

LAPORAN TUGAS AKHIR PEMROGRAMAN BERBASIS TEKS GRAFIS DAN MULTIMEDIA

Sistem Informasi Toko Handphone



Disusun oleh:

Kelompok 1

Johannes Alexander Putra (XI-B/40)

....

.....

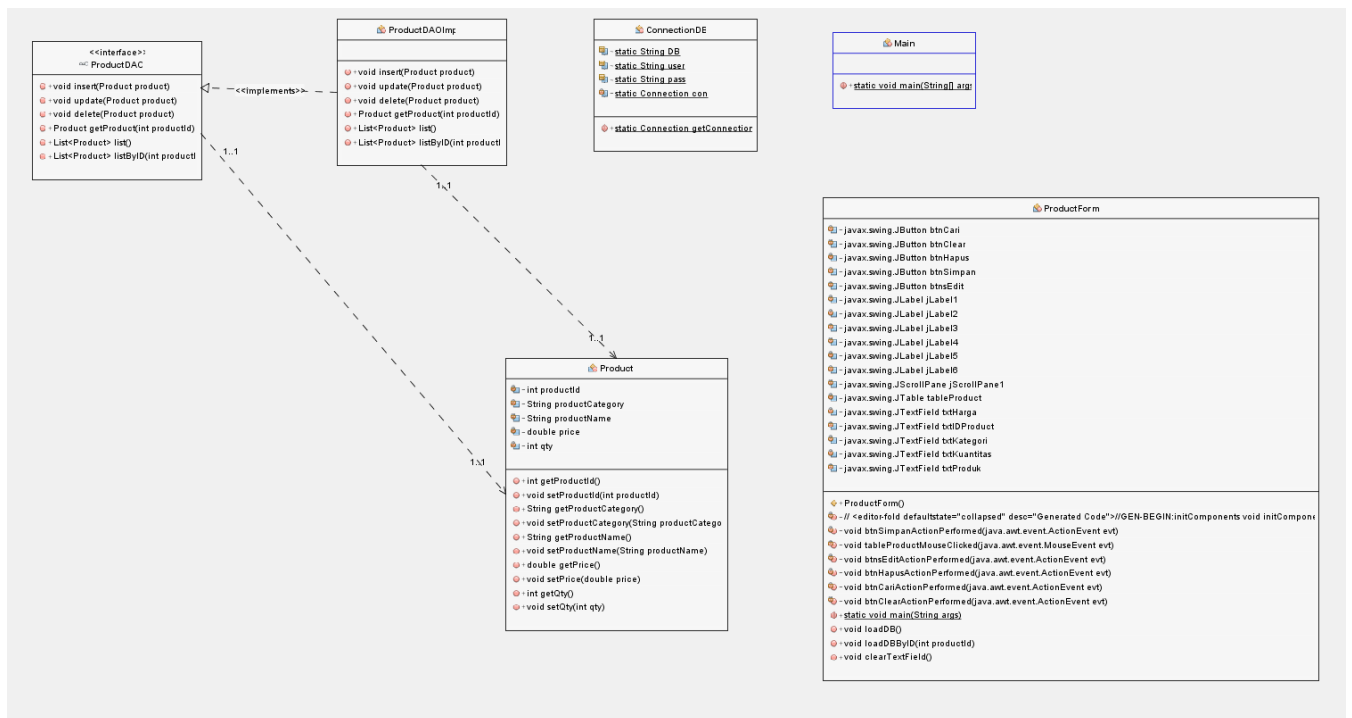
....

