

**LAPORAN PRAKTIKUM**  
**BAHASA PEMROGRAMAN 2**  
**MODUL 6**

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

DAFTAR ISI .....	ii
BAB I .....	1
PRETEST .....	1
Jelaskan Package JDBC API di bawah ini: .....	1
BAB II .....	2
PRAKTIKUM .....	2
1. Input Data .....	2
BAB III .....	24
POSTTEST .....	24
1. Sempurnakan Program. Ketika di Klik Tombol Simpan akan menampilkan YesNoQuestion penyimpanan Data. ....	24
2. Ketika Disimpan Juga, Form Input Data Kembali Kosong seperti kondisi awal. ....	24
3. Setelah Disimpan, Data yang dimasukkan akan langsung tampil pada Table. ....	24
BAB IV .....	46
TUGAS .....	46
1. Buatlah Event Handler, untuk kondisi ketika Form Input Data masih kosong, dan button simpan di klik akan memberikan pesan “Data Harus Diisi”! .....	46
2. Lengkapi Button Simpan, Edit, Delete dengan Icon yang sesuai. ....	46
BAB V .....	68
KESIMPULAN .....	68

## **BAB I**

### **PRETEST**

Jelaskan Package JDBC API di bawah ini:

- `java.sql.Connection`
- `java.sql.DriverManager`
- `java.sql.Statement`
- `java.sql.ResultSet`

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

1. **`java.sql.Connection`**

Adalah antarmuka (interface) yang mewakili koneksi ke database. Objek `Connection` digunakan untuk membuat pernyataan SQL, mengelola transaksi, dan mengatur properti koneksi database.

2. **`java.sql.DriverManager`**

Merupakan class yang digunakan untuk mengatur set driver JDBC. Fungsinya adalah untuk membuat koneksi ke database melalui method `getConnection()`. `DriverManager` akan memilih driver yang sesuai dari daftar driver yang telah terdaftar.

3. **`java.sql.Statement`**

Adalah antarmuka yang digunakan untuk mengeksekusi perintah SQL statis (seperti `SELECT`, `INSERT`, `UPDATE`, `DELETE`) terhadap database. Objek `Statement` dapat digunakan untuk mengirim perintah SQL ke database.

4. **`java.sql.ResultSet`**

Adalah antarmuka yang digunakan untuk menangani hasil query `SELECT`. Objek `ResultSet` menyimpan data hasil dari eksekusi perintah SQL, dan menyediakan metode untuk menelusuri dan mengambil data dari hasil tersebut.

## **BAB II**

### **PRAKTIKUM**

#### 1. Input 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; // Ditambahkan untuk PreparedStatement

/**
 *
 * @author Muhammad Rizal Nur F
 */
public class FUser extends javax.swing.JFrame {

    Statement st;

    ResultSet rs;

    Koneksi koneksi;

    // Ganti jTextField2 menjadi INamaUser (Anda mungkin perlu melakukannya
    di Design View NetBeans)
```

// Jika tidak, pastikan Anda telah mengganti nama variabel jTextField2 menjadi INamaUser di bagian deklarasi variabel di bawah.

// Untuk contoh ini, saya akan mengasumsikan Anda telah mengubahnya menjadi INamaUser.

```
public FUser() {
    koneksi = new Koneksi();
    initComponents();
    load_data();
    IDOtomatis(); // Panggil IDOtomatis
    level();      // Panggil level (jika Anda ingin mengisi CLevel dari DB)
                // Jika tidak, biarkan CLevel diisi manual seperti di initComponents
    bersih();     // Bersihkan form dan siapkan ID baru
    IIdUser.setEditable(false); // ID User tidak boleh diedit manual
}

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;"; // Tambahkan ORDER BY

    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();
}
}

// Implementasi dari gambar
private void IDOtomatis() {
    try {
        if (koneksi.con == null) {
            JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");
            return;
        }
        st = koneksi.con.createStatement();

```

```

// Gunakan id_user sesuai dengan yang ada di load_data
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); // Ambil angka
    setelah "USER"

