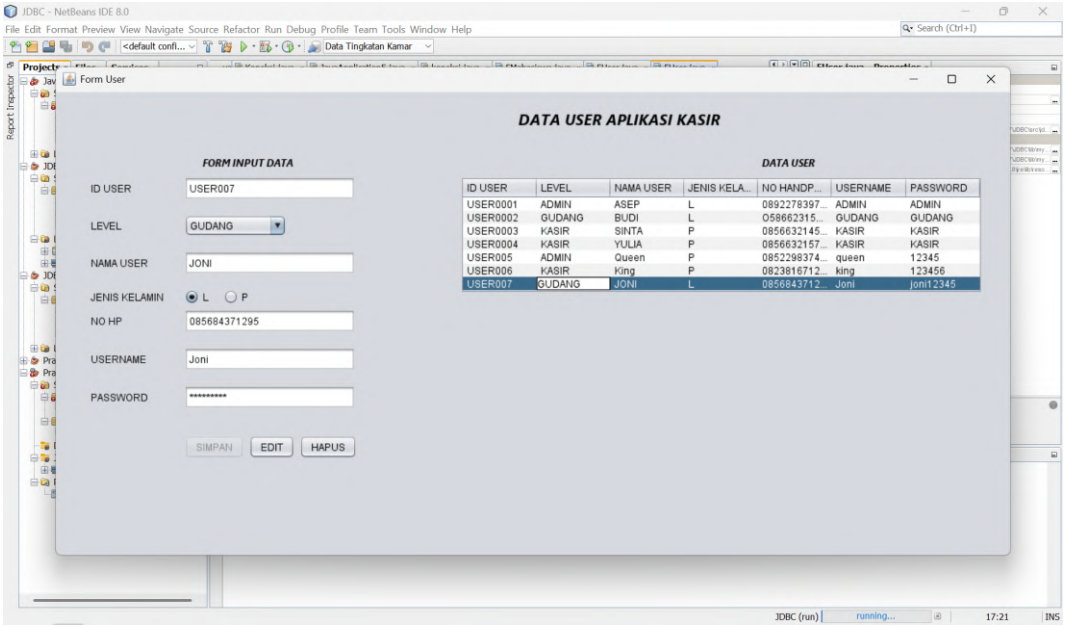
```
String e = TUser.getValueAt(bar, 4).toString(); // NO HANDPHONE
String f = TUser.getValueAt(bar, 5).toString(); // USERNAME
String g = TUser.getValueAt(bar, 6).toString(); // PASSWORD

IIdUser.setText(a);
// Cari item di CLevel yang cocok dengan string b dan set sebagai selected
for (int i = 0; i < CLevel.getItemCount(); i++) {
   if (CLevel.getItemAt(i).toString().equalsIgnoreCase(b)) {
      CLevel.setSelectedIndex(i);
      break;
   }
}
INamaUser.setText(c);
INope.setText(e);
IUser.setText(f);
IPassword.setText(g);

if ("L".equalsIgnoreCase(d)) {
   JkL.setSelected(true);
} else if ("P".equalsIgnoreCase(d)) {
   JkP.setSelected(true);
} else {
   Gjk.clearSelection(); // Jika tidak L atau P, bersihkan pilihan
}

// Atur state tombol
BSimpan.setEnabled(false);
BEdit.setEnabled(true);
BHapus.setEnabled(true);
```

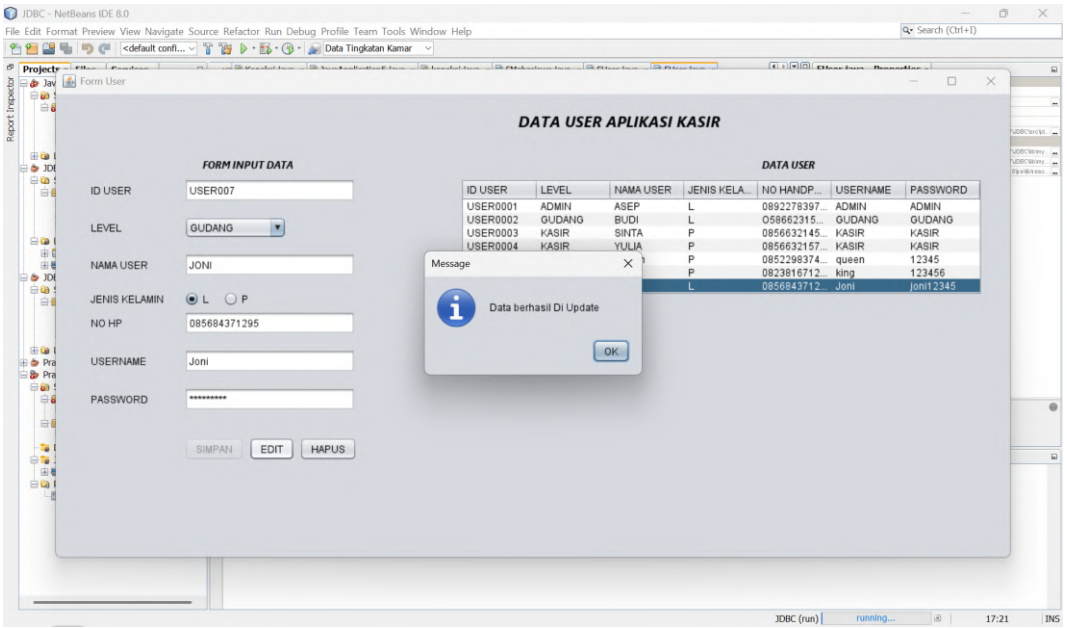
```
        IIdUser.setEditable(false); // ID User tidak boleh diedit
    }
}//GEN-LAST:event_TUserMouseClicked


/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

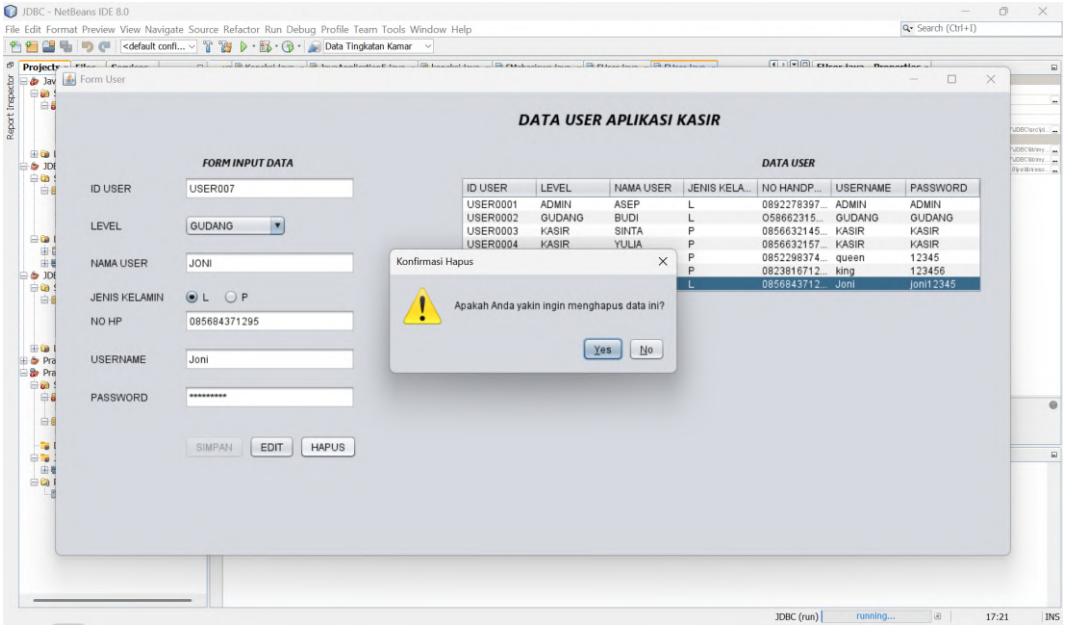
```
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new FUser().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton BEdit;
    private javax.swing.JButton BHapus;
    private javax.swing.JButton BSimpan;
    private javax.swing.JComboBox<String> CLevel;
    private javax.swing.ButtonGroup Gjk;
    private javax.swing.JTextField IIdUser;
    private javax.swing.JTextField INamaUser;
    private javax.swing.JTextField INope;
    private javax.swing.JPasswordField IPassword;
    private javax.swing.JTextField IUser;
    private javax.swing.JRadioButton JkL;
```

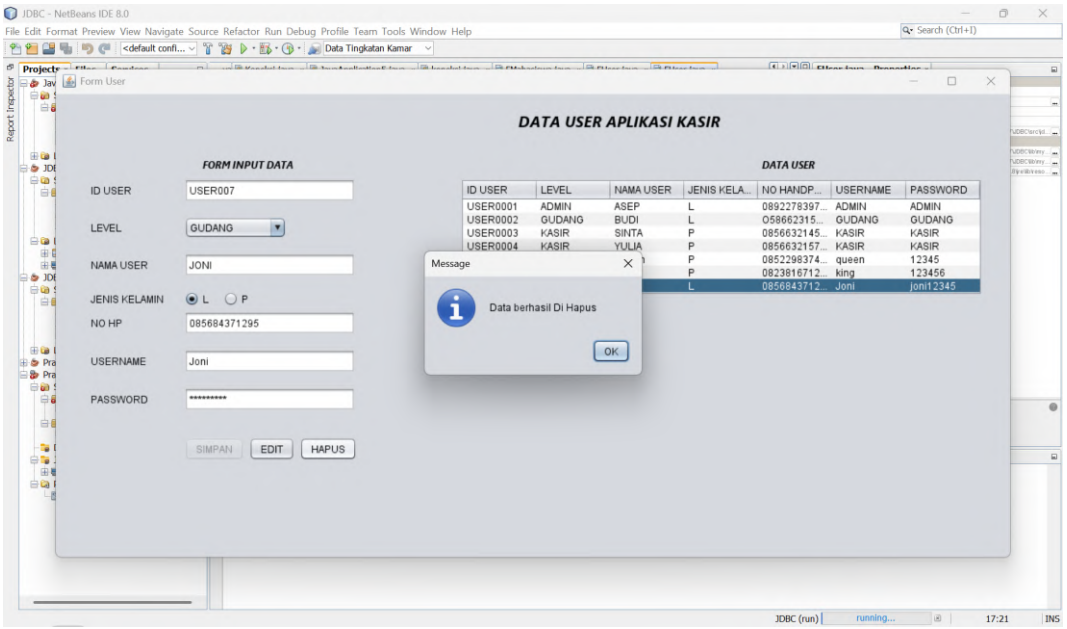
```
    private javax.swing.JRadioButton JkP;
    private javax.swing.JTable TUser;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration//GEN-END:variables
}
```




JDBC - NetBeans IDE 8.0
Form User
DATA USER APLIKASI KASIR
FORM INPUT DATA
ID USER
USER007
LEVEL
GUDANG
NAMA USER
JONI
JENIS KELAMIN
L
P
NO HP
085684371295
USERNAME
Joni
PASSWORD
SIMPAN
EDIT
HAPUS
DATA USER
ID USER
LEVEL
NAMA USER
JENIS KELA...
NO HANDP...
USERNAME
PASSWORD
USER0001 ADMIN ASEP L 0892278397... ADMIN ADMIN
USER0002 GUDANG BUDI L 058662315... GUDANG GUDANG
USER0003 KASIR SINTA P 0856632145... KASIR KASIR
USER0004 KASIR YULIA P 0856632157... KASIR KASIR
USER005 ADMIN Queen P 0852298374... queen 12345
USER006 KASIR King P 0823816712... king 123456
USER007 GUDANG JONI L 0856843712... Joni joni12345


JDBC - NetBeans IDE 8.0
Form User
DATA USER APLIKASI KASIR
FORM INPUT DATA
ID USER
USER007
LEVEL
GUDANG
NAMA USER
JONI
JENIS KELAMIN
L
P
NO HP
085684371295
USERNAME
Joni
PASSWORD
SIMPAN
EDIT
HAPUS
DATA USER
Message
Data berhasil Di Update
OK





Analisis : Implementasi terbaru pada aplikasi manajemen data pengguna, yang berfokus pada penambahan interaktivitas tabel dan fungsionalitas manipulasi data lanjutan (Edit dan Hapus), secara signifikan meningkatkan kedalaman dan kematangan aplikasi. Penambahan *event listener* mouseClicked pada komponen JTable (TUser) merupakan langkah krusial yang mentransformasi tabel dari sekadar penampil data statis menjadi gerbang interaktif untuk operasi lebih lanjut. Kemampuan untuk secara otomatis mengisi *form input* dengan data dari baris tabel yang dipilih pengguna tidak hanya menghemat waktu tetapi juga mengurangi potensi kesalahan input manual saat pengguna ingin memodifikasi atau menghapus data spesifik. Logika pengaturan status tombol (menonaktifkan "Simpan" dan mengaktifkan "Edit" serta "Hapus" saat data dipilih) juga

mencerminkan perancangan alur kerja pengguna yang intuitif dan terarah, membimbing pengguna menuju tindakan yang relevan.

Pengembangan fungsi Edit() dan Hapus() yang diintegrasikan dengan tombol masing-masing, meskipun pada contoh awal menggunakan metode konstruksi *query* SQL yang rentan (namun telah direkomendasikan perbaikannya menggunakan PreparedStatement), menandai pencapaian penting dalam melengkapi siklus operasi CRUD (*Create, Read, Update, Delete*) pada aplikasi. Fungsi Edit memungkinkan pengguna untuk melakukan koreksi atau pembaruan pada data yang sudah ada dengan mudah setelah data tersebut ditampilkan di *form*, sementara fungsi Hapus, yang dilengkapi dengan dialog konfirmasi, memberikan mekanisme yang aman untuk menghilangkan data yang tidak lagi diperlukan dari sistem. Keberhasilan integrasi kedua fungsi ini, yang dilanjutkan dengan pembaruan otomatis pada JTable dan pengosongan *form* melalui pemanggilan load_data() dan bersih(), memastikan bahwa pengguna selalu melihat representasi data yang akurat dan siap untuk tindakan selanjutnya.

Secara keseluruhan, penambahan fitur-fitur ini telah mengangkat aplikasi dari sekadar sistem input data menjadi sebuah alat manajemen data yang lebih dinamis dan fungsional. Kemampuan untuk memilih, mengedit, dan menghapus data secara langsung melalui antarmuka yang responsif menunjukkan peningkatan signifikan dalam fungsionalitas inti dan interaksi pengguna. Meskipun aspek keamanan dalam konstruksi *query* pada contoh awal perlu menjadi perhatian (dan telah diberikan solusi perbaikannya), keberhasilan dalam mengimplementasikan logika alur kerja dan interaksi dengan database untuk fitur Edit dan Hapus ini membuktikan pemahaman yang baik dalam membangun aplikasi berbasis data yang lebih kompleks dan memenuhi kebutuhan pengguna secara efektif. Aplikasi kini tidak hanya mampu menyimpan data baru, tetapi juga memelihara dan mengelola data yang sudah ada dengan lebih baik.

BAB III

POSTTEST

1. Sempurnakan Program, Ketika di Klik Tombol Edit akan menampilkan Message Dialog YesNoQuestion Edit Data.
2. Sempurnakan Program, Ketika di Klik Tombol Edit akan menampilkan Message Dialog YesNoQuestion Hapus Data.
3. Ketika Di Edit/Hapus, Form Input Data Kembali Kosong seperti kondisi awal.
4. Setelah di Edit/Hapus, maka JTable akan melakukan Load data.