REKAYASA PERANGKAT LUNAK

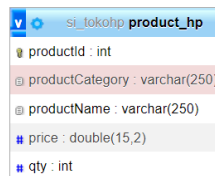
2023

1. IMPLEMENTASI

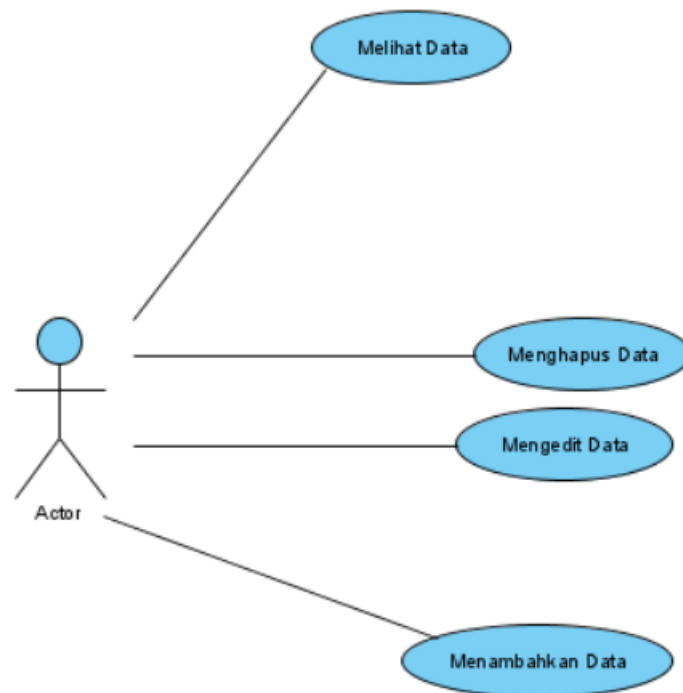
a. Class Diagram



b. ERD Basis Data



c. Use Case Diagram



d. Source Code

- **Main.java**

```
package com.johanes.tokohapevisual;

import com.johanes.tokohapevisual.view.ProductForm;

/**
 *
 * @author acer
 */
public class Main {
    public static void main(String[] args) {
        ProductForm product = new ProductForm();
        product.setVisible(true);
    }
}
```

- **ConnectionDB.java**

```

package com.johanes.tokohapevisual.DB;

/**
 *
 * @author Johannes Alexander Putra
 */

import java.sql.*;

public class ConnectionDB {

    // Menyiapkan paramter JDBC untuk koneksi ke database
    static String DB="jdbc:mysql://localhost/si_tokohp"; // si_tokohp database
    static String user="root"; // user database
    static String pass=""; // password database
    private static Connection con;

    public static Connection getConnection() throws SQLException {
        if(con==null){
            try {
                DriverManager.registerDriver(new com.mysql.cj.jdbc.Driver());
                con = (Connection) DriverManager.getConnection(DB,user,pass);
            } catch (Exception e) {
                System.out.println("Database connection is failed");
            }
        }

        return con;
    }
}

```

- ProductDAO.java

```

package com.johanes.tokohapevisual.controller;

import com.johanes.tokohapevisual.model.Product;
import java.util.List;

/**
 *
 * @author erlanga
 */

public interface ProductDAO {
    public void insert(Product product);
    public void update(Product product);
    public void delete(Product product);
    public Product getProduct(int productId);
    public List<Product> list();
    public List<Product> listByID(int productId);
}