    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 = "";
    }
    // Sesuaikan prefix jika berbeda, di gambar adalah "USER"
    IidUser.setText("USER" + Nol + AN);
} else {
    IidUser.setText("USER001");
}
} catch (SQLException e) {
    JOptionPane.showMessageDialog(null, "Error ID Otomatis: " +
e.getMessage());
    e.printStackTrace();
}
}

// Method untuk mengisi ComboBox Level (jika dari database)

```

// Jika Anda sudah mengisi CLevel secara manual di initComponents, method ini bisa di-skip

// atau disesuaikan. Untuk saat ini, kita anggap CLevel sudah diisi dari Design.

```
private void level() {  
    // CLevel.removeAllItems(); // Hapus item lama jika diisi dari DB  
    // CLevel.addItem("- Pilih Level -"); // Item default  
    // try {  
    //     if (koneksi.con == null) return;  
    //     st = koneksi.con.createStatement();  
    //     String sql = "SELECT level FROM tbl_level ORDER BY level ASC";  
    //     rs = st.executeQuery(sql);  
    //     while(rs.next()){  
    //         CLevel.addItem(rs.getString("level"));  
    //     }  
    // } catch (SQLException e) {  
    //     JOptionPane.showMessageDialog(null, "Error memuat level:  
"+e.getMessage());  
    // }  
    // Karena CLevel sudah diisi di initComponents, kita bisa biarkan kosong  
atau  
    // hanya memastikan ada item yang terpilih jika perlu.  
    if (CLevel.getItemCount() > 0) {  
        CLevel.setSelectedIndex(0); // Pilih item pertama (misal "ADMIN")  
    }  
}
```

```
private void bersih() {  
    // IIdUser.setText(""); // ID User akan di-set oleh IDOtomatis()  
    if (CLevel.getItemCount() > 0) {  
        CLevel.setSelectedIndex(0);  
    }  
}
```



```

    }

    INamaUser.setText(""); // Pastikan jTextField2 sudah di-rename menjadi
    INamaUser

    Gjk.clearSelection(); // Membersihkan pilihan radio button

    INope.setText("");

    IUser.setText("");

    IPassword.setText("");

    INamaUser.requestFocus(); // Fokus ke Nama User

    IDOtomatis(); // Panggil lagi untuk ID baru setelah simpan/bersih
}

private int getIdLevelByName(String levelName) {
    // Fungsi ini untuk mendapatkan id_level berdasarkan nama level
    // Ini asumsi, sesuaikan dengan struktur tabel tbl_level Anda
    // Misalnya: ADMIN -> 1, GUDANG -> 2, KASIR -> 3
    // Cara yang lebih baik adalah query ke tbl_level
    try {
        if (koneksi.con == null) return -1; // Indikasi error

        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 atau error (sebaiknya tidak terjadi jika data konsisten)
```

```
// Atau bisa juga return -1 dan handle errornya
```

```
if (levelName.equalsIgnoreCase("ADMIN")) return 1;
```

```
if (levelName.equalsIgnoreCase("GUDANG")) return 2;
```

```
if (levelName.equalsIgnoreCase("KASIR")) return 3;
```

```
return 0; // Default atau error
```

```
}
```

```
/**
```

```
 * 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(); // Diganti dari jTextField2
INope = new javax.swing.JTextField();
IUser = new javax.swing.JTextField();
CLevel = new javax.swing.JComboBox<String>();
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(); // Diganti dari jButton1
BEdit = new javax.swing.JButton(); // Diganti dari jButton2
BHapus = new javax.swing.JButton(); // Diganti dari jButton3

Gjk.add(JkL);
Gjk.add(JkP);

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Form User"); // Menambahkan Judul Window

```

```

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" }));
```

```
JkL.setText("L");
```

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

```
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"
    }
));
jScrollPane1.setViewportViewView(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.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BEditActionPerformed(evt);
    }
});

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

```

```

    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.LEADING)

        .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.LEADING, 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)) // Diganti dari jTextField2

