LAPORAN PEMOGRAMAN

BERORIENTASI OBJECT

PERIODE: SEMESTER GENAP 2024/2025

NAMA :

LULU ANGELITA SUGIARTO

24104410021

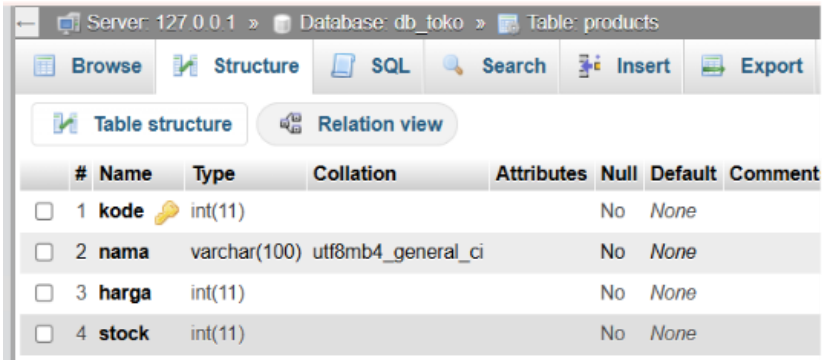
PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

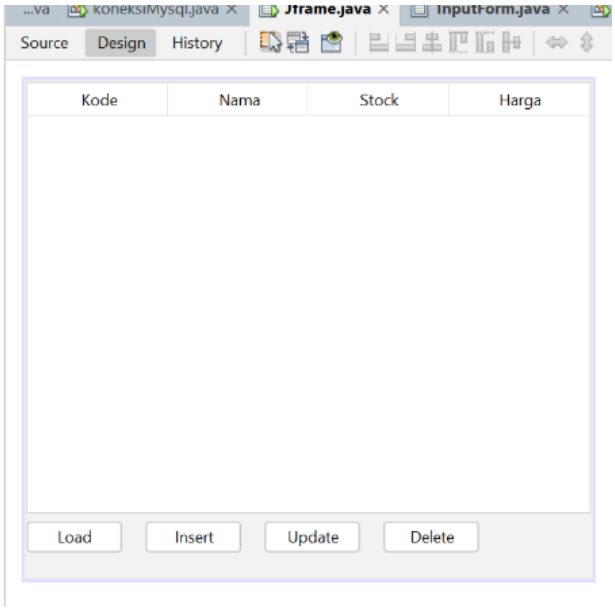
UNIVERSITASISLAM BALITAR

2025

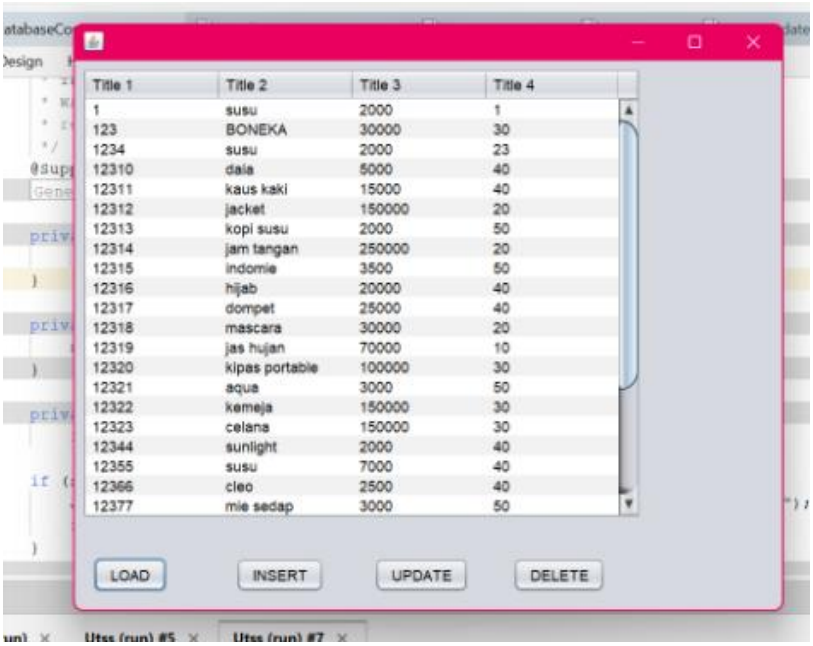
1. Pembuatan Database dengan nama db_toko dan table product