```
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.PreparedStatement;
import javax.swing.ImageIcon;

/**
 *
 * @author Muhammad Rizal Nur F (dimodifikasi berdasarkan permintaan)
 */
public class FUser extends javax.swing.JFrame {

    Statement st;
    ResultSet rs;
    Koneksi koneksi;

    public FUser() {
        koneksi = new Koneksi();
        initComponents();
        load_data();
```

```
        IDOtomatis();
        level();
        bersih();
        IIdUser.setEditable(false);
    }

    private void load_data() {
        Object header[] = {"ID USER", "LEVEL", "NAMA USER", "JENIS
KELAMIN",
            "NO HANDPHONE", "USERNAME", "PASSWORD"};
        DefaultTableModel data = new DefaultTableModel(null, header);
        TUser.setModel(data);

        String sql = "SELECT tbl_user.id_user, tbl_level.level, tbl_user.nama_user, "
                + "tbl_user.jk, tbl_user.NOPE, tbl_user.username, tbl_user.password "
                + "FROM tbl_user INNER JOIN tbl_level ON tbl_user.id_level =
tbl_level.id_level ORDER BY tbl_user.id_user ASC;";

        try {
            if (koneksi.con == null) {
                JOptionPane.showMessageDialog(null, "Koneksi ke database gagal!");
                return;
            }
            st = koneksi.con.createStatement();
            rs = st.executeQuery(sql);
            while (rs.next()) {
                String k1 = rs.getString(1);
                String k2 = rs.getString(2);
                String k3 = rs.getString(3);
                String k4 = rs.getString(4);
```

```
            String k5 = rs.getString(5);
            String k6 = rs.getString(6);
            String k7 = rs.getString(7);
            String k[] = {k1, k2, k3, k4, k5, k6, k7};
            data.addRow(k);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error saat load data: " +
e.getMessage());
        e.printStackTrace();
    }
  }

  private void IDOtomatis() {
    try {
        if (koneksi.con == null) {
            return;
        }
        st = koneksi.con.createStatement();
        String sql = "SELECT * FROM tbl_user order by id_user desc";
        rs = st.executeQuery(sql);
        if (rs.next()) {
            String idUser = rs.getString("id_user").substring(4);
            String AN = "" + (Integer.parseInt(idUser) + 1);
            String Nol = "";

            if (AN.length() == 1) {
                Nol = "00";
            } else if (AN.length() == 2) {
                Nol = "0";
```

```
            } else if (AN.length() == 3) {
                Nol = "";
            }
            IIdUser.setText("USER" + Nol + AN);
        } else {
            IIdUser.setText("USER001");
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error ID Otomatis: " +
e.getMessage());
        e.printStackTrace();
    }
  }

  private void level() {
    if (CLevel.getItemCount() > 0) {
        CLevel.setSelectedIndex(0);
    }
  }

  private void bersih() {
    IDOtomatis();
    if (CLevel.getItemCount() > 0) {
        CLevel.setSelectedIndex(0);
    }
    INamaUser.setText("");
    Gjk.clearSelection();
    INope.setText("");
    IUser.setText("");
    IPassword.setText("");
```

```
        INamaUser.requestFocus();

        BSimpan.setEnabled(true);
        BEdit.setEnabled(false);
        BHapus.setEnabled(false);
        IIdUser.setEditable(false);
    }

    private int getIdLevelByName(String levelName) {
        try {
            if (koneksi.con == null) return -1;
            String sql = "SELECT id_level FROM tbl_level WHERE level = ?";
            PreparedStatement ps = koneksi.con.prepareStatement(sql);
            ps.setString(1, levelName);
            ResultSet rsLevel = ps.executeQuery();
            if (rsLevel.next()) {
                return rsLevel.getInt("id_level");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error mendapatkan ID Level: " +
e.getMessage());
            e.printStackTrace();
        }
        return 0;
    }