.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)) // Disesuaikan agar lebih ke tengah
    );
    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.BASELINE)
```

```
    .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.BASELINE)
```

```
    .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.BASELINE)
```

```
    .addComponent(jLabel4)
```

```
    .addComponent(INamaUser,  
        javax.swing.GroupLayout.PREFERRED_SIZE,  
        javax.swing.GroupLayout.DEFAULT_SIZE,  
        javax.swing.GroupLayout.PREFERRED_SIZE)) // Diganti dari jTextField2
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .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); // Menengahkan window saat muncul
    } // </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(); // Pastikan ini INamaUser

        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());

```

```

// Validasi input dasar

if (namaUser.isEmpty() || jk.isEmpty() || nope.isEmpty() ||
username.isEmpty() || password.isEmpty() || levelName.equals("- Pilih Level -"))
{
    JOptionPane.showMessageDialog(this, "Semua field harus diisi!", "Error",
JOptionPane.ERROR_MESSAGE);

    return;
}

int idLevel = getIdLevelByName(levelName);

if (idLevel <= 0) { // Jika idLevel tidak valid (misal 0 atau -1 dari
getIdLevelByName)

    JOptionPane.showMessageDialog(this, "Level tidak valid atau tidak
ditemukan di database.", "Error", JOptionPane.ERROR_MESSAGE);

    return;
}

try {
    // Cek apakah username sudah ada

    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); // Untuk kasus EDIT, agar tidak konflik
dengan diri sendiri

    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"); // Sesuai gambar

    load_data();

    bersih();

    // IDotomatis(); // Sudah dipanggil di dalam bersih()

}

} catch (SQLException e) {

    // Cek duplikasi ID_USER jika primary key violation

    if (e.getSQLState().startsWith("23")){ // Kode error untuk integrity
constraint violation

        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_BEEditActionPerformed

    // TODO: Implementasi fungsi Edit

    JOptionPane.showMessageDialog(this, "Fungsi Edit belum
diimplementasikan.");

} //GEN-LAST:event_BEEditActionPerformed

private void BHapusActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_BHapusActionPerformed

    // TODO: Implementasi fungsi Hapus

    JOptionPane.showMessageDialog(this, "Fungsi Hapus belum
diimplementasikan.");

} //GEN-LAST:event_BHapusActionPerformed

/**
 * @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.L
evel.SEVERE, null, ex);

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.logging.L
evel.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
}

```



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
			P	0852298374...	queen	12345

Analisis : Struktur basis data yang terdiri dari tabel `tbl_user` dan `tbl_level` mencerminkan hubungan satu ke banyak (one-to-many), di mana setiap level dalam `tbl_level` dapat dimiliki oleh banyak pengguna di `tbl_user`, namun setiap pengguna hanya memiliki satu level. Tabel `tbl_level` bertindak sebagai referensi atau lookup table untuk mendefinisikan jenis level pengguna (misalnya admin, user biasa, dsb.), sedangkan tabel `tbl_user` menyimpan data pribadi pengguna seperti nama, jenis kelamin, nomor telepon, username, dan password. Penggunaan `ID_LEVEL` sebagai foreign key dalam `tbl_user` menjamin integritas referensial antara kedua tabel. Selain itu, pemilihan tipe data `VARCHAR` untuk sebagian besar kolom memberikan fleksibilitas penyimpanan teks, sementara kolom kunci seperti `ID_USER` dan `ID_LEVEL` ditandai sebagai `NOT NULL` dan `PRIMARY KEY`, yang menegaskan pentingnya nilai unik dan tidak boleh kosong dalam menjaga keunikan data pengguna dan level. Struktur ini cukup umum dan efisien digunakan dalam sistem manajemen pengguna seperti aplikasi web atau sistem informasi internal.

### **BAB III**

#### **POSTTEST**

1. Sempurnakan Program. Ketika di Klik Tombol Simpan akan menampilkan YesNoQuestion penyimpanan Data.
2. Ketika Disimpan Juga, Form Input Data Kembali Kosong seperti kondisi awal.
3. Setelah Disimpan, Data yang dimasukkan akan langsung tampil pada Table.

