

Tutorial membuat koneksi database dan Master Form Produk Pemrograman Visual I (Java menggunakan Netbeans)

Hasil Output Form Produk :

Id Produk	Id Kategori	Nama Produk	Unit Cost	Id_Supplier
B001	1	Penggaris 30 ...	2500	SP001
B002	2	Pulpen Hitam	3000	SP001
B003	1	Penggaris 15 ...	2000	SP002
B004	5	Kertas A4	35000	SP001
B005	5	Kertas Legal	37000	SP001
B006	5	Kertas Kuarto	36000	SP002

1. Buatlah database dengan nama : *oso_db_NIM.sql*
Kemudian buat Tabel : *produk*
2. **Sruktur tabel**

field	type	null	ket
Id_produk	Char(10)	No	Primary key*
Id_kategori	Int (11)	Yes	
Nama_produk	Varchar(30)	Yes	
Unitcost	Double	Yes	
Id_supplier	Char(10)	yes	

Isi tabel

id_produk	id_kategori	nama_produk	unitcost	id_supplier
B001	1	Penggaris 30 cm	2500	SP001
B002	2	Pulpen Hitam	3000	SP001
B003	1	Penggaris 15 cm	2000	SP002
B004	5	Kertas A4	35000	SP001
B005	5	Kertas Legal	37000	SP001
B006	5	Kertas Kuarto	36000	SP002
B007	2	Pulpen Biru	3000	SP002

3. Buat Project baru pada Netbeans
 - a. Project : *osoJava1_NIM*
 - b. Buat class (java class) : *koneksi.java*
 - c. Buat form (JFrameForm) : *produk.java*

4. Lalu buka file *koneksi.java* yang sudah dibuat sebelumnya
 - a. tambahkan 2 baris koding import dibawah ini untuk memanggil class dalam package Java yang akan digunakan untuk membuat koneksi database

```
package oso;
import java.sql.*;
import javax.swing.JOptionPane;
```

- b. Kemudian tambahkan 3 variabel berikut di dalam class **koneksi**

```
public class koneksi {
    Connection conn;
    Statement st;
    ResultSet rs;
```

- c. Berikutnya adalah buat sebuah method dengan type Connection dengan nama setKoneksi berikut

```
public Connection setKoneksi()
{
    try{
        Class.forName("com.mysql.jdbc.Driver");
        conn=DriverManager.getConnection("jdbc:mysql://localhost/oso_db","root","");
        st=conn.createStatement();
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,"Koneksi Gagal : " +e);
    }
    return conn;
}
}
```

File *koneksi.java* telah selesai dan telah siap untuk mengkoneksikan dengan database yang akan digunakan pada form *produk.java*

5. Buka file *produk.java* yang sudah dibuat sebelumnya

Desain form seperti pada gambar berikut :

Keterangan beberapa komponen Variable Name :

jButton1	: bt_keluar	jTextField1	:tid_produk
jButton2	: bt_tambah	jTextField2	:tid_supplier
jButton3	: bt_edit	jTextField3	:tid_kategori
jButton4	: bt_update	jTextField4	:tnm_produk
jButton5	: bt_batal	jTextField5	:tunit_cost
jButton6	: bt_hapus	jTextField6	:tcari
jTable1	: tbl_produk		

- a. tambahkan 7 baris koding import dibawah ini untuk memanggil class dalam package Java yang akan digunakan untuk membuat koneksi database

```
package oso;
import java.awt.event.KeyEvent;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import sun.misc.Cleaner;
```

- b. kemudian sertakan terlebih dahulu deklarasi beberapa objek dan variable dibawah ini

```
public class Produk extends javax.swing.JFrame {
    koneksi kon=new koneksi();
    private Object [][] dataproduk=null;
    private String[] label={"Id Produk","Id Kategori","Nama Produk","Unit Cost","Id_Supplier"};
```

- c. lalu tambahkan 4 method :

isiNamaKategori(), *isiNamaKategori2()*, *isiNamaSupplier()*, *isiNamaSupplier2()* dan *isiProduk()* yang berfungsi untuk mengolah data kategori dan nama supplier dalam tabel produk

Coding Method isiNamaKategori()

```
void isiNamaKategori() {
    try{
        String sql="Select *From kategori";
        kon.rs=kon.st.executeQuery(sql);
        while (kon.rs.next()){
            cbkategori.addItem(kon.rs.getString("nama_kategori"));
        }
    }catch(SQLException e){
        System.out.println("Koneksi Gagal"+ e.toString());
    }
}
```

