

# Modul Koneksi Basis Data MySQL ke Java



Pengampu : Seffy K.N, S.Kom

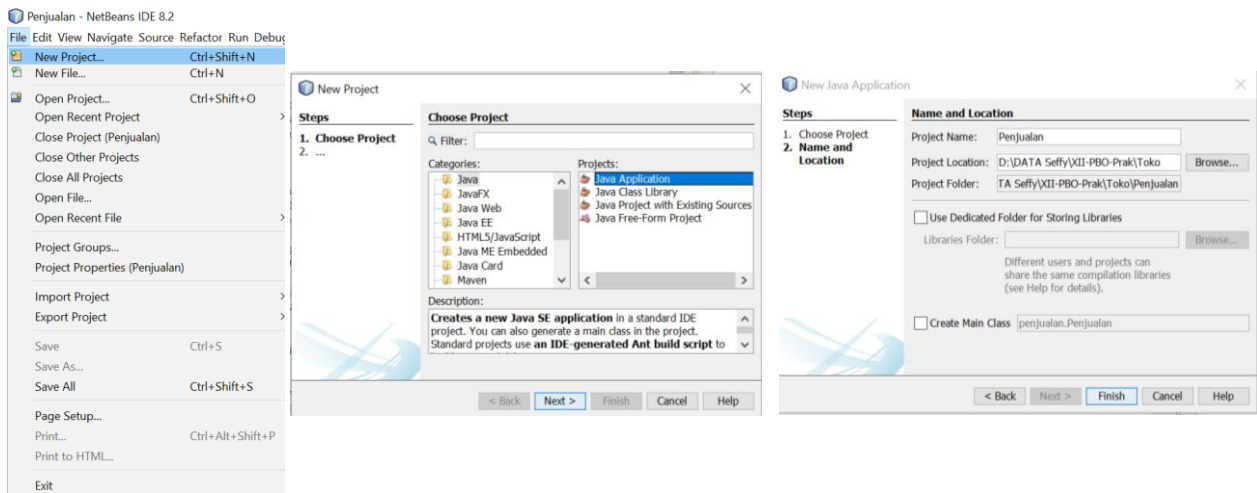
Mata Pelajaran Pemrograman Berorientasi Objek

SMK Negeri 02  
Jl. Dr.Cipto 121A Semarang  
024-8455757  
024-8455757  
10/2/2023

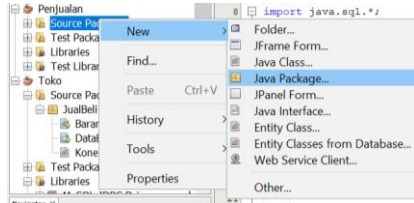
# MEMBUAT KONEKSI BASIS DATA DENGAN MERANCANG GRAPHIC USER INTERFACE

Langkah-langkahnya sebagai berikut :

1. Membuat Project Baru dengan nama Penjualan dengan cara klik File – New Project – pilih java application – klik Next - Project Name (Penjualan) – Finish



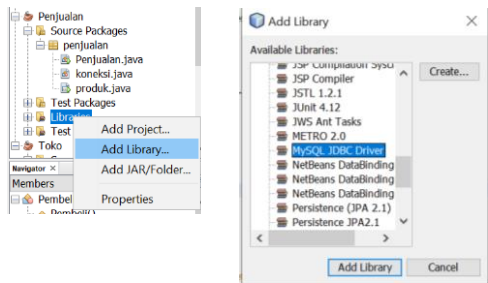
2. Kemudian membuat package baru dengan nama package penjualan dengan cara klik kanan Source Packages – New – Java Package



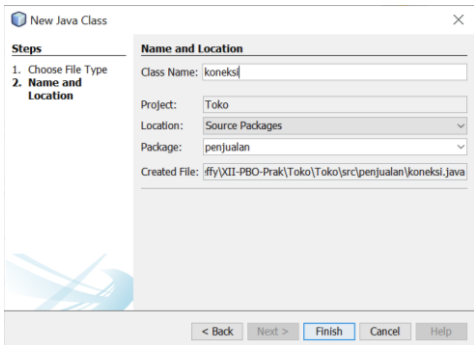
3. Membuat database baru dengan nama penjualan dan nama table produk. Dengan struktur table produk sebagai berikut :

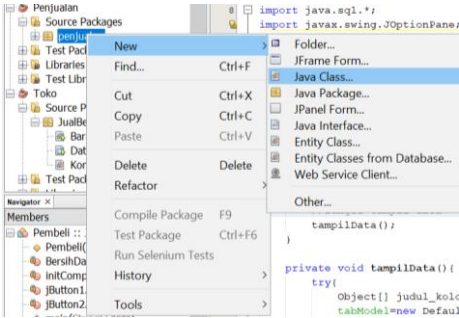
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	IDProduk	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	KodeProduk	varchar(32)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
3	NamaProduk	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
4	Satuan	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
5	Harga	int(11)			Yes	NULL			Change Drop More
6	Stok	int(11)			Yes	NULL			Change Drop More
7	CreateDate	datetime			Yes	NULL			Change Drop More