2. Pembuatan JFrame yang dilengkapi dengan 4 JButton dan 1 JTable

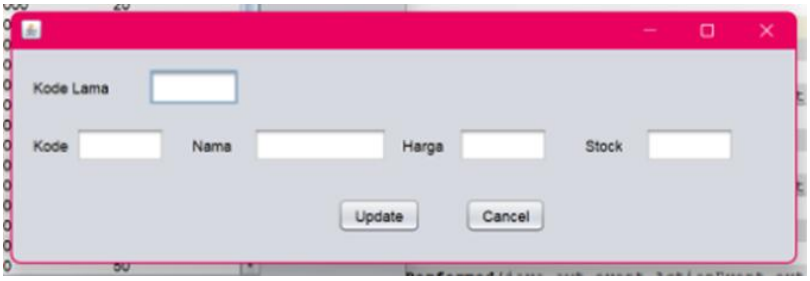


3. Load: seleksi semua data di tabel produk

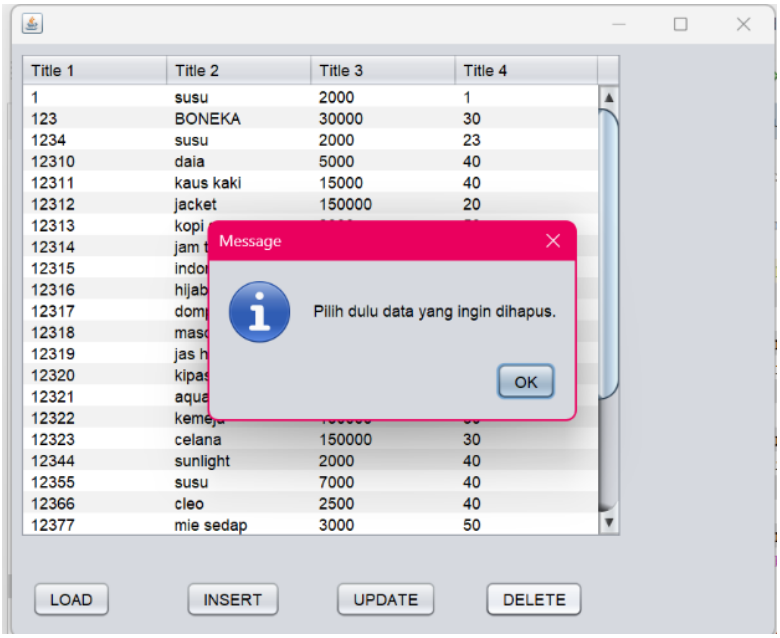


- Insert: memasukkan satu produk baru

- Update: menyeleksi salah satu isi tabel dan melakukan update



- Delete: menyeleksi salah satu isi tabel dan melakukan delete



4. Pada project anda harus menerapkan enkapsulasi, Inheritance, try-catch SQLException, GUI Swing
   a. Enkaspulasi

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class Product extends AbstrackProduk {

  public Product(int kode, String nama, int harga, int stok) {
    super(kode, nama, harga, stok);
  }

  @Override
  public void simpanKeDatabase() {
    try {
      Connection conn = DatabaseConnection.getConnection();
      PreparedStatement stmt = conn.prepareStatement(
        "INSERT INTO product (kode, nama, harga, stock) VALUES (?, ?, ?, ?)"
      );
      stmt.setInt(1, getKode());
      stmt.setString(2, getNama());
      stmt.setInt(3, getHarga());
      stmt.setInt(4, getStok());
      stmt.executeUpdate();
      System.out.println("Produk berhasil disimpan ke database.");
    } catch (SQLException e) {
```

```
            System.err.println("Gagal menyimpan produk: " + e.getMessage());
        }
    }

    Object gettock() {
        throw new UnsupportedOperationException("Not supported yet."); //
Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBod
y
    }