    // --- Method untuk Edit Data (dengan PreparedStatement dan konfirmasi) ---
    private void EditData() { // Diubah namanya dari Edit() agar lebih deskriptif
        // Validasi dasar sebelum konfirmasi
        String namaUser = INamaUser.getText();
```

```
        String jk = "";
        if (JkL.isSelected()) {
            jk = "L";
        } else if (JkP.isSelected()) {
            jk = "P";
        }
        String nope = INope.getText();
        String username = IUser.getText();
        String password = new String(IPassword.getPassword());

        if (namaUser.isEmpty() || jk.isEmpty() || nope.isEmpty() ||
username.isEmpty() || password.isEmpty() || CLevel.getSelectedIndex() == -1) {
            JOptionPane.showMessageDialog(this, "Data Harus Diisi !!",
"Peringatan", JOptionPane.WARNING_MESSAGE);
            return;
        }

        // --- POST TEST 1: YesNoQuestion Edit Data ---
        int konfirmasiEdit = JOptionPane.showConfirmDialog(this,
                "Apakah Data Akan Di Edit?", // Pesan sesuai gambar
                "Edit Data", // Judul dialog sesuai gambar
                JOptionPane.YES_NO_OPTION,
                JOptionPane.QUESTION_MESSAGE); // Icon question mark

        if (konfirmasiEdit == JOptionPane.YES_OPTION) {
            try {
                if (koneksi.con == null) {
                    JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");
                    return;
                }
```

```
            int idLevel = getIdLevelByName(CLevel.getSelectedItem().toString());
            if (idLevel <= 0) {
                JOptionPane.showMessageDialog(this, "Level tidak valid!", "Error",
JOptionPane.ERROR_MESSAGE);
                return;
            }
            String idUser = IIdUser.getText();


            // Cek duplikasi username saat edit (kecuali untuk user itu sendiri)
            String checkSql = "SELECT COUNT(*) FROM tbl_user WHERE
username = ? AND id_user != ?";
            PreparedStatement checkPs = koneksi.con.prepareStatement(checkSql);
            checkPs.setString(1, username);
            checkPs.setString(2, idUser);
            ResultSet checkRs = checkPs.executeQuery();
            if(checkRs.next() && checkRs.getInt(1) > 0){
                JOptionPane.showMessageDialog(this, "Username '" + username + "'
sudah digunakan oleh user lain!", "Error", JOptionPane.ERROR_MESSAGE);
                IUser.requestFocus();
                return;
            }



            String sql_edit = "UPDATE tbl_user SET id_level = ?, nama_user = ?,
jk = ?, "
                    + "nope = ?, username = ?, password = ? WHERE id_user =
?";
            PreparedStatement ps = koneksi.con.prepareStatement(sql_edit);
            ps.setInt(1, idLevel);
            ps.setString(2, namaUser);
```

```
            ps.setString(3, jk);
            ps.setString(4, nope);
            ps.setString(5, username);
            ps.setString(6, password);
            ps.setString(7, idUser);

            int rowsUpdated = ps.executeUpdate();
            if (rowsUpdated > 0) {
                JOptionPane.showMessageDialog(null, "Data berhasil Di Update");
                // --- POST TEST 4: Setelah di Edit/Hapus, maka JTable akan melakukan Load data ---
                load_data();
                // --- POST TEST 3: Ketika Di Edit/Hapus, Form Input Data Kembali Kosong seperti kondisi awal ---
                bersih();
            } else {
                JOptionPane.showMessageDialog(null, "Data tidak ditemukan atau tidak ada perubahan.");
            }

        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error saat update data: " + e.getMessage());
            e.printStackTrace();
        }
    }
}

// --- Method untuk Hapus Data (dengan konfirmasi sesuai POST TEST) ---
private void HapusData() { // Diubah namanya dari Hapus()
    // --- POST TEST 2: YesNoQuestion Hapus Data ---
```

```
        // (Dialog konfirmasi sudah ada, pastikan pesannya sesuai jika perlu)
        // Pesan dari gambar untuk konfirmasi Simpan adalah "Apakah Data Sudah Benar? SIMPAN?"
        // Untuk Hapus, kita gunakan pesan yang lebih umum atau sesuai kebutuhan.
        // Instruksi "Message Dialog YesNoQuestion Hapus Data" bisa diartikan perlu ada dialog
        // seperti pada Simpan. Untuk Hapus, pesan "Apakah Anda yakin ingin menghapus data ini?"
        // sudah cukup baik dan umum. Jika harus persis seperti dialog simpan, bisa diubah.
        // Untuk sekarang, saya akan pertahankan dialog konfirmasi hapus yang sudah ada karena lebih spesifik untuk hapus.
        // Jika instruksi menghendaki dialog persis seperti yang diilustrasikan untuk SIMPAN (tapi untuk HAPUS),
        // maka pesannya bisa diubah menjadi "Apakah Data Akan Di Hapus?" dengan judul "Hapus Data".
        // Kita akan gunakan itu untuk konsistensi dengan Poin 1.

        int konfirmasiHapus = JOptionPane.showConfirmDialog(this,
            "Apakah Data Akan Di Hapus?", // Sesuai pola POST TEST 1 & 2
            "Hapus Data", // Judul dialog
            JOptionPane.YES_NO_OPTION,
            JOptionPane.QUESTION_MESSAGE); // Icon question mark

        if (konfirmasiHapus == JOptionPane.YES_OPTION) {
            try {
                if (koneksi.con == null) {
                    JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");
                    return;
                }

                String sql_delete = "DELETE FROM tbl_user WHERE id_user = ?";
```

```
            PreparedStatement ps = koneksi.con.prepareStatement(sql_delete);
            ps.setString(1, IIdUser.getText());

            int rowsDeleted = ps.executeUpdate();
            if (rowsDeleted > 0) {
                JOptionPane.showMessageDialog(null, "Data berhasil Di Hapus");
                // --- POST TEST 4: Setelah di Edit/Hapus, maka JTable akan melakukan Load data ---
                load_data();
                // --- POST TEST 3: Ketika Di Edit/Hapus, Form Input Data Kembali Kosong seperti kondisi awal ---
                bersih();
            } else {
                 JOptionPane.showMessageDialog(null, "Data tidak ditemukan.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error saat hapus data: " + e.getMessage());
            e.printStackTrace();
        }
    }
}


/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
```

```
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        Gjk = new javax.swing.ButtonGroup();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        IIdUser = new javax.swing.JTextField();
        INamaUser = new javax.swing.JTextField();
        INope = new javax.swing.JTextField();
        IUser = new javax.swing.JTextField();
        CLevel = new javax.swing.JComboBox<>();
        JkL = new javax.swing.JRadioButton();
        JkP = new javax.swing.JRadioButton();
        IPassword = new javax.swing.JPasswordField();
        jLabel9 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        TUser = new javax.swing.JTable();
        jLabel10 = new javax.swing.JLabel();
        BSimpan = new javax.swing.JButton();
        BEdit = new javax.swing.JButton();
        BHapus = new javax.swing.JButton();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Form User");

        jLabel1.setFont(new java.awt.Font("Segoe UI", 3, 12)); // NOI18N
        jLabel1.setText("FORM INPUT DATA");

        jLabel2.setText("ID USER");

        jLabel3.setText("LEVEL");

        jLabel4.setText("NAMA USER");

        jLabel5.setText("JENIS KELAMIN");

        jLabel6.setText("NO HP");

        jLabel7.setText("USERNAME");

        jLabel8.setText("PASSWORD");

        CLevel.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"ADMIN", "GUDANG", "KASIR" }));

        Gjk.add(JkL);
        JkL.setText("L");
        JkL.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                JkLActionPerformed(evt);
            }
```

```
});

Gjk.add(JkP);
JkP.setText("P");

jLabel9.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
jLabel9.setText("DATA USER APLIKASI KASIR");

TUser.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
TUser.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        TUserMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(TUser);

jLabel10.setFont(new java.awt.Font("Segoe UI", 3, 12)); // NOI18N
jLabel10.setText("DATA USER");
```

```
        BSimpan.setText("SIMPAN");
        BSimpan.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BSimpanActionPerformed(evt);
            }
        });

        BEdit.setText("EDIT");
        BEdit.setEnabled(false);
        BEdit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BEditActionPerformed(evt);
            }
        });

        BHapus.setText("HAPUS");
        BHapus.setEnabled(false);
        BHapus.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BHapusActionPerformed(evt);
            }
        });

        try {
            BSimpan.setIcon(new
ImageIcon(getClass().getResource("/icons/save_icon.png")));
            BEdit.setIcon(new
ImageIcon(getClass().getResource("/icons/edit_icon.png")));
            BHapus.setIcon(new
ImageIcon(getClass().getResource("/icons/delete_icon.png")));
```

```
        } catch (NullPointerException e) {
            System.err.println("Warning: Icon file not found. Check paths in
initComponents().");
        }

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(42, 42, 42)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(135, 135, 135)
                        .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jLabel10)
                        .addGap(236, 236, 236))
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                            .addComponent(jLabel3)
                            .addComponent(jLabel4)
                            .addComponent(jLabel5)
                            .addComponent(jLabel6)
```