4. Untuk membuat koneksi ke database mysql, sediakan komponen mysql di Libraries dengan cara klik kanan Libraries – Add Library – Mysql JDBC klik Add Library



5. Membuat Class baru untuk koneksi database dengan nama class koneksi dengan cara klik kanan package penjualan – New – Java Class – New Class (koneksi) - Finish

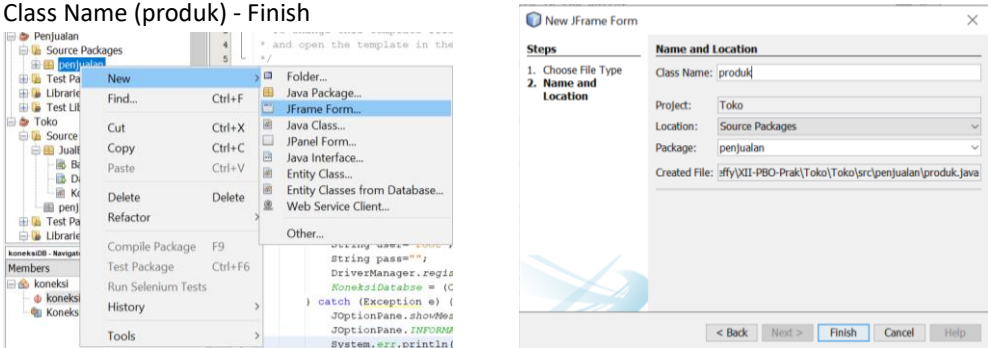




6. Sebelum membuat koding koneksi, tuliskan script guna membaca sql dan JOptionPane package penjualan;
- ```
import java.sql.*; // membaca komponen paket yang dibutuhkan untuk koneksi database
import javax.swing.JOptionPane; //membaca JFrame JOptionPane
```

7. Tuliskan script seperti dibawah ini di dalam public class koneksi () {
- ```
//deklarasi variabel connection database
private static Connection KoneksiDatabase;
//fungsi/method connection
public static Connection koneksiDB() throws SQLException {
    try {
        String DB ="jdbc:mysql://localhost/penjualan";
        String user="root";
        String pass="";
        DriverManager.registerDriver(new com.mysql.jdbc.Driver());
        KoneksiDatabase = (Connection) DriverManager.getConnection(DB,user,pass);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,"Tidak ada koneksi","Error",
        JOptionPane.INFORMATION_MESSAGE);
        System.err.println(e.getMessage());
        System.exit(0);
    }
    return KoneksiDatabase;
}
```

8. Membuat JFrame Form lagi guna mendesain GUI, klik kanan pada package penjualan – New – JFrame Form – Class Name (produk) - Finish



9. Desain JFrame Produk seperti gambar dibawah ini

Data Produk				
Kode Produk	<input type="text" value="JTextField1"/>	Title 1	Title 2	Title 3
Nama	<input type="text" value="JTextField2"/>			
Satuan	<input type="text" value="Pcs"/>			
Harga	<input type="text" value="JTextField3"/>			
Stok	<input type="text" value="JTextField4"/>			
<input type="button" value="AddNew"/>				
<input type="button" value="Save"/>		<input type="button" value="Update"/>	<input type="button" value="Delete"/>	<input type="button" value="Cancel"/>

Dengan Ketentuan sebagai berikut :

No	Komponen GUI	Text/Model	Change Variabel Name
----	--------------	------------	----------------------

1	JLabel 1	Data Produk	JLabelDataProduk
2	JLabel2	Kode Produk	JLabelKodeProduk
3	JLabel3	Nama	JLabelNama
4	JLabel4	Satuan	JLabelSatuan
5	JLabel5	Harga	JLabelHarga
6	JLabel6	Stok	JLabelStok
7	TextField1	TextField1	TextKodeProduk
8	TextField2	TextField2	TextNama
9	TextField3	TextField3	TextHarga
10	TextField4	TextField4	TextStok
11	ComboBox1	Satuan, Pcs, Kg	ComboBoxSatuan
12	Table1	TableModel	TabelProduk
13	Button1	AddNew	ButtonAdd
14	Button2	Save	ButtonSave
15	Button3	Update	ButtonUpdate
16	Button4	Delete	ButtonDelete
17	Button5	Cancel	ButtonCancel