    Object getStock() {
        throw new UnsupportedOperationException("Not supported yet."); //
Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBod
y
    }
}
```

b. Inheritance

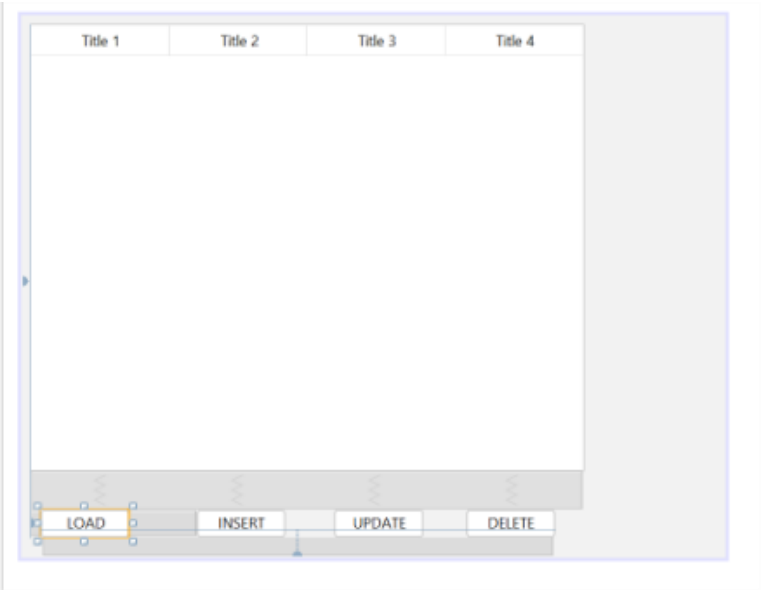
```
public class Product extends AbstrackProduk {

    public Product(int kode, String nama, int harga, int stok) {
        super(kode, nama, harga, stok);
    }
```

c. try-catch SQLException

```
    @Override
    public void simpanKeDatabase() {
        try {
            Connection conn = DatabaseConnection.getConnection();
            PreparedStatement stmt = conn.prepareStatement(
                "INSERT INTO product (kode, nama, harga, stock) VALUES (?, ?, ?, ?)"
            );
            stmt.setInt(1, getKode());
            stmt.setString(2, getNama());
            stmt.setInt(3, getHarga());
            stmt.setInt(4, getStok());
            stmt.executeUpdate();
            System.out.println("Produk berhasil disimpan ke database.");
        } catch (SQLException e) {
            System.err.println("Gagal menyimpan produk: " + e.getMessage());
        }
    }
```

d. GUI Swing



5. Kerjakan codingnya dan tulis laporannya disertai penjelasan coding dan hasil screenshot
outputnya.

A. Kelas kasir

```
package projectpbo;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import java.sql.Statement;
import java.sql.ResultSet;

/*
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to
edit this template
 */

/**
 *
 * @author ASUS
 */
public class Kasir extends javax.swing.JFrame {

    /**
     * Creates new form Kasir
     */
    public Kasir() {
        initComponents();
    }

    /**
     * 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">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        Load = new javax.swing.JButton();
        Insert = new javax.swing.JButton();
        Update = new javax.swing.JButton();
        Delete = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jTable1.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 [] {
                "Kode", "Nama", "Harga", "Stock"
            }
        ));
        jScrollPane1.setViewportView(jTable1);

        Load.setText("Load");
        Load.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                LoadActionPerformed(evt);
            }
        });

        Insert.setText("Insert");
        Insert.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                InsertActionPerformed(evt);
            }
        });

        Update.setText("Update");
        Update.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                UpdateActionPerformed(evt);
            }
        });

        Delete.setText("Delete");
        Delete.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                DeleteActionPerformed(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()
                .addContainerGap()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 664, Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(Load)
                    .addGap(18, 18, 18)
                    .addComponent(Insert)
                    .addGap(18, 18, 18)
                    .addComponent(Update)
                    .addGap(18, 18, 18)
                    .addComponent(Delete)
                    .addGap(0, 0, Short.MAX_VALUE)))
            .addContainerGap())
      );
      layout.setVerticalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(layout.createSequentialGroup()
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 280,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(36, 36, 36)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(Load)
                .addComponent(Insert)
                .addComponent(Update)
                .addComponent(Delete))
            .addGap(0, 10, Short.MAX_VALUE))
      );

      pack();
   }// </editor-fold>

   private void LoadActionPerformed(java.awt.event.ActionEvent evt) {

try (Connection conn = DatabaseConnection.DatabaseConnection(); Statement
stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery("SELECT *
FROM products")) {
      DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
       model.setRowCount(0);
```