```

- ProductDAOImp.java

```

package com.johanes.tokohapevisual.controller;

import com.johanes.tokohapevisual.DB.ConnectionDB;
import com.johanes.tokohapevisual.model.Product;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;

/**
 *
 * @author erlanga
 */
public class ProductDAOImp implements ProductDAO {

    @Override
    public void insert(Product product) {
        try {
            Connection conDB = ConnectionDB.getConnection();
            String sql = "INSERT INTO product_hp (productCategory,productName,price,qty) VALUES (?,?,?,?)";
            PreparedStatement ps = conDB.prepareStatement(sql);
            ps.setString(1, product.getProductCategory());
            ps.setString(2, product.getProductCategory());
            ps.setString(2, product.getProductName());
            ps.setDouble(3, product.getPrice());
            ps.setInt(4, product.getQty());
            ps.executeUpdate();
            JOptionPane.showMessageDialog(null, "Data has been saved");
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "Cannot insert into table");
        }
    }

    @Override
    public void update(Product product) {
        try {
            Connection conDB = ConnectionDB.getConnection();
            String sql = "UPDATE product_hp SET productCategory=?, productName=?, price=?, qty=? WHERE productId=?";
            PreparedStatement ps = conDB.prepareStatement(sql);
            ps.setString(1, product.getProductCategory());
            ps.setString(2, product.getProductName());
            ps.setDouble(3, product.getPrice());
            ps.setInt(4, product.getQty());
            ps.setInt(5, product.getProductId());
            ps.executeUpdate();
            JOptionPane.showMessageDialog(null, "Data has been update");
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "Cannot update into table");
        }
    }

    @Override
    public void delete(Product product) {
        try {
            Connection conDB = ConnectionDB.getConnection();
            String sql = "DELETE FROM product_hp WHERE productId=?";
            PreparedStatement ps = conDB.prepareStatement(sql);
            ps.setInt(1, product.getProductId());
            ps.executeUpdate();
            JOptionPane.showMessageDialog(null, "Data has been delete");
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "Cannot update into table");
        }
    }

    @Override
    public Product getProduct(int productId) {
        Product product = new Product();
        try {
            Connection conDB = ConnectionDB.getConnection();
            String sql = "SELECT * FROM product_hp WHERE productId=?";
            PreparedStatement ps = conDB.prepareStatement(sql);
            ps.setInt(1, productId);
            ResultSet rs = ps.executeQuery();
            if(rs.next()){
                product.setProductId(rs.getInt("productId"));
                product.setProductCategory(rs.getString("productCategory"));
                product.setProductName(rs.getString("productName"));
                product.setPrice(rs.getInt("price"));
                product.setQty(rs.getInt("qty"));
            }
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "Cannot get data from table by using productId");
        }
        return product;
    }

    @Override
    public List<Product> list() {
        List<Product> listProduct = new ArrayList<Product>();
        try{
            Connection conDB = ConnectionDB.getConnection();
            String sql = "SELECT * FROM product_hp";
            PreparedStatement ps = conDB.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();

            while(rs.next()){
                Product product = new Product();
                product.setProductId(rs.getInt("productId"));
                product.setProductCategory(rs.getString("productCategory"));
                product.setProductName(rs.getString("productName"));
                product.setPrice(rs.getInt("price"));
                product.setQty(rs.getInt("qty"));
                listProduct.add(product);
            }
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "Cannot get data from table");
        }
        return listProduct;
    }

    @Override
    public List<Product> listByID(int productId) {
        List<Product> listProduct = new ArrayList<Product>();
        try{
            Connection conDB = ConnectionDB.getConnection();
            String sql = "SELECT * FROM product_hp WHERE productId=?";
            PreparedStatement ps = conDB.prepareStatement(sql);
            ps.setInt(1, productId);
            ResultSet rs = ps.executeQuery();

            while(rs.next()){
                Product product = new Product();
                product.setProductId(rs.getInt("productId"));
                product.setProductCategory(rs.getString("productCategory"));
                product.setProductName(rs.getString("productName"));
                product.setPrice(rs.getInt("price"));
                product.setQty(rs.getInt("qty"));
                listProduct.add(product);
            }
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "Cannot get data from table");
        }
        return listProduct;
    }
}

```

- Product.java

```
package com.johanes.tokohapevisual.model;

/**
 *
 * @author erlanga
 */
public class Product {
    private int productId;
    private String productCategory;
    private String productName;
    private double price;
    private int qty;

    public int getProductId() {
        return productId;
    }

    public void setProductId(int productId) {
        this.productId = productId;
    }

    public String getProductCategory() {
        return productCategory;
    }

    public void setProductCategory(String productCategory) {
        this.productCategory = productCategory;
    }

    public String getProductName() {
        return productName;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public int getQty() {
        return qty;
    }

    public void setQty(int qty) {
        this.qty = qty;
    }
}
```

- ProductForm.java

```

package com.johanes.tokohapevisual.view;

import com.johanes.tokohapevisual.controller.ProductDAOImp;
import com.johanes.tokohapevisual.model.Product;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author erlanga
 */
public class ProductForm extends javax.swing.JFrame {

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

    /**
     * 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() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        txtKategori = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        txtProduk = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        txtHarga = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        txtKuantitas = new javax.swing.JTextField();
        btnSimpan = new javax.swing.JButton();
        btnsEdit = new javax.swing.JButton();
        btnHapus = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        tableProduct = new javax.swing.JTable();
        jLabel6 = new javax.swing.JLabel();
        txtIDProduct = new javax.swing.JTextField();
        btnCari = new javax.swing.JButton();
        btnClear = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Data Product");

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
        jLabel1.setText("Data Produk");

        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel2.setText("Kategori Produk");

        txtKategori.setToolTipText("");

        jLabel3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel3.setText("Nama Produk");

        jLabel4.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel4.setText("Harga");

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel5.setText("Kuantitas");

        btnSimpan.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        btnSimpan.setText("Simpan");
        btnSimpan.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnSimpanActionPerformed(evt);
            }
        });

        btnsEdit.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        btnsEdit.setText("Edit");
        btnsEdit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnsEditActionPerformed(evt);
            }
        });
    }
}