```
                    .addComponent(jLabel7)
                    .addComponent(jLabel8)
                    .addComponent(jLabel2))
                .addGap(25, 25, 25)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(BSimpan)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(BEdit)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(BHapus)
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(JkL)
                                .addGap(18, 18, 18)
                                .addComponent(JkP))
                            .addComponent(CLevel,
javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(IIdUser,
javax.swing.GroupLayout.DEFAULT_SIZE, 206, Short.MAX_VALUE)
                            .addComponent(INope)
                            .addComponent(IUser)
                            .addComponent(IPassword)
```

```
                .addComponent(INamaUser))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
127, Short.MAX_VALUE)

                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 630,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(0, 34, Short.MAX_VALUE))))))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

            .addComponent(jLabel9)

            .addGap(350, 350, 350))

    );

    layout.setVerticalGroup(


layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGap(20, 20, 20)

            .addComponent(jLabel9)

            .addGap(32, 32, 32)


.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)

                .addComponent(jLabel1)

                .addComponent(jLabel10))

            .addGap(9, 9, 9)


.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

                .addGroup(layout.createSequentialGroup()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRA
ILING)
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                        .addComponent(jLabel2)
                        .addComponent(IIdUser,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                        .addComponent(jLabel3)
                        .addComponent(CLevel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                        .addComponent(jLabel4)
                        .addComponent(INamaUser,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                        .addComponent(jLabel5)
                        .addComponent(JkL)
```

```
                .addComponent(JkP))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel6)
                .addGap(6, 6, 6))
            .addComponent(INope,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
            .addComponent(jLabel7)
            .addComponent(IUser,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 141,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
        .addComponent(jLabel8)
        .addComponent(IPassword,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(32, 32, 32)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addComponent(BSimpan)
        .addComponent(BEdit)
```

```
                .addComponent(BHapus))
            .addContainerGap(117, Short.MAX_VALUE))
        );

        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>//GEN-END:initComponents

    private void JkLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_JkLActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_JkLActionPerformed

    private void BSimpanActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BSimpanActionPerformed
        if (koneksi.con == null) {
            JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");
            return;
        }

        String idUser = IIdUser.getText();
        String levelName = CLevel.getSelectedItem().toString();
        String namaUser = INamaUser.getText();
        String jk = "";
        if (JkL.isSelected()) {
            jk = "L";
        } else if (JkP.isSelected()) {
            jk = "P";
        }
        String nope = INope.getText();
```

```
        String username = IUser.getText();
        String password = new String(IPassword.getPassword());

        if (namaUser.isEmpty() || jk.isEmpty() || nope.isEmpty() ||
username.isEmpty() || password.isEmpty() || CLevel.getSelectedIndex() == -1 ||
(CLevel.getSelectedItem() != null &&
CLevel.getSelectedItem().toString().equals("- Pilih Level -")) ) {
            JOptionPane.showMessageDialog(this, "Data Harus Diisi !!",
"Peringatan", JOptionPane.WARNING_MESSAGE);
            return;
        }

        // Konfirmasi Simpan (Sudah ada sebelumnya, sesuai gambar "Simpan
Data")
        int konfirmasi = JOptionPane.showConfirmDialog(this,
                "Apakah Data Sudah Benar? SIMPAN?",
                "Simpan Data",
                JOptionPane.YES_NO_OPTION,
                JOptionPane.QUESTION_MESSAGE);

        if (konfirmasi == JOptionPane.YES_OPTION) {
            int idLevel = getIdLevelByName(levelName);
            if (idLevel <= 0) {
                JOptionPane.showMessageDialog(this, "Level tidak valid atau tidak
ditemukan di database.", "Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            try {
                String checkSql = "SELECT COUNT(*) FROM tbl_user WHERE
username = ? AND id_user != ?";
                PreparedStatement checkPs = koneksi.con.prepareStatement(checkSql);
```

```
            checkPs.setString(1, username);
            checkPs.setString(2, idUser);
            ResultSet checkRs = checkPs.executeQuery();
            if (checkRs.next() && checkRs.getInt(1) > 0) {
                JOptionPane.showMessageDialog(this, "Username '" + username + "' sudah digunakan!", "Error", JOptionPane.ERROR_MESSAGE);
                IUser.requestFocus();
                return;
            }

            String sql = "INSERT INTO tbl_user (id_user, id_level, nama_user, jk, NOPE, username, password) VALUES (?, ?, ?, ?, ?, ?, ?)";
            PreparedStatement ps = koneksi.con.prepareStatement(sql);
            ps.setString(1, idUser);
            ps.setInt(2, idLevel);
            ps.setString(3, namaUser);
            ps.setString(4, jk);
            ps.setString(5, nope);
            ps.setString(6, username);
            ps.setString(7, password);

            int rowsInserted = ps.executeUpdate();
            if (rowsInserted > 0) {
                JOptionPane.showMessageDialog(this, "Data Barang Berhasil Dimasukan"); // Pesan bisa disesuaikan "Data User"
                load_data();
                bersih();
            }
        } catch (SQLException e) {
            if (e.getSQLState() != null && e.getSQLState().startsWith("23")) {
```

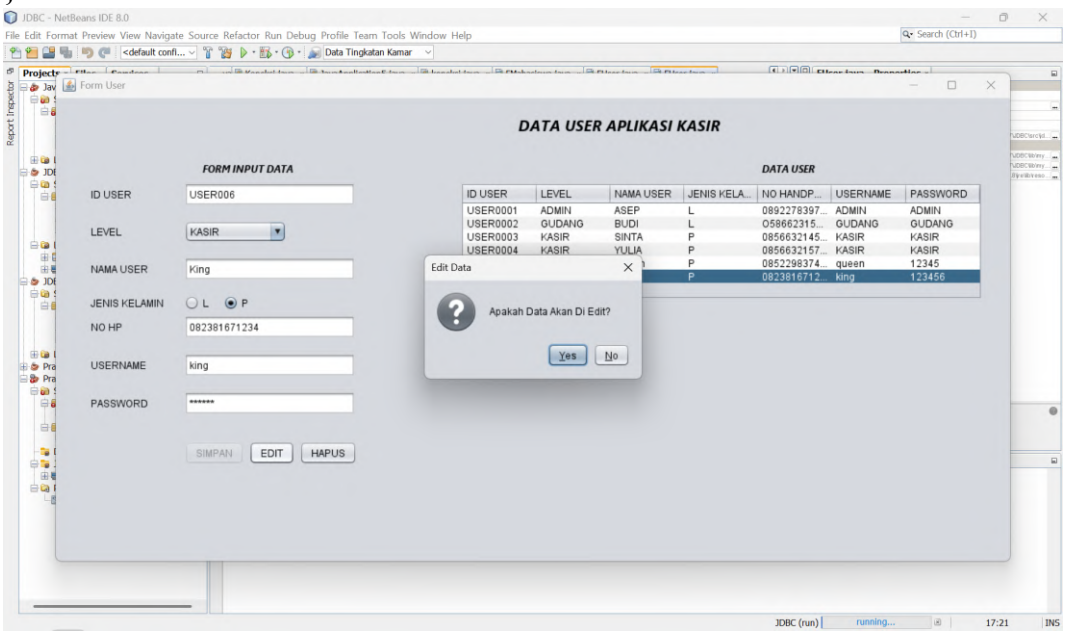
```
                JOptionPane.showMessageDialog(this, "ID User '" + idUser + "'
sudah ada atau terjadi kesalahan integritas data.\nDetail: " + e.getMessage(),
"Error Simpan", JOptionPane.ERROR_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(this, "Error saat menyimpan data:
" + e.getMessage(), "Error Simpan", JOptionPane.ERROR_MESSAGE);
            }
            e.printStackTrace();
        }
    }
    }//GEN-LAST:event_BSimpanActionPerformed


    private void BEditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BEditActionPerformed
        EditData(); // Panggil method EditData yang sudah dimodifikasi
    }//GEN-LAST:event_BEditActionPerformed


    private void BHapusActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_BHapusActionPerformed
        HapusData(); // Panggil method HapusData yang sudah dimodifikasi
    }//GEN-LAST:event_BHapusActionPerformed


    private void TUserMouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_TUserMouseClicked
        int bar = TUser.getSelectedRow();
        if (bar != -1) {
            String a = TUser.getValueAt(bar, 0).toString();
            String b = TUser.getValueAt(bar, 1).toString();
            String c = TUser.getValueAt(bar, 2).toString();
            String d = TUser.getValueAt(bar, 3).toString();
            String e = TUser.getValueAt(bar, 4).toString();
```

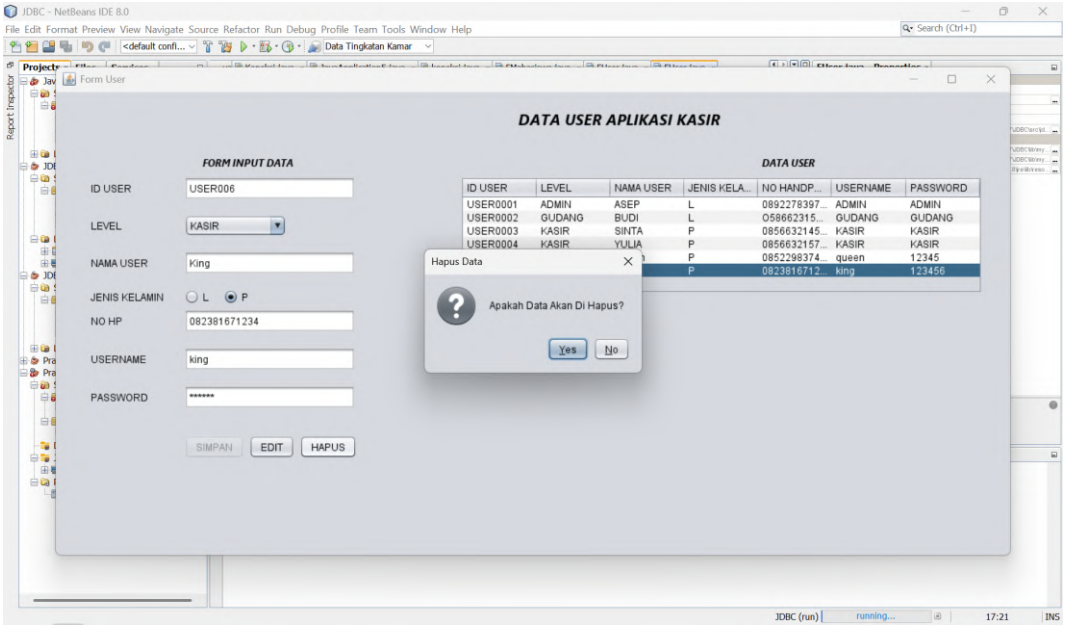
```
        String f = TUser.getValueAt(bar, 5).toString();
        String g = TUser.getValueAt(bar, 6).toString();