```
        while (rs.next()) {
            Vector<Object> row = new Vector<>();
            row.add(rs.getInt("kode"));
            row.add(rs.getString("nama"));
            row.add(rs.getInt("harga"));
            row.add(rs.getInt("stock"));
            model.addRow(row);
        }

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Error saat memuat data: " +
ex.getMessage(), "Database Error", JOptionPane.ERROR_MESSAGE);
        ex.printStackTrace();
    }

    }

    private void InsertActionPerformed(java.awt.event.ActionEvent evt) {
        new FormInput().setVisible(true);
    }

    private void UpdateActionPerformed(java.awt.event.ActionEvent evt) {
    new FormUpdate().setVisible(true);
    }

    private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
          int selectedRow = jTable1.getSelectedRow();

if (selectedRow == -1) {
    JOptionPane.showMessageDialog(this, "Pilih dulu data yang ingin dihapus.");
    return;
}

int confirm = JOptionPane.showConfirmDialog(this, "Yakin ingin menghapus data
ini?", "Konfirmasi Hapus", JOptionPane.YES_NO_OPTION);

if (confirm == JOptionPane.YES_OPTION) {
    try {
       int id = Integer.parseInt(jTable1.getValueAt(selectedRow, 0).toString()); //
Ambil ID dari kolom pertama (index 0)
        Connection conn = DatabaseConnection.DatabaseConnection();
        String sql = "DELETE FROM products WHERE kode = ?";
        PreparedStatement stmt = conn.prepareStatement(sql);
```

```
        stmt.setInt(1, id);
        stmt.executeUpdate();

        // Hapus dari JTable juga
         DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        model.removeRow(selectedRow);

        JOptionPane.showMessageDialog(this, "Data berhasil dihapus!");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Gagal hapus data: " +
e.getMessage());
    }
}