```
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; // Ditambahkan untuk PreparedStatement

/**
 *
 * @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(); // Memastikan CLevel memiliki item terpilih jika perlu
```

```

        bersih();

        IIdUser.setEditable(false); // ID User tidak boleh diedit manual
    }

    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) {

            JOptionPane.showMessageDialog(this, "Koneksi Database Gagal!");

            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); // Ambil angka
setelah "USER"

            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() {
    // CLevel diisi manual via initComponents.
    // Pastikan ada item yang terpilih jika CLevel tidak kosong.
    if (CLevel.getItemCount() > 0) {
        CLevel.setSelectedIndex(0); // Pilih item pertama (misal "ADMIN")
    }
    // Jika ingin mengisi dari DB:
    // CLevel.removeAllItems();
    // CLevel.addItem("- Pilih Level -");
    // try {
    //     if (koneksi.con == null) return;
    //     st = koneksi.con.createStatement();
    //     String sql = "SELECT level FROM tbl_level ORDER BY level ASC";
    //     rs = st.executeQuery(sql);
    //     while(rs.next()){
    //         CLevel.addItem(rs.getString("level"));
    //     }
    // }
    // catch (SQLException e) {
    //     JOptionPane.showMessageDialog(null, "Error Level: " +
    // e.getMessage());
    //     e.printStackTrace();
    // }
}

```

```

        // }

        // } catch (SQLException e) {

        // JOptionPane.showMessageDialog(null, "Error memuat level:
        "+e.getMessage());

        // }

    }

    private void bersih() {

        // IIdUser.setText(""); // ID User akan di-set oleh IDOtomatis()

        if (CLevel.getItemCount() > 0) {

            CLevel.setSelectedIndex(0); // Kembali ke kondisi awal untuk Level

        }

        INamaUser.setText("");

        Gjk.clearSelection(); // Membersihkan pilihan radio button

        INope.setText("");

        IUser.setText("");

        IPassword.setText("");

        INamaUser.requestFocus();

        IDOtomatis(); // Panggil lagi untuk ID baru

    }

    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();
    }
    // Fallback jika tidak ketemu atau error (sesuaikan jika nama level berbeda di
DB)
    // Ini hanya contoh fallback, idealnya query DB berhasil.
    if (levelName.equalsIgnoreCase("ADMIN")) return 1;
    if (levelName.equalsIgnoreCase("GUDANG")) return 2;
    if (levelName.equalsIgnoreCase("KASIR")) return 3;
    return 0; // Default atau error
}

/**
 * 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"
    }
));
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.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BEditActionPerformed(evt);
    }
});

BHapus.setText("HAPUS");

```

```

BHapus.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BHapusActionPerformed(evt);
    }
});

```

```

    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.LEADING)
                .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()
                    .addComponent(jLabel3)
                    .addComponent(jLabel4)
                    .addComponent(jLabel5)
                    .addComponent(jLabel6)
                    .addComponent(jLabel7)

```

```

        .addComponent(jLabel8)
        .addComponent(jLabel2))
    .addGap(25, 25, 25)
    .addGroup(layout.createParallelGroup.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(BSimpan)
            .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(BEdit)
            .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(BHapus)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup.LEADING, 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(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(
G)
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

        .addGap(20, 20, 20)

        .addComponent(jLabel9)

        .addGap(32, 32, 32)

        .addGroup(layout.createSequentialGroup()
        javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel1)

        .addComponent(jLabel10))

        .addGap(9, 9, 9)

        .addGroup(layout.createSequentialGroup()
        javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

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

        .addGroup(layout.createSequentialGroup()

        .addGroup(layout.createSequentialGroup()
        javax.swing.GroupLayout.Alignment.BASELINE)

        .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.BASELINE)

        .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.BASELINE)

        .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.BASELINE)

        .addComponent(jLabel5)

        .addComponent(JkL)

        .addComponent(JkP))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlace
ment.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.BASELINE)