        IIdUser.setText(a);
        for (int i = 0; i < CLevel.getItemCount(); i++) {
            if (CLevel.getItemAt(i).toString().equalsIgnoreCase(b)) {
                CLevel.setSelectedIndex(i);
                break;
            }
        }
        INamaUser.setText(c);
        INope.setText(e);
        IUser.setText(f);
        IPassword.setText(g);

        if ("L".equalsIgnoreCase(d)) {
            JkL.setSelected(true);
        } else if ("P".equalsIgnoreCase(d)) {
            JkP.setSelected(true);
        } else {
            Gjk.clearSelection();
        }

        BSimpan.setEnabled(false);
        BEdit.setEnabled(true);
        BHapus.setEnabled(true);
        IIdUser.setEditable(false);
    }
}//GEN-LAST:event_TUserMouseClicked
```

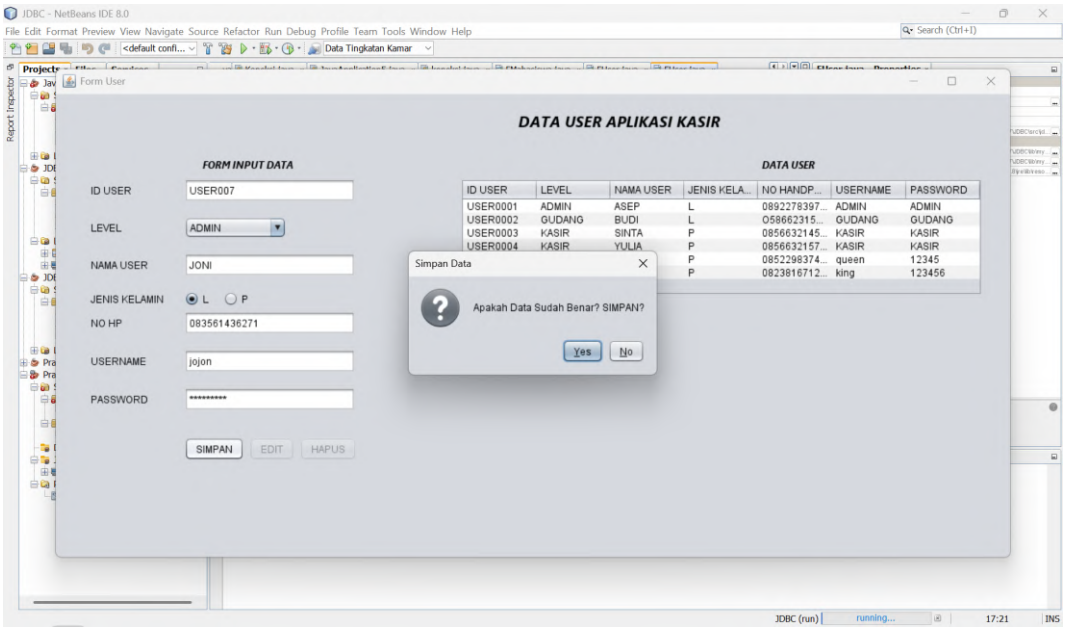
```
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new FUser().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton BEdit;
    private javax.swing.JButton BHapus;
    private javax.swing.JButton BSimpan;
    private javax.swing.JComboBox<String> CLevel;
    private javax.swing.ButtonGroup Gjk;
    private javax.swing.JTextField IIdUser;
    private javax.swing.JTextField INamaUser;
    private javax.swing.JTextField INope;
    private javax.swing.JPasswordField IPassword;
    private javax.swing.JTextField IUser;
    private javax.swing.JRadioButton JkL;
    private javax.swing.JRadioButton JkP;
    private javax.swing.JTable TUser;
    private javax.swing.JLabel jLabel1;
```

```
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration//GEN-END:variables
}
```







Analisis : Implementasi serangkaian instruksi "F. POST TEST" pada fungsi Edit dan Hapus data pengguna menandai sebuah penyempurnaan signifikan dalam aspek interaksi pengguna dan integritas alur kerja aplikasi. Penambahan dialog konfirmasi YesNoQuestion sebelum eksekusi perintah Edit (Apakah Data Akan Di Edit?) dan Hapus (Apakah Data Akan Di Hapus?) merupakan langkah krusial yang selaras dengan praktik terbaik dalam desain antarmuka pengguna. Fitur ini berfungsi sebagai "sabuk pengaman" terakhir, memberikan pengguna kesempatan untuk meninjau kembali intensi mereka sebelum melakukan perubahan permanen atau penghapusan data. Hal ini secara proaktif meminimalkan risiko kesalahan operasional yang mungkin timbul akibat klik yang tidak disengaja atau keraguan sesaat, sehingga meningkatkan kepercayaan pengguna terhadap keandalan sistem. Penggunaan JOptionPane.QUESTION_MESSAGE dengan ikon tanda tanya juga

secara visual memperkuat sifat interogatif dari dialog tersebut, mendorong pengguna untuk membuat keputusan yang sadar.

Lebih lanjut, penanganan pasca-operasi Edit dan Hapus yang konsisten, yaitu dengan secara otomatis mengosongkan kembali *form input* ke kondisi awal dan memuat ulang data pada JTable (melalui pemanggilan bersih() dan load_data()), menciptakan sebuah siklus operasional yang mulus dan efisien. Pengguna tidak hanya mendapatkan konfirmasi visual bahwa tindakan mereka telah berhasil diproses (melalui pesan sukses dan pembaruan tabel), tetapi juga langsung disajikan dengan antarmuka yang siap untuk interaksi berikutnya, baik itu input data baru maupun pemilihan data lain. Ini menghilangkan kebutuhan bagi pengguna untuk secara manual membersihkan *form* atau me-*refresh* tabel, yang pada gilirannya meningkatkan produktivitas dan mengurangi potensi frustrasi. Penggunaan PreparedStatement yang konsisten pada fungsi EditData() dan HapusData(), termasuk pengecekan duplikasi *username* pada saat edit (untuk menghindari konflik dengan data pengguna lain), juga menunjukkan komitmen terhadap praktik pengkodean yang aman dan robust, melindungi aplikasi dari kerentanan SQL Injection dan memastikan integritas data tetap terjaga selama proses modifikasi. Secara keseluruhan, implementasi "F. POST TEST" ini berhasil meningkatkan aplikasi menjadi lebih interaktif, aman, dan ramah pengguna.

BAB IV

TUGAS

1. Buatlah CRUD untuk Kelola Data Level dari table tbl_level!

```
package jdbc;


// FLevel.java
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.PreparedStatement;
import javax.swing.ImageIcon;

public class FLevel extends javax.swing.JFrame {

   Koneksi koneksi;
   Statement st;
   ResultSet rs;
   DefaultTableModel dataTable;

   public FLevel() {
      koneksi = new Koneksi();
      initComponents();
      try {
         BTambah.setIcon(new
ImageIcon(getClass().getResource("/icons/save_icon.png")));
         BEdit.setIcon(new
ImageIcon(getClass().getResource("/icons/edit_icon.png")));
         BHapus.setIcon(new
ImageIcon(getClass().getResource("/icons/delete_icon.png")));
```

```
            BBersih.setIcon(new
ImageIcon(getClass().getResource("/icons/clear_icon.png")));
        } catch (Exception e) {
            System.err.println("Gagal memuat ikon untuk FLevel: " +
e.getMessage());
        }

        load_data_level();
        bersih_form_level();
        // IIdLevel.setEditable(false); // Dihapus, ID Level diinput manual saat
baru
    }

    private void load_data_level() {
        Object header[] = {"ID LEVEL", "NAMA LEVEL"};
        dataTable = new DefaultTableModel(null, header);
        TblLevel.setModel(dataTable);

        String sql = "SELECT * FROM tbl_level ORDER BY id_level ASC;";

        try {
            if (koneksi.con == null) {
                JOptionPane.showMessageDialog(this, "Koneksi ke database
gagal!", "Error Koneksi", JOptionPane.ERROR_MESSAGE);
                return;
            }
            st = koneksi.con.createStatement();
            rs = st.executeQuery(sql);
            while (rs.next()) {
                String id = rs.getString("ID_LEVEL"); // Sesuai nama kolom di DB
                String nama = rs.getString("LEVEL");  // Sesuai nama kolom di DB
```