    }

    /**
     * @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(Kasir.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Kasir.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 Kasir().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton Delete;
    private javax.swing.JButton Insert;
    private javax.swing.JButton Load;
    private javax.swing.JButton Update;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    // End of variables declaration
}
```

Analisa :
Kelas Kasir merupakan jendela utama aplikasi kasir berbasis Java Swing yang berfungsi untuk menampilkan, menambah, mengubah, dan menghapus data produk pada database MySQL. Di dalamnya terdapat sebuah tabel (JTable) yang menampilkan daftar produk dengan kolom Kode, Nama, Harga, dan Stock, serta empat tombol utama: Load, Insert, Update, dan Delete. Tombol Load akan mengambil seluruh data produk dari tabel products di database menggunakan query SELECT * FROM products, lalu menampilkannya di tabel dengan memanfaatkan DefaultTableModel. Tombol Insert membuka form input baru (FormInput) agar pengguna dapat menambah data produk, sedangkan tombol

Update membuka form update (FormUpdate) untuk mengedit data produk yang dipilih.

Tombol Delete memungkinkan pengguna menghapus data produk yang sedang dipilih di tabel. Sebelum menghapus, aplikasi akan menampilkan dialog konfirmasi. Jika pengguna setuju, aplikasi akan mengambil nilai kode produk dari baris yang dipilih, lalu menjalankan perintah SQL DELETE FROM products WHERE kode = ? menggunakan PreparedStatement. Setelah data berhasil dihapus dari database, baris tersebut juga dihapus dari tabel pada antarmuka, sehingga tampilan dan data tetap sinkron. Jika terjadi kesalahan saat mengakses database, aplikasi akan menampilkan pesan error melalui dialog.

Selain itu, kelas ini juga mengatur tampilan aplikasi dengan mencoba mengaktifkan tema “Nimbus” jika tersedia, serta memastikan seluruh komponen GUI dijalankan di Event Dispatch Thread untuk menjaga stabilitas aplikasi Swing. Secara keseluruhan, kode ini sudah menerapkan operasi CRUD dasar dengan antarmuka yang sederhana dan interaksi database yang langsung, namun validasi input dan penanganan error dapat dikembangkan lebih lanjut agar aplikasi lebih andal dan ramah pengguna

B. FORM UPDATE

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package projectpbo;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import java.sql.Statement;
import java.sql.ResultSet;

/**
```

```
 *
 * @author ASUS
 */
public class FormUpdate extends javax.swing.JFrame {

    /**
     * Creates new form FormUpdate
     */
    public FormUpdate() {
        initComponents();
    }

    /**
     * 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">
    private void initComponents() {

        jLabel4 = new javax.swing.JLabel();
        Stock = new javax.swing.JTextField();
        INSERT = new javax.swing.JButton();
        CANCEL = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        tfKodeBaru = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        tfNama = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        Harga = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        tfKode = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel4.setText("Stock");

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

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

        CANCEL.setText("CANCEL");

        jLabel2.setText("Kode Baru");

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

        jLabel1.setText("Nama");

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

        jLabel3.setText("Harga");

        jLabel5.setText("Kode Lama");

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

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
```

```
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(INSERT)
            .addGap(18, 18, 18)
            .addComponent(CANCEL)
            .addGap(69, 69, 69))
         .addGroup(layout.createSequentialGroup()
            .addGap(19, 19, 19)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
               .addGroup(layout.createSequentialGroup()
                  .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                  .addComponent(tfKodeBaru,
javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE))
               .addGroup(layout.createSequentialGroup()
                  .addComponent(jLabel5)
                  .addGap(18, 18, 18)
                  .addComponent(tfKode, javax.swing.GroupLayout.PREFERRED_SIZE,
71, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(40, 40, 40)
            .addComponent(jLabel1)
            .addGap(18, 18, 18)
            .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(55, 55, 55)
            .addComponent(jLabel3)
            .addGap(18, 18, 18)
            .addComponent(Harga, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(68, 68, 68)
            .addComponent(jLabel4)
            .addGap(18, 18, 18)
            .addComponent(Stock, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(14, Short.MAX_VALUE))
      );
      layout.setVerticalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                .addGroup(layout.createSequentialGroup()
                    .addGap(63, 63, 63)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                        .addComponent(jLabel1)
                        .addComponent(tfNama,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel3)
                        .addComponent(Harga,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel4)
                        .addComponent(Stock,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                        .addComponent(jLabel5)
                        .addComponent(tfKode,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(26, 26, 26)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(jLabel2)
```

```
                .addComponent(tfKodeBaru,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(INSERT)
                .addComponent(CANCEL))
            .addContainerGap(24, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void StockActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void tfKodeBaruActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void tfNamaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void tfKodeActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void INSERTActionPerformed(java.awt.event.ActionEvent evt) {
    int kodeLama = Integer.parseInt(tfKode.getText());
    int kodeBaru = Integer.parseInt(tfKodeBaru.getText());
    String nama = tfNama.getText();
    int harga = Integer.parseInt(Harga.getText());
    int stok = Integer.parseInt(Stock.getText());

try {
    Connection conn = DatabaseConnection.DatabaseConnection(); // pastikan
class koneksi sudah sesuai
```

```
   String sql = "UPDATE products SET kode = ?, nama = ?, harga = ?, stock = ?
WHERE kode = ?";
   PreparedStatement stmt = conn.prepareStatement(sql);
   stmt.setInt(1, kodeBaru);
   stmt.setString(2, nama);
   stmt.setInt(3, harga);
   stmt.setInt(4, stok);
   stmt.setInt(5, kodeLama);


   int rowsUpdated = stmt.executeUpdate();
   if (rowsUpdated > 0)
   {
      JOptionPane.showMessageDialog(this, "Data berhasil diupdate.");
      this.dispose();
   } else {
      JOptionPane.showMessageDialog(this, "Data gagal diupdate. Kode lama
tidak ditemukan.");
   }

   stmt.close();
   conn.close();
} catch (NumberFormatException e) {
   JOptionPane.showMessageDialog(this, "Harga dan Stock harus berupa
angka.");
} catch (SQLException e) {
   JOptionPane.showMessageDialog(this, "Kesalahan database: " +
e.getMessage());
}
   }

   /**
    * @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(FormUpdate.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(FormUpdate.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(FormUpdate.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(FormUpdate.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
        }
        //</editor-fold>

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

    // Variables declaration - do not modify
    private javax.swing.JButton CANCEL;
    private javax.swing.JTextField Harga;
    private javax.swing.JButton INSERT;
    private javax.swing.JTextField Stock;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
```

```
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JTextField tfKode;
    private javax.swing.JTextField tfKodeBaru;
    private javax.swing.JTextField tfNama;
    // End of variables declaration
}
```

- ANALISA

  Kelas FormUpdate adalah jendela aplikasi berbasis Java Swing yang digunakan untuk mengubah data produk pada tabel products di database MySQL. Form ini menyediakan beberapa field input: Kode Lama (untuk mencari data yang akan diubah), Kode Baru, Nama, Harga, dan Stock, serta dua tombol aksi, yaitu INSERT (yang sebenarnya berfungsi sebagai UPDATE) dan CANCEL. Seluruh komponen diatur menggunakan GroupLayout agar tampilan rapi dan mudah digunakan.