Coding Method isiNamaKategori2() :

```
void isiNamaKategori2() {
    try{
        kon.setKoneksi();
        String sql="Select *From kategori where id_kategori='"+tid_kategori.getText()+"'";
        kon.rs=kon.st.executeQuery(sql);
        if (kon.rs.next()){
            cbkategori.setSelectedItem(kon.rs.getString("nama_kategori"));
        }
    }catch(SQLException e){
        System.out.println("Koneksi Gagal"+ e.toString());
    }
}
```

Coding Method isiNamaSupplier() :

```
void isiNamaSupplier() {
    try{
        String sql="Select *From supplier ";
        kon.rs=kon.st.executeQuery(sql);
        while (kon.rs.next()){
            cbsupplier.addItem(kon.rs.getString("nama_supplier"));
        }
    }catch(SQLException e){
        System.out.println("Koneksi Gagal"+ e.toString());
    }
}
```

Coding Method isiNamaSupplier2() :

```
void isiNamaSupplier2() {
    try{
        String sql="Select *From supplier where id_supplier='"+tid_supplier.getText()+"'";
        kon.rs=kon.st.executeQuery(sql);
        if(kon.rs.next()){
            cbsupplier.setSelectedItem(kon.rs.getString("nama_supplier"));
        }
    }catch(SQLException e){
        System.out.println("Koneksi Gagal"+ e.toString());
    }
}
```

Coding Method isiProduk() :

```
private String idProduk()
{
    String no=null;
    try{
        kon.setKoneksi();
        String sql = "Select right(id_produk,3)+1 from produk ";
        ResultSet rs = kon.st.executeQuery(sql);
        if (rs.next()){
            rs.last();
            no = rs.getString(1);
            while (no.length()<3){
                no="00"+no;
                no="B"+no;
                tid_produk.setText(no);
            }
        }else{
            no="B001";
            tid_produk.setText(no);
        }
    }catch (Exception e){
    }return no;
}
```

- d. lalu tambahkan 2 method :

BacaTabelProduk() dan **BacaTabelProduk2()** yang berfungsi untuk membaca dan mengolah isi jTable1

Coding Method BacaTabelProduk() :

```

private void BacaTabelProduk() {
    try{
        String sql="Select *From produk order by id_produk";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        dataproduk=new Object[baris][kolom];
        int x=0;
        kon.rs.beforeFirst();
        while(kon.rs.next()){
            dataproduk[x][0]=kon.rs.getString("id_produk");
            dataproduk[x][1]=kon.rs.getString("id_kategori");
            dataproduk[x][2]=kon.rs.getString("nama_produk");
            dataproduk[x][3]=kon.rs.getString("unitcost");
            dataproduk[x][4]=kon.rs.getString("id_supplier");
            x++;
        }
        tbl_produk.setModel(new DefaultTableModel(dataproduk,label));
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Coding Method BacaTabelProduk2() :

```

private void BacaTabelProduk2() {
    try{
        String sql="select *from produk where nama_produk like '%" +tcari.getText()+"% " ";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        dataproduk=new Object[baris][kolom];
        int x=0;
        kon.rs.beforeFirst();
        while(kon.rs.next()){
            dataproduk[x][0]=kon.rs.getString("id_produk");
            dataproduk[x][1]=kon.rs.getString("id_kategori");
            dataproduk[x][2]=kon.rs.getString("nama_produk");
            dataproduk[x][3]=kon.rs.getString("unitcost");
            dataproduk[x][4]=kon.rs.getString("id_supplier");
            x++;
        }
        tbl_produk.setModel(new DefaultTableModel(dataproduk,label));
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Berikutnya akan dibuat method *setTable()* yang akan di gunakan untuk menampilkan isi table di komponen jTable1.

```
private void setTable() {
int row=tbl_produk.getSelectedRow();
tid_produk.setText((String)tbl_produk.getValueAt(row,0));
tid_kategori.setText((String)tbl_produk.getValueAt(row,1));
tnm_produk.setText((String)tbl_produk.getValueAt(row,2));
tunit_cost.setText((String)tbl_produk.getValueAt(row,3));
tid_supplier.setText((String)tbl_produk.getValueAt(row,4));
}
```