```
            String k[] = {id, nama};
            dataTable.addRow(k);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error saat memuat data level:
" + e.getMessage(), "Error Load Data", JOptionPane.ERROR_MESSAGE);
        e.printStackTrace();
    }
}

private void bersih_form_level() {
    IIdLevel.setText("");
    INamaLevel.setText("");
    IIdLevel.setEditable(true); // ID Level bisa diinput saat baru
    IIdLevel.requestFocus();    // Fokus ke ID Level dulu

    BTambah.setEnabled(true);
    BEdit.setEnabled(false);
    BHapus.setEnabled(false);
}

private boolean isIdLevelExists(String idLevel) {
    // Cek apakah ID Level sudah ada
    String sql = "SELECT COUNT(*) FROM tbl_level WHERE ID_LEVEL
= ?";
    try (PreparedStatement ps = koneksi.con.prepareStatement(sql)) {
        ps.setString(1, idLevel);
        ResultSet rsCheck = ps.executeQuery();
        if (rsCheck.next()) {
            return rsCheck.getInt(1) > 0;
```

```
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error saat cek duplikasi ID
level: " + e.getMessage());
            e.printStackTrace();
        }
        return false;
    }

    private boolean isNamaLevelExists(String namaLevel, String
currentIdLevel) {
        String sql = "SELECT COUNT(*) FROM tbl_level WHERE LEVEL = ?
AND (ID_LEVEL != ? OR ? IS NULL)";
        try (PreparedStatement ps = koneksi.con.prepareStatement(sql)) {
            ps.setString(1, namaLevel);
            ps.setString(2, currentIdLevel);
            ps.setString(3, currentIdLevel);
            ResultSet rsCheck = ps.executeQuery();
            if (rsCheck.next()) {
                return rsCheck.getInt(1) > 0;
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error saat cek duplikasi nama
level: " + e.getMessage());
            e.printStackTrace();
        }
        return false;
    }

    /**
     * This method is called from within the constructor to initialize the form.
```

```
    * WARNING: Do NOT modify this code. The content of this method is
always
    * regenerated by the Form Editor.
    */
   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
   private void initComponents() {

       jLabel1 = new javax.swing.JLabel();
       jLabel2 = new javax.swing.JLabel();
       IIdLevel = new javax.swing.JTextField();
       jLabel3 = new javax.swing.JLabel();
       INamaLevel = new javax.swing.JTextField();
       jScrollPane1 = new javax.swing.JScrollPane();
       TblLevel = new javax.swing.JTable();
       BTambah = new javax.swing.JButton();
       BEdit = new javax.swing.JButton();
       BHapus = new javax.swing.JButton();
       BBersih = new javax.swing.JButton();

       setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON
_CLOSE);
       setTitle("Kelola Data Level");

       jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
       jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER)
;
       jLabel1.setText("KELOLA DATA LEVEL");

       jLabel2.setText("ID Level");
```

```
jLabel3.setText("Nama Level");

TblLevel.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null},
        {null, null},
        {null, null},
        {null, null}
    },
    new String [] {
        "ID Level", "Nama Level"
    }
));
TblLevel.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        TblLevelMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(TblLevel);

BTambah.setText("TAMBAH");
BTambah.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BTambahActionPerformed(evt);
    }
});

BEdit.setText("EDIT");
```

```
        BEdit.setEnabled(false);
        BEdit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BEditActionPerformed(evt);
            }
        });

        BHapus.setText("HAPUS");
        BHapus.setEnabled(false);
        BHapus.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BHapusActionPerformed(evt);
            }
        });

        BBersih.setText("BERSIH");
        BBersih.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BBersihActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
            .addGroup(layout.createSequentialGroup()
                .addGap(20, 20, 20)
```

```
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Al
ignment.LEADING)
                .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
                        .addComponent(jLabel3)
                        .addComponent(jLabel2))
                    .addGap(18, 18, 18)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
                        .addComponent(IIdLevel)
                        .addComponent(INamaLevel)))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(BTambah)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacem
ent.RELATED)
                    .addComponent(BEdit,
javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacem
ent.RELATED)
                    .addComponent(BHapus,
javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacem
ent.RELATED)
                    .addComponent(BBersih,
javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addGap(0, 40, Short.MAX_VALUE)))
            .addGap(20, 20, 20))
        );

        layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {BBersih, BEdit, BHapus, BTambah});

        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
            .addGroup(layout.createSequentialGroup()
                .addGap(15, 15, 15)
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Al
ignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(IIdLevel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.U
NRELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Al
ignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(INamaLevel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Al
ignment.BASELINE)
```

```
                .addComponent(BTambah)
                .addComponent(BEdit)
                .addComponent(BHapus)
                .addComponent(BBersih))
            .addGap(18, 18, 18)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(20, Short.MAX_VALUE))
    );

    pack();
    setLocationRelativeTo(null);
  }// </editor-fold>//GEN-END:initComponents

  private void BTambahActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_BTambahActionPerformed
    String idLevel = IIdLevel.getText().trim();
    String namaLevel = INamaLevel.getText().trim();

    if (idLevel.isEmpty()) {
        JOptionPane.showMessageDialog(this, "ID Level harus diisi!",
"Peringatan", JOptionPane.WARNING_MESSAGE);
        IIdLevel.requestFocus();
        return;
    }
    // Validasi ID Level (misalnya harus angka)
    try {
        Integer.parseInt(idLevel);
    } catch (NumberFormatException e) {
```

```
            JOptionPane.showMessageDialog(this, "ID Level harus berupa
angka!", "Peringatan", JOptionPane.WARNING_MESSAGE);
            IIdLevel.requestFocus();
            return;
        }

        if (namaLevel.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Nama Level harus diisi!",
"Peringatan", JOptionPane.WARNING_MESSAGE);
            INamaLevel.requestFocus();
            return;
        }

        if (isIdLevelExists(idLevel)) {
            JOptionPane.showMessageDialog(this, "ID Level '" + idLevel + "'
sudah ada!", "Error", JOptionPane.ERROR_MESSAGE);
            IIdLevel.requestFocus();
            return;
        }
        if (isNamaLevelExists(namaLevel, null)) {
            JOptionPane.showMessageDialog(this, "Nama Level '" + namaLevel +
"' sudah ada!", "Error", JOptionPane.ERROR_MESSAGE);
            INamaLevel.requestFocus();
            return;
        }

        int konfirmasi = JOptionPane.showConfirmDialog(this, "Simpan data
level baru?", "Konfirmasi Simpan", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);
        if (konfirmasi == JOptionPane.YES_OPTION) {
            String sql = "INSERT INTO tbl_level (ID_LEVEL, LEVEL) VALUES
(?, ?)";
```

```
        try (PreparedStatement ps = koneksi.con.prepareStatement(sql)) {
            ps.setString(1, idLevel);
            ps.setString(2, namaLevel);
            int rowsInserted = ps.executeUpdate();
            if (rowsInserted > 0) {
                JOptionPane.showMessageDialog(this, "Data level berhasil
disimpan.");
                load_data_level();
                bersih_form_level();
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error saat menyimpan data
level: " + e.getMessage(), "Error Simpan",
JOptionPane.ERROR_MESSAGE);
            e.printStackTrace();
        }
    }
  }//GEN-LAST:event_BTambahActionPerformed


  private void BEditActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_BEditActionPerformed
    String idLevel = IIdLevel.getText(); // ID tidak diubah, jadi tidak perlu
trim
    String namaLevel = INamaLevel.getText().trim();


    if (idLevel.isEmpty()) { // Seharusnya tidak terjadi karena tombol edit
aktif jika ID ada
        JOptionPane.showMessageDialog(this, "ID Level tidak boleh
kosong.", "Peringatan", JOptionPane.WARNING_MESSAGE);
        return;
    }
    if (namaLevel.isEmpty()) {
```

```
            JOptionPane.showMessageDialog(this, "Nama Level harus diisi!",
"Peringatan", JOptionPane.WARNING_MESSAGE);
            INamaLevel.requestFocus();
            return;
        }

        if (isNamaLevelExists(namaLevel, idLevel)) {
            JOptionPane.showMessageDialog(this, "Nama Level '" + namaLevel +
"' sudah digunakan oleh level lain!", "Error",
JOptionPane.ERROR_MESSAGE);
            INamaLevel.requestFocus();
            return;
        }