        .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.Align
ment.BASELINE)

        .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.Align
ment.LEADING)

        .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());

        // Validasi input dasar

        if (namaUser.isEmpty() || jk.isEmpty() || nope.isEmpty() ||
            username.isEmpty() || password.isEmpty() || levelName.equals("- Pilih Level -") ||
            CLevel.getSelectedIndex() == -1) { // Ditambahkan pengecekan jika CLevel
            punya item default

                JOptionPane.showMessageDialog(this, "Semua field harus diisi!", "Error",
                JOptionPane.ERROR_MESSAGE);

                return;
            }
        }
    } //GEN-LAST:event_BSimpanActionPerformed

```



```

    }

    // --- POST TEST 1: YesNoQuestion Penyimpanan Data ---
    int konfirmasi = JOptionPane.showConfirmDialog(this,
        "Apakah Data Sudah Benar? SIMPAN?", // Pesan sesuai gambar
        "Simpan Data", // Judul dialog sesuai gambar
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE); // Icon question mark

    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 {
            // Cek apakah username sudah ada (untuk menghindari duplikasi
username)

            String checkSql = "SELECT COUNT(*) FROM tbl_user WHERE
username = ? AND id_user != ?"; // id_user != ? untuk edit

            PreparedStatement checkPs = koneksi.con.prepareStatement(checkSql);
            checkPs.setString(1, username);

            checkPs.setString(2, idUser); // Meskipun ini insert, tidak apa-apa,
karena id_user baru tidak akan sama dengan yang ada.

            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"); // Sesuai gambar (meskipun ini data user)

        // --- POST TEST 3: Data yang dimasukan akan langsung tampil pada
        Table ---

        load_data();

        // --- POST TEST 2: Form Input Data Kembali Kosong seperti
        kondisi awal ---

        bersih();

        // IDotomatis(); // Sudah dipanggil di dalam bersih()
    }
} catch (SQLException e) {
    if (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();

        }

    } else {

        // User memilih NO, tidak melakukan apa-apa

        // JOptionPane.showMessageDialog(this, "Penyimpanan dibatalkan."); //
Opsional

    }

} //GEN-LAST:event_BSimpanActionPerformed

private void BEditActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_BEEditActionPerformed

    // TODO: Implementasi fungsi Edit

    JOptionPane.showMessageDialog(this, "Fungsi Edit belum
diimplementasikan.");

    } //GEN-LAST:event_BEEditActionPerformed

private void BHapusActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_BHapusActionPerformed

    // TODO: Implementasi fungsi Hapus

    JOptionPane.showMessageDialog(this, "Fungsi Hapus belum
diimplementasikan.");

    } //GEN-LAST:event_BHapusActionPerformed

/**

 * @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.lo
gging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(FUser.class.getName()).log(java.util.lo
gging.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 IldUser;
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;