Lalu selanjutnya tambahkan 3 method yaitu **bersihField()**, **aktif()** dan **nonaktif()** yang berfungsi untuk membantu jalannya form saat komponen-komponen didalamnya dalam keadaan tertentu (kosong kembali, aktif dan non aktif)

```
private void BersihField() {
    tunit_cost.setText("");
    tid_produk.setText("");
    tnm_produk.setText("");
    tnm_produk.setText("");
    tcari.setText("");
}
```

Coding method *aktif()* :

```
private void aktif() {
    tid_produk.setEnabled(true);
    tnm_produk.setEnabled(true);
    tunit_cost.setEnabled(true);
    cbkategori.setEnabled(true);
    cbsupplier.setEnabled(true);
}
```

Coding method *nonAktif()* :

```
private void nonaktif() {
    tid_produk.setEnabled(false);
    tnm_produk.setEnabled(false);
    tunit_cost.setEnabled(false);
    cbkategori.setEnabled(false);
    cbsupplier.setEnabled(false);
    bt_edit.setEnabled(false);
    bt_update.setEnabled(false);
    bt_hapus.setEnabled(false);
    bt_simpan.setEnabled(false);
}
```

- e. Kemudian langkah selanjutnya adalah tambahkan 3 method untuk menjalankan fungsi tombol *Simpan*, *Edit* dan *Hapus*

Coding method *SimpanData ()* :

```
private void SimpanData(){
    try{
        String sql="insert into produk
values('"+tid_produk.getText()+"','"+tid_kategori.getText()+"','"+tnm_produk.getText()+"','"+tunit_cost
.getText()+"','"+tid_supplier.getText()+"')";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil disimpan");
        BersihField();
        BacaTabelProduk();
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null,e);
    }
}
```

Coding Method *EditData()* :

```
private void EditData(){
    try{
        String sql="Update produk set
id_produk='"+tid_produk.getText()+"',id_kategori='"+tid_kategori.getText()+"',id_supplier='"+tid_suppl
ier.getText()+"',nama_produk='"+tnm_produk.getText()+"',unitcost='"+tunit_cost.getText()+"' where
id_produk='"+tid_produk.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil diupdate");
        BersihField();
        BacaTabelProduk();
        kon.st.close();
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null,e);
    }
}
```

Coding method *HapusData()* :

```
private void HapusData(){
    try{
        String sql="Delete from produk where id_produk='"+tid_produk.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil dihapus");
        BersihField();
        BacaTabelProduk();
        kon.st.close();
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
}
```

6. Berikutnya kita akan menambahkan beberapa baris coding pada komponen yang ada pada form produk antara lain :

a. **formWindowActivated**

perintah ini diberikan untuk membuat komponen yang ada di form produk dalam kondisi kosong dan non aktif pada saat form pertama kali dijalankan


```
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    BersihField();
    nonaktif();
}
```

b. cbkategoriActionPerformed

perintah dibawah ini digunakan untuk memanggil isi field **nama_kategori** pada combobox **cbkategoriActionPerformed**

```
private void cbkategoriActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        kon.setKoneksi();
        String sql="Select *From kategori where nama_kategori='"+cbkategori.getSelectedItem()+"'";
        kon.rs=kon.st.executeQuery(sql);
        if (kon.rs.next()){
            tid_kategori.setText(kon.rs.getString("id_kategori"));
        }
    }catch(SQLException e){
        System.out.println("Koneksi Gagal"+ e.toString());
    }
}
```

c. cbsupplierActionPerformed

perintah dibawah ini digunakan untuk memanggil isi field **nama_supplier** pada combobox **cbsupplierActionPerformed**

```
private void cbsupplierActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        kon.setKoneksi();
        String sql="Select *From supplier where nama_supplier='"+cbsupplier.getSelectedItem()+"'";
        kon.rs=kon.st.executeQuery(sql);
        if (kon.rs.next()){
            tid_supplier.setText(kon.rs.getString("id_supplier"));
        }
    }catch(SQLException e){
        System.out.println("Koneksi Gagal"+ e.toString());
    }
}
```

d. tnm_produkActionPerformed

perintah dibawah ini digunakan untuk memanggil isi field **nmbarang** pada JTextField1 **tnm_produk**, serta menyeleksi produk yang belum atau telah tersimpan sebelumnya pada table produk

```
private void tnm_produkActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        String sql = "select *from barang where nmbarang=" + tnm_produk.getText() + " ";
        kon.rs = kon.st.executeQuery(sql);
        if (kon.rs.next()) {
            cbkategori.setSelectedItem("=PILIH=");
            JOptionPane.showMessageDialog(this, "Barang sudah ada...", "Informasi", JOptionPane.INFORMATION_MESSAGE);
            bt_tambah.setEnabled(true);
        } else {
            cbsupplier.setEnabled(true);
            cbsupplier.requestFocus();
        }
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

e. tunit_costActionPerformed

perintah dibawah ini difungsikan untuk memberikan focus pada komponen jTextField1 **tunit_cost**