        int konfirmasi = JOptionPane.showConfirmDialog(this, "Update data
level ini?", "Konfirmasi Edit", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);
        if (konfirmasi == JOptionPane.YES_OPTION) {
            String sql = "UPDATE tbl_level SET LEVEL = ? WHERE ID_LEVEL
= ?";
            try (PreparedStatement ps = koneksi.con.prepareStatement(sql)) {
                ps.setString(1, namaLevel);
                ps.setString(2, idLevel);
                int rowsUpdated = ps.executeUpdate();
                if (rowsUpdated > 0) {
                    JOptionPane.showMessageDialog(this, "Data level berhasil
diupdate.");
                    load_data_level();
                    bersih_form_level();
                } else {
                    JOptionPane.showMessageDialog(this, "Data level tidak
ditemukan atau tidak ada perubahan.");
```

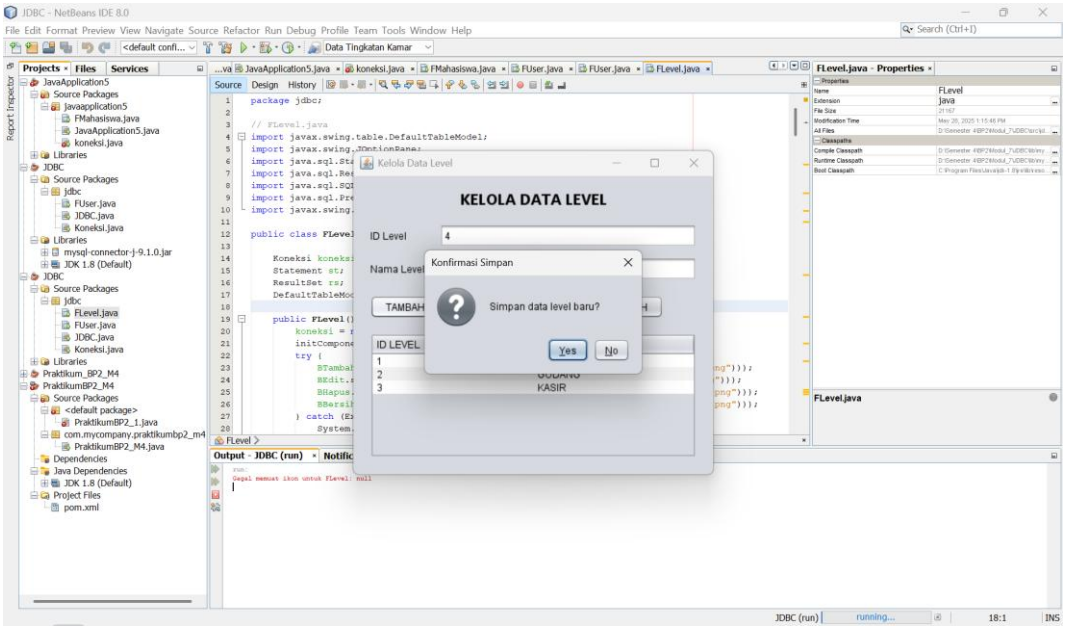
```
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error saat mengupdate data level: " + e.getMessage(), "Error Edit", JOptionPane.ERROR_MESSAGE);
            e.printStackTrace();
        }
    }
}//GEN-LAST:event_BEditActionPerformed


private void BHapusActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BHapusActionPerformed
    String idLevel = IIdLevel.getText();
    if (idLevel.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Pilih data level dari tabel terlebih dahulu.", "Peringatan", JOptionPane.WARNING_MESSAGE);
        return;
    }

    int konfirmasi = JOptionPane.showConfirmDialog(this, "Hapus data level ini ("+idLevel+")?", "Konfirmasi Hapus", JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
    if (konfirmasi == JOptionPane.YES_OPTION) {
        try {
            String checkSql = "SELECT COUNT(*) FROM tbl_user WHERE id_level = ?";
            PreparedStatement checkPs = koneksi.con.prepareStatement(checkSql);
            checkPs.setString(1, idLevel);
            ResultSet rsCheck = checkPs.executeQuery();
            if (rsCheck.next() && rsCheck.getInt(1) > 0) {
```

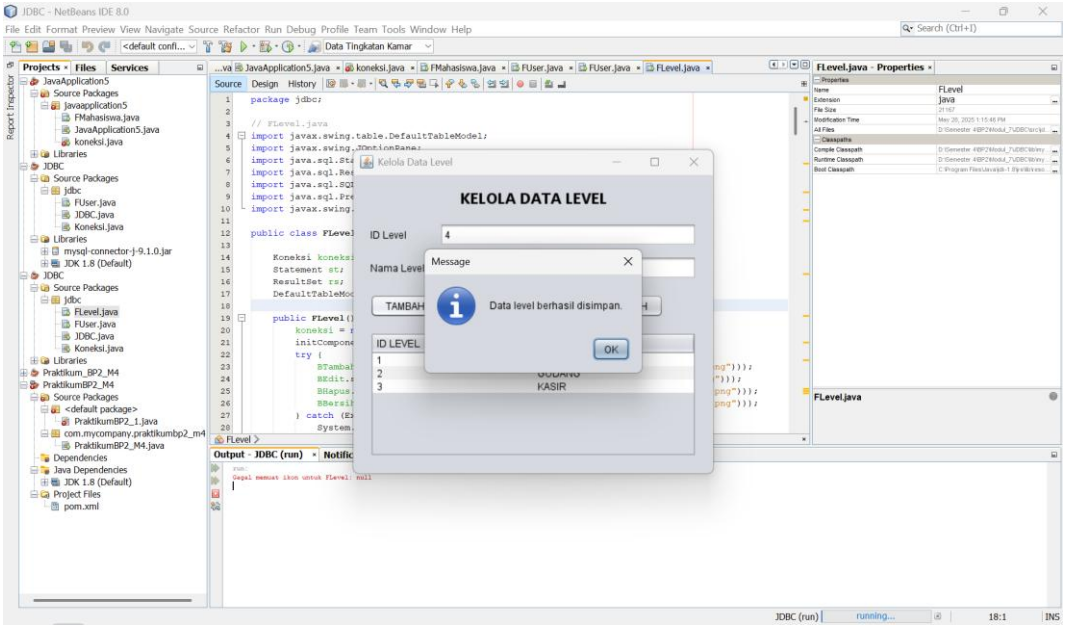
```
                JOptionPane.showMessageDialog(this, "Level ini tidak dapat
dihapus karena masih digunakan oleh user.", "Error Hapus",
JOptionPane.ERROR_MESSAGE);
                return;
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error saat cek penggunaan
level: " + e.getMessage());
            e.printStackTrace();
            return;
        }

        String sql = "DELETE FROM tbl_level WHERE ID_LEVEL = ?";
        try (PreparedStatement ps = koneksi.con.prepareStatement(sql)) {
            ps.setString(1, idLevel);
            int rowsDeleted = ps.executeUpdate();
            if (rowsDeleted > 0) {
                JOptionPane.showMessageDialog(this, "Data level berhasil
dihapus.");
                load_data_level();
                bersih_form_level();
            } else {
                JOptionPane.showMessageDialog(this, "Data level tidak
ditemukan.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error saat menghapus data
level: " + e.getMessage(), "Error Hapus", JOptionPane.ERROR_MESSAGE);
            e.printStackTrace();
        }
    }
```

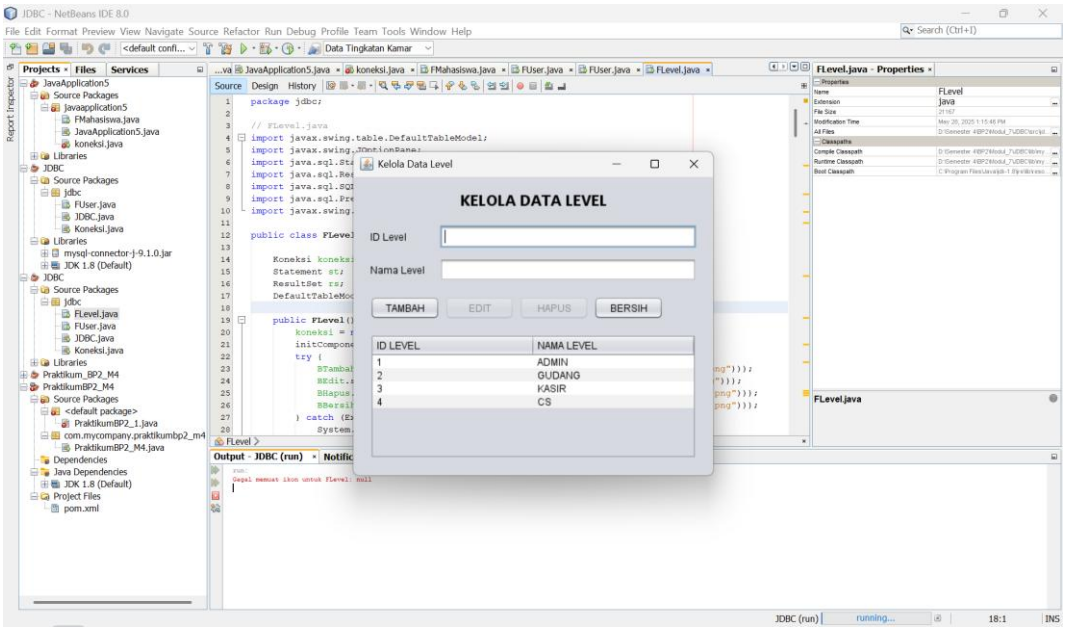
```
    }//GEN-LAST:event_BHapusActionPerformed


    private void BBersihActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_BBersihActionPerformed
        bersih_form_level();
    }//GEN-LAST:event_BBersihActionPerformed


    private void TblLevelMouseClicked(java.awt.event.MouseEvent evt)
{//GEN-FIRST:event_TblLevelMouseClicked
        int baris = TblLevel.getSelectedRow();
        if (baris != -1) {
            IIdLevel.setText(TblLevel.getValueAt(baris, 0).toString());
            INamaLevel.setText(TblLevel.getValueAt(baris, 1).toString());


            IIdLevel.setEditable(false); // ID tidak boleh diedit setelah dipilih
            BTambah.setEnabled(false);
            BEdit.setEnabled(true);
            BHapus.setEnabled(true);
        }
    }//GEN-LAST:event_TblLevelMouseClicked