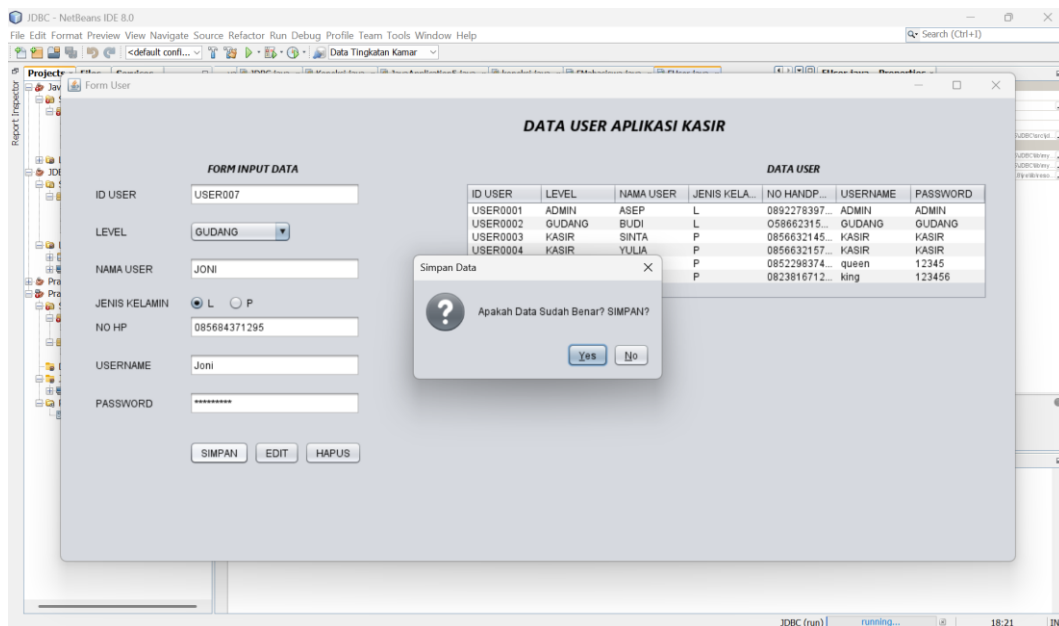
```

```

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 : Rangkaian tugas dalam "POST TEST" ini berfokus pada penyempurnaan fungsionalitas inti dari proses penyimpanan data dalam aplikasi, dengan tujuan utama meningkatkan interaksi pengguna (user experience) dan integritas data. Poin pertama, yaitu menampilkan dialog konfirmasi "YesNoQuestion" sebelum data disimpan, merupakan langkah krusial untuk mencegah kesalahan input atau penyimpanan yang tidak disengaja. Ini memberikan pengguna kesempatan kedua untuk meninjau tindakannya, yang sangat penting dalam aplikasi yang mengelola data. Dengan adanya konfirmasi, pengguna menjadi lebih sadar akan proses yang sedang berlangsung dan dapat membatalkan jika ada keraguan, sehingga mengurangi risiko data yang salah masuk ke sistem.

Secara teknis dan alur kerja, poin kedua dan ketiga melengkapi proses pasca-penyimpanan. Poin kedua, yang mengharuskan form input kembali kosong seperti kondisi awal setelah penyimpanan berhasil, secara signifikan meningkatkan efisiensi untuk entri data berikutnya. Pengguna tidak perlu secara manual menghapus field satu per satu, sehingga alur kerja menjadi lebih cepat dan

mengurangi kebosanan. Poin ketiga, yaitu data yang baru disimpan langsung tampil pada tabel, memberikan umpan balik visual yang instan kepada pengguna bahwa operasi penyimpanan telah berhasil dan data mereka telah tercatat dengan benar dalam sistem. Kombinasi ketiga poin ini menciptakan siklus penyimpanan data yang lengkap, interaktif, dan efisien, mulai dari konfirmasi, eksekusi, hingga visualisasi hasil, yang pada akhirnya meningkatkan kepercayaan pengguna terhadap aplikasi.

## BAB IV

### TUGAS

1. Buatlah Event Handler, untuk kondisi ketika Form Input Data masih kosong, dan button simpan di klik akan memberikan pesan “Data Harus Diisi”!
2. Lengkapi Button Simpan, Edit, Delete dengan Icon yang sesuai.

```
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; // Ditambahkan untuk 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) {
            JOptionPane.showMessageDialog(this, "Koneksi Database
Gagal!");
            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() {
    if (CLevel.getItemCount() > 0) {
        CLevel.setSelectedIndex(0);
    }
    INamaUser.setText("");
    Gjk.clearSelection();
    INope.setText("");
}

```

```

        IUser.setText("");
        IPassword.setText("");
        INamaUser.requestFocus();
        IDOtomatis();
    }

    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();
        }

        if (levelName.equalsIgnoreCase("ADMIN")) return 1;
        if (levelName.equalsIgnoreCase("GUDANG")) return 2;
        if (levelName.equalsIgnoreCase("KASIR")) return 3;
        return 0;
    }

    /**
     * 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"
    }
));
jScrollPane1.setViewportViewView(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.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BEditActionPerformed(evt);
    }
});