```

```

btnHapus.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnHapus.setText("hapus");
btnHapus.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnHapusActionPerformed(evt);
    }
});

tableProduct.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null}
    },
    new String [] {
        "ID Produk", "Kategori Produk", "Nama Produk", "Harga (Rp.)", "Kuantitas"
    }
) {
    Class[] types = new Class [5] {
        java.lang.Object.class, java.lang.Object.class, java.lang.Object.class,
        java.lang.Double.class, java.lang.Integer.class
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }
});
tableProduct.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tableProductMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tableProduct);

jLabel6.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel6.setText("ID Produk");

txtIDProduct.setEditable(false);
txtIDProduct.setToolTipText("");

btnCaril.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnCaril.setText("Carit");
btnCaril.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCarilActionPerformed(evt);
    }
});

btnClear.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnClear.setText("Clear");
btnClear.setToolTipText("");
btnClear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnClearActionPerformed(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()
            .addComponent(btnSimpan)
            .addGap(10, 10, 10)
            .addComponent(btnEdit)
            .addPreferredGap(Short.MAX_VALUE)
            .addComponent(txtKategori)
            .addComponent(txtProduk)
            .addComponent(txtKategori)
            .addComponent(txtKuantitas, javax.swing.GroupLayout.PREFERRED_SIZE, 197,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtIDProduct))
        .addGroup(layout.createSequentialGroup()
            .addComponent(btnClear)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jLabel1)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel2)
                .addGap(375, 375, 375)))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 866,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel3)
                .addGap(42, 42, 42)))));

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addComponent(jLabel2))
            .addGap(23, 23, 23)
            .addComponent(jLabel3)
            .addGap(PreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 37,
                Short.MAX_VALUE))
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(txtIDProduct, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel6))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(txtKategori, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel2))
            .addGap(21, 21, 21)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(txtProduk, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel3))
            .addGap(20, 20, 20)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(txtKategori, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel4))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(txtKuantitas, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel5))
            .addGap(32, 32, 32)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel6)
                .addComponent(btnSimpan)
                .addComponent(btnEdit)
                .addComponent(btnHapus)
                .addComponent(btnCaril)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 178,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(btnClear))
            .addGap(30, 30, 30))
        .addComponent(pack())
        .setLocationRelativeTo(null);
//editor-folds