  Saat tombol INSERT ditekan, aplikasi akan mengambil nilai dari setiap field input, lalu melakukan parsing ke tipe data yang sesuai (misalnya, harga dan stok menjadi integer). Kemudian, aplikasi membuat koneksi ke database menggunakan kelas DatabaseConnection, menyiapkan perintah SQL UPDATE products SET kode = ?, nama = ?, harga = ?, stock = ? WHERE kode = ? dengan PreparedStatement, dan mengisi parameter sesuai input pengguna. Jika eksekusi update berhasil (jumlah baris yang terpengaruh lebih dari nol), aplikasi menampilkan pesan sukses dan menutup form. Jika tidak ada data yang diubah (kode lama tidak ditemukan), akan muncul pesan gagal update. Penanganan error juga sudah diterapkan: jika input angka tidak valid, muncul peringatan, dan jika terjadi error database, pesan error ditampilkan ke pengguna.

  Secara keseluruhan, kode ini sudah cukup baik dalam mengelola proses update data produk dengan antarmuka yang jelas dan umpan balik yang informatif. Namun, ada beberapa hal yang bisa ditingkatkan, seperti mengganti label tombol INSERT menjadi UPDATE agar lebih sesuai dengan fungsinya, menambahkan validasi agar semua field wajib diisi, serta memastikan hanya angka positif yang diterima untuk harga dan stok. Selain itu, tombol CANCEL sebaiknya diberi aksi untuk menutup form, sehingga pengalaman pengguna menjadi lebih baik

C. Form input

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to
edit this template
 */
package projectpbo;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import java.sql.Statement;
import java.sql.ResultSet;

/**
 *
 * @author ASUS
 */
public class FormInput extends javax.swing.JFrame {

    /**
     * Creates new form FormInput
     */
    public FormInput() {
        initComponents();
    }

    /**
     * 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">
    private void initComponents() {

        jLabel2 = new javax.swing.JLabel();
        tfKode = new javax.swing.JTextField();
```

```
jLabel1 = new javax.swing.JLabel();
Nama = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
Harga = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
Stock = new javax.swing.JTextField();
INSERT = new javax.swing.JButton();
CANCEL = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel2.setText("Kode");

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

jLabel1.setText("Nama");

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

jLabel3.setText("Harga");

Harga.setText("jTextField1");

jLabel4.setText("Stock");

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

INSERT.setText("INSERT");
INSERT.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      INSERTActionPerformed(evt);
```

```
        }
    });

    CANCEL.setText("CANCEL");

    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(22, 22, 22)
        .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(tfKode, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(49, 49, 49)
        .addComponent(jLabel1)
        .addGap(18, 18, 18)
        .addComponent(Nama, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(55, 55, 55)
        .addComponent(jLabel3)
        .addGap(18, 18, 18)
        .addComponent(Harga, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(68, 68, 68)
        .addComponent(jLabel4)
        .addGap(18, 18, 18)
        .addComponent(Stock, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(49, Short.MAX_VALUE))
      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(INSERT)
        .addGap(18, 18, 18)
        .addComponent(CANCEL)
        .addGap(69, 69, 69))
    );
    layout.setVerticalGroup(
```

```
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(63, 63, 63)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(jLabel2)
                .addComponent(tfKode, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel1)
                .addComponent(Nama, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel3)
                .addComponent(Harga, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel4)
                .addComponent(Stock, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(34, 34, 34)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(INSERT)
                .addComponent(CANCEL))
            .addContainerGap(31, Short.MAX_VALUE))
      );

      pack();
   }// </editor-fold>

   private void NamaActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
   }

   private void StockActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
   }

   private void tfKodeActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // TODO add your handling code here:
    }