```

```

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

```

```

// --- TUGAS NOMOR 2: Lengkapi Button dengan Icon ---
// Pastikan Anda memiliki file ikon di path yang benar (misal:
src/icons/save_icon.png)
// Jika ikon tidak ditemukan, tidak akan ada error, tapi ikon tidak tampil.
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().");

        // Anda bisa menambahkan pesan ini jika ingin tahu jika ikon tidak
termuat.

        // Atau biarkan saja, tombol akan tampil tanpa ikon.

    }

    // --- Akhir Tugas Nomor 2 ---

    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(42, 42, 42)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Al
ignment.LEADING)

                .addGroup(layout.createSequentialGroup()

                    .addGap(135, 135, 135)

                    .addComponent(jLabel1)

                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacem
ent.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

                    .addComponent(jLabel10)

                    .addGap(236, 236, 236))

                .addGroup(layout.createSequentialGroup()

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)

                        .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.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(BSimpan)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPla
cement.RELATED)
            .addComponent(BEdit)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPla
cement.RELATED)
            .addComponent(BHapus)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.Group
Layout.Alignment.LEADING, 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.ComponentPlac
cement.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.ALEA
DING)

        .addGroup(layout.createSequentialGroup())

        .addGap(20, 20, 20)

        .addComponent(jLabel9)

        .addGap(32, 32, 32)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
ignment.BASELINE)

        .addComponent(jLabel1)

        .addComponent(jLabel10))

        .addGap(9, 9, 9)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
ignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.TRAILING)

        .addGroup(layout.createSequentialGroup())

```

```

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .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.BASELINE)

            .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.BASELINE)

            .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.BASELINE)

            .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.A
ignment.BASELINE)

        .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.A
ignment.BASELINE)

        .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.A
ignment.LEADING)

        .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());

    // --- TUGAS NOMOR 1: Event Handler untuk Form Kosong ---

    if (namaUser.isEmpty() || jk.isEmpty() || nope.isEmpty() ||
username.isEmpty() || password.isEmpty() || CLevel.getSelectedIndex() == -1

```

```

/* Bisa juga cek levelName jika punya nilai default seperti "- Pilih Level -" */)
{

    JOptionPane.showMessageDialog(this, "Data Harus Diisi !!",
    "Peringatan", JOptionPane.WARNING_MESSAGE); // Pesan sesuai
    permintaan

    return; // Hentikan proses jika ada data yang kosong

}

// --- Akhir Tugas Nomor 1 ---


// --- POST TEST 1 (dari soal sebelumnya): YesNoQuestion
Penyimpanan 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");

```

```

        load_data(); // POST TEST 3 (dari soal sebelumnya)

```

```

        bersih(); // POST TEST 2 (dari soal sebelumnya)

```

```

        }

```

```

    } catch (SQLException e) {

```

```

        if (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_BEEditActionPerformed

    // TODO: Implementasi fungsi Edit

    JOptionPane.showMessageDialog(this, "Fungsi Edit belum
diimplementasikan.");

} //GEN-LAST:event_BEEditActionPerformed

```

```

private void BHapusActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_BHapusActionPerformed

    // TODO: Implementasi fungsi Hapus

    JOptionPane.showMessageDialog(this, "Fungsi Hapus belum
diimplementasikan.");

} //GEN-LAST:event_BHapusActionPerformed

```

```

/**

 * @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

}

```

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
USER0005	ADMIN	Queen	P	0852298374...	queen	12345
USER0006	KASIR	Kling	P	0823816712...	king	123456
USER0007	GUDANG	JONI	L	0856843712...	Joni	joni12345

Analisis : Dua tugas dalam ini lebih menekankan pada validasi input data dasar dan peningkatan aspek visual antarmuka pengguna (UI). Tugas pertama, yaitu membuat event handler yang memeriksa kondisi form input kosong dan menampilkan pesan "Data Harus Diisi !!" jika tombol simpan diklik, adalah bentuk validasi front-end yang esensial. Tujuannya adalah untuk memastikan bahwa semua field yang wajib diisi benar-benar telah diisi oleh pengguna sebelum data tersebut diproses lebih lanjut atau bahkan sebelum dialog

konfirmasi (dari POST TEST) muncul. Ini membantu menjaga kualitas dan kelengkapan data yang masuk ke database, serta memberikan panduan langsung kepada pengguna mengenai field mana yang belum mereka lengkapi, sehingga mengurangi frustrasi dan mempercepat proses koreksi.