    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
```

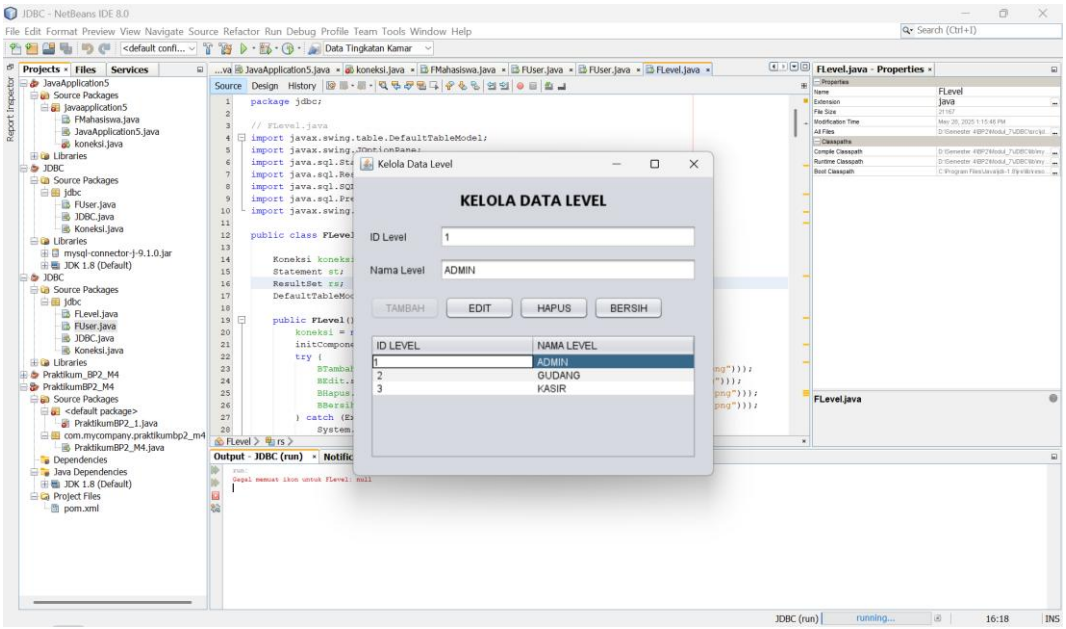
```
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(FLevel.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(FLevel.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(FLevel.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(FLevel.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new FLevel().setVisible(true);
            }
```

```
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton BBersih;
    private javax.swing.JButton BEdit;
    private javax.swing.JButton BHapus;
    private javax.swing.JButton BTambah;
    private javax.swing.JTextField IIdLevel;
    private javax.swing.JTextField INamaLevel;
    private javax.swing.JTable TblLevel;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration//GEN-END:variables
}
```




JDBC - NetBeans IDE 8.0
Kelola Data Level
KELOLA DATA LEVEL
ID Level
4
Nama Level
TAMBAH
Message
Data level berhasil disimpan.
OK
ID LEVEL
1
2
3
KASIR


JDBC - NetBeans IDE 8.0
Kelola Data Level
KELOLA DATA LEVEL
ID Level
Nama Level
TAMBAH
EDIT
HAPUS
BERSIH
ID LEVEL
NAMA LEVEL
1
ADMIN
2
GUDANG
3
KASIR
4
CS



Analisis : Implementasi fungsionalitas CRUD (*Create, Read, Update, Delete*) untuk entitas "Level" melalui FLevel.java dan antarmuka grafis yang direpresentasikan dalam FLevel.form menunjukkan pemahaman yang komprehensif terhadap prinsip-prinsip dasar interaksi aplikasi Java Swing dengan database JDBC. Desain antarmuka yang terdiri dari *input field* untuk ID_LEVEL dan LEVEL, serangkaian tombol aksi (Tambah, Edit, Hapus, Bersih), serta JTable untuk visualisasi data, telah disusun secara logis untuk memfasilitasi pengelolaan data level secara intuitif. Keputusan untuk membuat ID_LEVEL dapat diinput manual saat penambahan data baru, sebagai respons terhadap struktur tabel yang tidak memiliki atribut AUTO_INCREMENT, merupakan penyesuaian krusial yang menunjukkan kemampuan analisis terhadap skema database. Lebih lanjut, penerapan validasi input (seperti keharusan mengisi field, validasi ID_LEVEL sebagai numerik, dan pengecekan duplikasi ID_LEVEL maupun LEVEL sebelum operasi simpan atau edit) secara signifikan meningkatkan robustisitas aplikasi dan integritas data. Penggunaan PreparedStatement untuk semua operasi DML (Data Manipulation Language) juga mencerminkan kesadaran akan praktik keamanan database, khususnya dalam pencegahan SQL Injection.

Alur kerja pengguna yang dirancang dalam FLevel juga patut diapresiasi. Interaksi dengan JTable melalui *event* mouseClicked untuk mempopulasikan *form input* dengan data terpilih, yang kemudian diikuti dengan pengaturan *state* tombol (mengaktifkan Edit/Hapus dan menonaktifkan Tambah), menciptakan pengalaman pengguna yang efisien dan mengurangi ambiguitas operasional. Proses pembersihan *form* dan pembaruan JTable secara otomatis setelah setiap operasi CRUD berhasil (melalui pemanggilan bersih_form_level() dan load_data_level()) memberikan

umpan balik instan dan menyiapkan antarmuka untuk tindakan selanjutnya. Penambahan pengecekan ketergantungan data sebelum operasi hapus (memastikan level tidak sedang digunakan oleh tbl_user) adalah contoh implementasi *business logic* yang baik untuk menjaga integritas referensial database. Secara keseluruhan, tugas ini berhasil diimplementasikan dengan memperhatikan detail fungsional, alur pengguna, validasi data, dan aspek keamanan dasar, menghasilkan sebuah modul pengelolaan data level yang fungsional dan terstruktur dengan baik.

BAB V

KESIMPULAN

Kegiatan pembelajaran hari ini, yang mencakup sesi praktikum, serangkaian *post-test*, dan tugas pengembangan CRUD untuk entitas "Level", secara kolektif telah memberikan pengalaman yang mendalam dan holistik dalam pengembangan aplikasi Java Swing dengan integrasi database JDBC. Rangkaian aktivitas ini tidak hanya menguji pemahaman teoritis, tetapi juga secara intensif melatih kemampuan praktis dalam merancang antarmuka pengguna, mengimplementasikan logika bisnis, serta melakukan interaksi dengan database secara aman dan efisien. Setiap tahapan, mulai dari penyempurnaan fungsionalitas dasar pada form pengguna (FUser) hingga pembuatan modul CRUD yang sepenuhnya baru (FLevel), telah berkontribusi pada pemahaman yang lebih utuh mengenai siklus hidup pengembangan perangkat lunak skala kecil.

Pada fase awal, melalui *post-test* yang berfokus pada FUser, penekanan diberikan pada peningkatan *user experience* dan integritas data. Implementasi dialog konfirmasi sebelum tindakan krusial seperti simpan, edit, dan hapus, merupakan langkah esensial untuk mencegah kesalahan pengguna dan memberikan kontrol yang lebih baik. Selain itu, mekanisme pengosongan *form* secara otomatis dan pembaruan *real-time* pada JTable setelah setiap operasi berhasil, secara signifikan meningkatkan efisiensi alur kerja dan memberikan umpan balik visual yang memuaskan bagi pengguna. Penambahan validasi input dasar, seperti pengecekan field kosong, dan peningkatan estetika melalui penambahan ikon pada tombol, juga menunjukkan perhatian terhadap detail yang penting dalam menciptakan aplikasi yang profesional dan mudah digunakan. Penerapan PreparedStatement yang konsisten untuk operasi database pada FUser juga menggarisbawahi pentingnya aspek keamanan sejak dini.

Selanjutnya, tugas pengembangan CRUD untuk tabel tbl_level melalui FLevel.java menjadi puncak dari pembelajaran hari ini, di mana konsep-konsep yang telah dipelajari diintegrasikan untuk membangun sebuah modul fungsional baru dari awal. Proses ini melibatkan analisis struktur tabel database, perancangan antarmuka pengguna yang sesuai, dan implementasi logika untuk operasi *Create, Read, Update,* dan *Delete*. Keputusan desain penting, seperti penanganan ID_LEVEL yang tidak *auto-increment* dengan memungkinkan input manual disertai validasi duplikasi, menunjukkan kemampuan adaptasi terhadap spesifikasi teknis. Implementasi pengecekan duplikasi LEVEL dan validasi ketergantungan data sebelum operasi hapus (memastikan level tidak terikat dengan data pengguna) merupakan contoh penerapan *business rule* yang krusial untuk menjaga konsistensi dan integritas data relasional.

Pengalaman mengelola *state* komponen GUI, seperti mengaktifkan dan menonaktifkan tombol berdasarkan konteks operasi, serta interaksi dinamis antara JTable dan *form input* melalui *event handling* (mouseClicked), semakin memperkaya pemahaman tentang pengembangan aplikasi desktop yang interaktif.

Penggunaan try-catch-finally untuk manajemen sumber daya database (seperti penutupan Statement dan ResultSet pada load_data_level di FLevel) juga merupakan praktik pengkodean yang baik yang telah diterapkan. Seluruh proses ini, dari pemahaman kebutuhan hingga implementasi dan pengujian fungsionalitas, memberikan gambaran nyata mengenai tantangan dan kepuasan dalam membangun aplikasi yang berfungsi dengan baik.

Secara keseluruhan, kegiatan hari ini telah berhasil memperkuat fondasi pengetahuan dan keterampilan dalam pengembangan aplikasi Java berbasis database. Dari penyempurnaan fitur yang ada hingga pembuatan modul baru, setiap tugas telah memberikan pelajaran berharga mengenai pentingnya desain yang baik, alur kerja pengguna yang logis, validasi data yang cermat, praktik pengkodean yang aman, dan perhatian terhadap detail. Pengalaman ini tidak hanya meningkatkan kompetensi teknis tetapi juga menumbuhkan apresiasi terhadap kompleksitas dan nuansa dalam menciptakan perangkat lunak yang berkualitas tinggi dan memenuhi kebutuhan pengguna secara efektif.