    private void INSERTActionPerformed(java.awt.event.ActionEvent evt) {
        try (Connection conn = DatabaseConnection.DatabaseConnection()) {
            String sql = "INSERT INTO products (kode, nama, harga, stock) VALUES (?, ?,
?, ?)";
            try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
                int kode = Integer.parseInt(tfKode.getText());
                String nama = Nama.getText();
                int harga = Integer.parseInt(Harga.getText());
                int stock = Integer.parseInt(Stock.getText());

                pstmt.setInt(1, kode);
                pstmt.setString(2, nama);
                pstmt.setInt(3, harga);
                pstmt.setInt(4, stock);

                int rowsAffected = pstmt.executeUpdate();
                if (rowsAffected > 0) {
                    JOptionPane.showMessageDialog(this, "Data berhasil ditambahkan!",
"Sukses", JOptionPane.INFORMATION_MESSAGE);
                    this.dispose();
                } else {
                    JOptionPane.showMessageDialog(this, "Gagal menambahkan data.",
"Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        } catch (SQLException | NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage(), "Input
Error", JOptionPane.ERROR_MESSAGE);
            ex.printStackTrace();
        }
    }

    /**
     * @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(FormInput.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(FormInput.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(FormInput.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(FormInput.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        }
        //</editor-fold>

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

    // Variables declaration - do not modify
    private javax.swing.JButton CANCEL;
    private javax.swing.JTextField Harga;
```

```
    private javax.swing.JButton INSERT;
    private javax.swing.JTextField Nama;
    private javax.swing.JTextField Stock;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JTextField tfKode;
    // End of variables declaration
}
```

- ANALISA

  Kelas **FormInput** adalah jendela aplikasi Java Swing yang digunakan untuk menambah data produk baru ke tabel `products` pada database MySQL. Antarmuka form ini terdiri dari field input untuk kode, nama, harga, dan stok produk, serta dua tombol aksi: **INSERT** untuk menyimpan data dan **CANCEL** untuk membatalkan proses. Setiap field input dihubungkan dengan label yang jelas agar pengguna mudah memahami data yang harus diisi. Layout form diatur menggunakan `GroupLayout` sehingga semua elemen tertata rapi secara horizontal dan vertikal.

  Fungsi utama dari form ini terletak pada aksi tombol **INSERT**. Ketika tombol tersebut ditekan, aplikasi akan mengambil nilai dari keempat field input, melakukan parsing ke tipe data yang sesuai (misalnya, kode, harga, dan stok menjadi integer), lalu menyiapkan query SQL `INSERT INTO products (kode, nama, harga, stock) VALUES (?, ?, ?, ?)` menggunakan `PreparedStatement`. Nilai-nilai input dimasukkan ke dalam parameter query secara aman. Setelah query dieksekusi, jika data berhasil ditambahkan ke database, form akan menampilkan pesan sukses dan otomatis menutup; jika gagal, akan muncul pesan error. Penanganan error juga sudah diterapkan: jika terjadi kesalahan dalam parsing angka atau saat koneksi ke database, aplikasi akan menampilkan pesan yang sesuai kepada pengguna dan mencetak detail error untuk debugging.

  Selain itu, kode ini juga menangani pengaturan tampilan aplikasi dengan mencoba mengaktifkan tema “Nimbus” jika tersedia, serta memastikan form ditampilkan di Event Dispatch Thread agar antarmuka tetap stabil. Secara umum, implementasi ini sudah cukup baik untuk kebutuhan input data produk secara langsung, namun masih dapat ditingkatkan dengan menambahkan validasi agar semua field wajib diisi, memastikan nilai angka tidak negatif, dan memberikan aksi pada tombol **CANCEL** agar form bisa ditutup tanpa menyimpan data[.

D. Data base

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this
template
 */
package projectpbo;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 *
 * @author ASUS
 */
public class DatabaseConnection {

   private static final String URL = "jdbc:mysql://localhost:3306/db_toko";
   private static final String USER = "root"; // Ganti dengan username MySQL
Anda
   private static final String PASSWORD = ""; // Ganti dengan password MySQL
Anda
   private static Connection connection = null;

   // Metode untuk mendapatkan koneksi
   public static Connection DatabaseConnection() throws SQLException {
   if (connection == null || connection.isClosed()) {
      try {
         Class.forName("com.mysql.cj.jdbc.Driver");
         connection = DriverManager.getConnection(URL, USER, PASSWORD);
         System.out.println("Koneksi ke database berhasil!");
      } catch (ClassNotFoundException e) {
         System.err.println("JDBC Driver tidak ditemukan.");
         throw new SQLException("Driver tidak ditemukan", e);
      }
   }
   return connection;
}

   // Metode untuk menutup koneksi
```

```
    public static void closeConnection() {
        if (connection != null) {
            try {
                connection.close();
                System.out.println("Koneksi database ditutup.");
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    static Connection getConnection() throws SQLException,
ClassNotFoundException {
    Class.forName("com.mysql.cj.jdbc.Driver");
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/db_toko",
"root", "");
}


}
```