Tugas kedua, melengkapi tombol "Simpan", "Edit", dan "Delete" dengan ikon yang sesuai, bertujuan untuk meningkatkan usability dan estetika aplikasi. Ikon berfungsi sebagai petunjuk visual yang dapat membantu pengguna mengenali fungsi tombol dengan lebih cepat, terutama bagi pengguna yang lebih responsif terhadap elemen grafis daripada teks. Penggunaan ikon yang konsisten dan familiar dapat membuat antarmuka terlihat lebih profesional, modern, dan intuitif. Secara teknis, ini melibatkan penggunaan kelas ImageIcon dan metode setIcon() pada komponen JButton. Pemilihan ikon yang tepat dan penempatannya yang baik akan memperkaya pengalaman pengguna secara keseluruhan, membuat aplikasi tidak hanya fungsional tetapi juga menarik secara visual.

## **BAB V**

### **KESIMPULAN**

Kesimpulan ini merangkum analisis mendalam terhadap serangkaian instruksi pengembangan yang terbagi menjadi dua bagian utama: "PRAKTIKUM". Kedua bagian ini bertujuan untuk meningkatkan fungsionalitas, interaksi pengguna, dan kualitas keseluruhan aplikasi manajemen data pengguna. Masing-masing set instruksi memiliki fokus spesifik yang, ketika diimplementasikan, akan memberikan kontribusi signifikan terhadap pengalaman pengguna dan keandalan sistem.

Analisis terhadap "PRAKTIKUM" mengungkapkan bahwa fokus utamanya adalah pada penyempurnaan alur proses penyimpanan data. Implementasi dialog konfirmasi (Yes/No Question) sebelum penyimpanan merupakan langkah krusial untuk meningkatkan integritas data dan memberikan kontrol lebih kepada pengguna, mencegah kesalahan input yang tidak disengaja. Selanjutnya, pengosongan form input secara otomatis setelah data berhasil disimpan, serta pembaruan tabel data secara real-time, secara signifikan meningkatkan efisiensi kerja pengguna dan memberikan umpan balik visual yang instan. Secara keseluruhan, ketiga poin dalam PRAKTIKUM ini bersinergi untuk menciptakan siklus penyimpanan data yang lebih aman, intuitif, dan efisien bagi pengguna akhir.

Sementara itu, "PRAKTIKUM" lebih menitikberatkan pada aspek validasi input dasar dan peningkatan elemen visual antarmuka pengguna. Tugas pertama, yaitu implementasi event handler untuk memeriksa kekosongan form input dan menampilkan pesan peringatan "Data Harus Diisi !!", adalah bentuk validasi front-end fundamental yang memastikan kelengkapan data sebelum diproses lebih lanjut. Tugas kedua, melengkapi tombol-tombol fungsional (Simpan, Edit, Hapus) dengan ikon yang relevan, tidak hanya mempercantik tampilan aplikasi tetapi juga meningkatkan usability dengan mempermudah pengenalan fungsi tombol secara visual dan intuitif.

Secara kolektif, implementasi seluruh instruksi dari "PRAKTIKUM" akan menghasilkan aplikasi yang tidak hanya lebih fungsional dan handal dalam pengelolaan data, tetapi juga lebih ramah pengguna (user-friendly) dan profesional. Peningkatan ini mencakup aspek integritas data, efisiensi alur kerja, validasi input, serta estetika dan kemudahan penggunaan antarmuka. Hasil akhirnya adalah sebuah aplikasi yang lebih matang dan memberikan pengalaman pengguna yang lebih baik secara menyeluruh.