```
private void tunit_costActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    bt_simpan.requestFocus();
}
```

f. tbl_produkMouseClicked

```
private void tbl_produkMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    setTable();
    is>NamaKategori2();
    is>NamaSupplier2();
    bt_hapus.setEnabled(true);
    bt_edit.setEnabled(true);
    bt_tambah.setEnabled(false);
}
```

g. bt_tambahActionPerformed

```
private void bt_tambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    idProduk();
    aktif();
    bt_batal.setEnabled(true);
    bt_tambah.setEnabled(false);
    bt_simpan.setEnabled(true);
}
```

h. bt_simpanActionPerformed

```
private void bt_simpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (tid_produk.getText().isEmpty() || tnm_produk.getText().isEmpty() || tunit_cost.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Lengkapi Data", "Konfirmasi", JOptionPane.INFORMATION_MESSAGE);
        bt_tambah.setEnabled(true);
    } else {
        bt_tambah.setEnabled(true);
        bt_keluar.setEnabled(true);
        SimpanData();
        cbkategori.setSelectedItem("=PILIH=");
        cbsupplier.setSelectedItem("=PILIH=");
        try {
            kon.st.close();
        } catch (SQLException ex) {
            Logger.getLogger(Produk.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

i. bt_simpanKeyPressed

```
private void bt_simpanKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        if (tid_produk.getText().isEmpty() || tnm_produk.getText().isEmpty() || tunit_cost.getText().isEmpty()) {
            JOptionPane.showMessageDialog(this, "Lengkapi Data", "Konfirmasi", JOptionPane.INFORMATION_MESSAGE);
            bt_tambah.setEnabled(true);
        } else {
            bt_tambah.setEnabled(true);
            bt_keluar.setEnabled(true);
            SimpanData();
            cbkategori.setSelectedItem("=PILIH=");
        }
    }
}
```

j. bt_editActionPerformed

```
private void bt_editActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    aktif();
    tid_produk.setEnabled(false);
    bt_edit.setEnabled(false);
    bt_update.setEnabled(true);
    bt_batal.setEnabled(true);
    bt_hapus.setEnabled(false);
    bt_tambah.setEnabled(false);
}
```

k. bt_updateActionPerformed

```
private void bt_updateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    bt_update.setEnabled(false);
    bt_tambah.setEnabled(true);
    EditData();
    cbkategori.setSelectedItem("=PILIH=");
    cbsupplier.setSelectedItem("=PILIH=");
}
```

l. bt_batalActionPerformed

```
private void bt_batalActionPerformed(java.awt.event.ActionEvent evt) {
    nonaktif();
    BersihField();
    bt_tambah.setEnabled(true);
    cbkategori.setSelectedItem("=PILIH=");
    cbsupplier.setSelectedItem("=PILIH=");
    try {
        kon.st.close();
    } catch (SQLException ex) {
        Logger.getLogger(Produk.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

m. bt_hapusActionPerformed

```
private void bt_hapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (JOptionPane.showConfirmDialog(this, "yakin mau dihapus?", "konfirmasi", JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
        HapusData();
        bt_tambah.setEnabled(true);
        nonaktif();
        BersihField();
        cbkategori.setSelectedItem("=PILIH=");
        cbsupplier.setSelectedItem("=PILIH=");
    } else {
        JOptionPane.showMessageDialog(this, "Data Batal Dihapus", "Konfirmasi", JOptionPane.INFORMATION_MESSAGE);
        bt_tambah.setEnabled(true);
        nonaktif();
        BersihField();
        cbkategori.setSelectedItem("=PILIH=");
        cbsupplier.setSelectedItem("=PILIH=");
    }
    return;
}
formWindowActivated(null);
}
```

n. bt_keluarActionPerformed

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

o. tcariActionPerformed

```
private void tcariActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    kon.setKoneksi();  
    BacaTabelProduk2();  
}
```

p. tcariKeyTyped

```
private void tcariKeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    kon.setKoneksi();  
    BacaTabelProduk2();  
}
```

Simpan file dan jalankan form Run – Run File