- ANALISA

  Kode DatabaseConnection ini berfungsi sebagai utilitas untuk mengelola koneksi ke database MySQL bernama db_toko di localhost. Kelas ini menggunakan pola singleton pada metode DatabaseConnection(), sehingga hanya akan ada satu objek koneksi aktif yang digunakan bersama di seluruh aplikasi—hal ini membantu mencegah pemborosan resource akibat terlalu banyak koneksi terbuka. Pada saat koneksi pertama kali dibutuhkan atau jika koneksi sebelumnya sudah ditutup, driver JDBC MySQL akan dimuat, lalu koneksi baru dibuat menggunakan username dan password yang telah ditentukan.

  Metode closeConnection() disediakan untuk menutup koneksi jika sudah tidak diperlukan, sehingga resource database dapat dilepaskan dengan baik. Selain itu, terdapat metode getConnection() yang selalu membuat koneksi baru setiap kali dipanggil, berbeda dengan metode singleton sebelumnya. Hal ini bisa menimbulkan inkonsistensi dan potensi kebocoran resource jika tidak digunakan secara hati-hati.

  Secara umum, kode ini sudah cukup baik untuk aplikasi skala kecil hingga menengah karena sederhana dan mudah dipahami. Namun, penggunaan dua metode berbeda untuk mendapatkan koneksi bisa membingungkan dan berisiko jika tidak dikelola dengan baik. Untuk aplikasi yang lebih besar atau multi-threaded, sebaiknya pola singleton ini dilengkapi dengan mekanisme thread-safe

atau connection pooling, dan konfigurasi database dipindahkan ke file eksternal agar lebih aman dan fleksibel.

E. Products

```
/*
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
*/
package UTS;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class Product extends AbstrackProduk {

    public Product(int kode, String nama, int harga, int stok) {
        super(kode, nama, harga, stok);
    }

    @Override
    public void simpanKeDatabase() {
        try {
            Connection conn = DatabaseConnection.getConnection();
            PreparedStatement stmt = conn.prepareStatement(
                "INSERT INTO product (kode, nama, harga, stock) VALUES (?, ?, ?, ?)"
            );
            stmt.setInt(1, getKode());
            stmt.setString(2, getNama());
            stmt.setInt(3, getHarga());
            stmt.setInt(4, getStok());
            stmt.executeUpdate();
            System.out.println("Produk berhasil disimpan ke database.");
        } catch (SQLException e) {
            System.err.println("Gagal menyimpan produk: " + e.getMessage());
        }
    }

    Object gettock() {
```

```
        throw new UnsupportedOperationException("Not supported yet."); //
Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }

    Object getStock() {
        throw new UnsupportedOperationException("Not supported yet."); //
Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
}
```

- Analisa
  DatabaseConnection.getConnection() dipanggil untuk memperoleh koneksi sebelum menyimpan data produk melalui PreparedStatement ke tabel product, di mana nilai kode, nama, harga, dan stok diisi menggunakan getter warisan dari AbstrackProduk. Implementasi simpanKeDatabase() sudah benar memanfaatkan executeUpdate() untuk mengeksekusi perintah INSERT dan menangani SQLException untuk memberi umpan balik kesalahan. Namun, terdapat dua metode gettock() dan getStock() yang belum diimplementasikan dan menyebabkan UnsupportedOperationException, sehingga memecah alur jika dipanggil; seharusnya hanya perlu satu metode getter untuk stok produk dengan penamaan konsisten. Selain itu, perlu dipastikan tabel di database memiliki kolom stock (bukan stok) agar sesuai dengan query SQL, serta menambahkan validasi data sebelum penyimpanan untuk mencegah nilai negatif atau duplikasi.

- Output