```



```

// fungsi tombol Simpan
private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        Product product = new Product();
        String kategori = txtKategori.getText();
        String produk = txtProduk.getText();
        double harga = Double.parseDouble(txtHarga.getText());
        int kuantitas = Integer.parseInt(txtKuantitas.getText());

        product.setProductCategory(kategori);
        product.setProductName(produk);
        product.setPrice(harga);
        product.setQty(kuantitas);

        ProductDAOImp dao = new ProductDAOImp();
        dao.insert(product);
        loadDB();
        clearTextField();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

private void tableProductMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try {
        int row = tableProduct.getSelectedRow();
        String dataKlik = (tableProduct.getModel().getValueAt(row, 0)).toString();
        int productId = Integer.parseInt(dataKlik);

        ProductDAOImp dao = new ProductDAOImp();
        Product productDAO = dao.getProduct(productId);

        txtIDProduct.setText(String.valueOf(productDAO.getProductID()));
        txtKategori.setText(productDAO.getProductCategory());
        txtProduk.setText(productDAO.getProductName());
        txtHarga.setText(String.format("%.0f", productDAO.getPrice()));
        txtKuantitas.setText(String.valueOf(productDAO.getQty()));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

private void btnsEditActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        Product product = new Product();
        int productId = Integer.parseInt(txtIDProduct.getText());
        String kategori = txtKategori.getText();
        String produk = txtProduk.getText();
        double harga = Double.parseDouble(txtHarga.getText());
        int kuantitas = Integer.parseInt(txtKuantitas.getText());
        product.setProductCategory(kategori);
        product.setProductName(produk);
        product.setPrice(harga);
        product.setQty(kuantitas);
        product.setProductId(productId);

        ProductDAOImp dao = new ProductDAOImp();
        dao.update(product);
        loadDB();
        clearTextField();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

private void btnhapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        int productId = Integer.parseInt(txtIDProduct.getText());
        String produk = txtProduk.getText();

        int pesan = JOptionPane.showOptionDialog(this, "Anda yakin akan menghapus produk " + produk +
            "?", "Konfirmasi Hapus Produk", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, null, null,
            null);

        if (pesan == JOptionPane.YES_OPTION) {
            Product product = new Product();
            product.setProductId(productId);
            ProductDAOImp dao = new ProductDAOImp();
            dao.delete(product);
            loadDB();
            clearTextField();
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

private void btnCariActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        int productId = Integer.parseInt(JOptionPane.showInputDialog("Masukan ID Produk"));
        ProductDAOImp dao = new ProductDAOImp();
        Product productDAO = dao.getProduct(productId);
        loadDBByID(productId);
        txtIDProduct.setText(String.valueOf(productDAO.getProductID()));
        txtKategori.setText(productDAO.getProductCategory());
        txtProduk.setText(productDAO.getProductName());
        txtHarga.setText(String.format("%.0f", productDAO.getPrice()));
        txtKuantitas.setText(String.valueOf(productDAO.getQty()));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

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

/**
 * @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(ProductForm.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(ProductForm.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(ProductForm.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(ProductForm.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
    }
    /*editor-fold*/
    /*editor-fold*/
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ProductForm().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnCari;
private javax.swing.JButton btnClear;
private javax.swing.JButton btnhapus;
private javax.swing.JButton btnSimpan;
private javax.swing.JButton btnsEdit;
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.JLabel jLabel6;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tableProduct;
private javax.swing.JTextField txtHarga;
private javax.swing.JTextField txtIDProduct;
private javax.swing.JTextField txtKategori;
private javax.swing.JTextField txtKuantitas;
private javax.swing.JTextField txtProduk;
// End of variables declaration

```

Data Produk

ID Produk
Kategori Produk
Nama Produk
Harga
Kuantitas

ID Produk	Kategori Produk	Nama Produk	Harga (Rp.)	Kuantitas

e. Implementasi

- **View Product**

Data Product
— □ ×

Data Produk

ID Produk
Kategori Produk
Nama Produk
Harga
Kuantitas

ID Produk	Kategori Produk	Nama Produk	Harga (Rp.)	Kuantitas
1	Smartphone	Xiaomi redmi note 5	2,000,000	30

- **Add Product**

Data Product

Data Produk

ID Produk

Kategori Produk

Nama Produk

Harga

Kuantitas

Simpan

Edit

Hapus

Cari

Clear

ID Produk	Kategori Produk	Nama Produk	Harga (Rp.)	Kuantias
1	Smartphone	Xiaomi redmi note 5	2,000,000	30
4	Tablet Android	Redmi Pad 5	3,000,000	10

- Edit Product

Data Product

Data Produk

ID Produk

4

Kategori Produk

Tablet Android

Nama Produk

Redmi Pad 6

Harga

3000000

Kuantitas

10

Simpan

Clear

Message

Data has been update

OK

ID Produk	Kategori Produk	Nama Produk	Harga (Rp.)	Kuantias
1	Smartphone	Xiaomi redmi note 5	2,000,000	30
4	Tablet Android	Redmi Pad 5	3,000,000	10

- Delete Product

Data Product

Data Produk

ID Produk: 4

Kategori Produk: Tablet Android

Nama Produk: Redmi Pad 6

Harga: 3000000

Kuantitas: 10

Simpan Edit Hapus Cari Clear

ID Produk	Kategori Produk	Nama Produk	Harga (Rp.)	Kuantitas
1	Smartphone		2,000,000	30
4	Tablet Android		3,000,000	10

Konfirmasi Hapus Produk
Anda yakin akan menghapus produk Redmi Pad 6?
Yes No

- Cari Product

Data Product

Data Produk

ID Produk: 1

Kategori Produk: Smartphone

Nama Produk: Xiaomi redmi note 5

Harga: 2000000

Kuantitas: 30

Simpan Edit Hapus Cari Clear

ID Produk	Kategori Produk	Nama Produk	Harga (Rp.)	Kuantitas
1	Smartphone		2,000,000	30

Input
Masukan ID Produk
1
OK Cancel