10. Tuliskan source kode produk guna membaca komponen yang dibutuhkan dibawah package penjualan  
package penjualan;  
import java.sql.\*; //membaca sql guna membaca koneksi antar basis data  
import javax.swing.JOptionPane; //membaca komponen JOptionPane  
import javax.swing.table.DefaultTableModel; //membaca table  
import java.util.Scanner; // membaca suatu inputan  
import java.awt.\*; //membaca gui
11. Tuliskan source kode guna mendeklarasikan table dan recorset(data) di bagian public class product extends  
javax.swing.JFrame{  
//mendeklarasikan table dan recorset  
DefaultTableModel tabModel;  
ResultSet RsProduk=null;
12. Membuat method baru tampilTengah guna menampilkan GUI agar ditampilkan di tengah-tengah windows  
//method baru tampilTengah  
void tampilanTengah(){  
Dimension screenSize=Toolkit.getDefaultToolkit().getScreenSize();  
Dimension frameSize=this.getSize();  
if(frameSize.height > screenSize.height){  
frameSize.height=screenSize.height;  
}  
if(frameSize.width > screenSize.width){  
frameSize.width=screenSize.width;  
}  
this.setLocation((screenSize.width - frameSize.width) / 2,  
(screenSize.height = screenSize.height) / 20);  
}
13. Panggil method tampilTengah ()di bagian public produk ( ) { setelah initComponents ( );  
//memanggil fungsi tampilTengah  
tampilTengah ( );
14. Membuat fungsi tampilData guna menampilkan fields yang akan ditampilkan ke table  
//fungsi untuk menampilkan data produk ke dalam tabel  
private void tampilData(){  
try{  
Object[] judul\_kolom = {"Kode Barang", "Nama Barang", "Satuan", "Harga", "Stok", "Create Date"};  
tabModel=new DefaultTableModel(null,judul\_kolom);  
TabelProduk.setModel(tabModel);  
  
Connection conn=(Connection)koneksi.koneksiDB();  
Statement stt=conn.createStatement();  
tabModel.getDataVector().removeAllElements();

```

RsProduk=stt.executeQuery("SELECT * from produk ");
while(RsProduk.next()){
    Object[] data={
        RsProduk.getString("KodeProduk"),
        RsProduk.getString("NamaProduk"),
        RsProduk.getString("Satuan"),
        RsProduk.getString("Harga"),
        RsProduk.getString("Stok"),
        RsProduk.getString("CreateDate")
    };
    tabModel.addRow(data);
}
} catch (Exception ex) {
System.err.println(ex.getMessage());
}

// disable button guna mengutamakan fungsi utama AddNew (hanya button AddNew yang bisa diklik)
jButtonSave.setEnabled(false);
jButtonUpdate.setEnabled(false);
jButtonDelete.setEnabled(false);
jButtonCancel.setEnabled(false);

```

15. Panggil fungsi tampilData di bagian public produk ( ) { initComponents ( ) ; dibawah fungsi tampilTengah( )

```

//memanggil fungsi tampilTengah
tampilTengah( );
//memanggil fungsi tampilData
tampilData( );

```

16. Membuat method baru SetEditOff() (mengutamakan fungsi AddNew)

```

//guna mematikan semua komponen
private void SetEditOff(){
    jTextKodeProduk.setEnabled(false);
    jTextNama.setEnabled(false);
    jComboBoxSatuan.setEnabled(false);
    jTextHarga.setEnabled(false);
    jTextStok.setEnabled(false);
}

```

17. Membuat method baru SetEditOn()

```

//guna menghidupkan kembali semua komponen agar bisa digunakan
private void seteditOn(){
    jTextKodeProduk.setEnabled(true);
    jTextNama.setEnabled(true);
    jComboBoxSatuan.setEnabled(true);
    jTextHarga.setEnabled(true);
    jTextStok.setEnabled(true);
}

```

18. Panggil fungsi tampilData di bagian public produk ( ) { initComponents ( ) ; dibawah fungsi tampilData( )

```

//memanggil fungsi tampilTengah
tampilTengah( );
//memanggil fungsi tampilData
tampilData( );
//mematikan semua komponen kecuali button Addnew
SetEditOff();

```

19. Membuat fungsi membersihkan form

```

//membersihkan form
private void BersihData(){

```

```

jTextKodeProduk.setText("");
jTextNama.setText("");
jComboBoxSatuan.setSelectedIndex(0);
jTextHarga.setText("");
jTextStok.setText("");
jTextKodeProduk.requestFocus(); //memfokuskan kursor ke kode produk
}

```

20. Panggil fungsi BersihData ke button AddNew, dengan cara double klik pada button AddNew

```

BersihData();
//menghidupkan Button Save
jButtonSave.SetEnable(True);
//mematikan/nonaktifkan Button Update dan Delete
jButtonUpdate.SetEnable(False);
jButtonDelete.SetEnable(False);
//menyalakan semua tools dalam gui
seteditOn();

```

21. Membuat script program button Save

```

private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {
String KodeProduk=jTextKodeProduk.getText();
String NamaProduk=jTextNama.getText();
String Satuan=jComboBoxSatuan.getSelectedItem().toString();
String Harga=jTextHarga.getText();
String Stok=jTextStok.getText();

java.util.Date tanggal = new java.util.Date();
java.text.SimpleDateFormat TanggalFormat= new java.text.SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
String CreateDate=TanggalFormat.format(tanggal);

if (KodeProduk.isEmpty() ) {
JOptionPane.showMessageDialog(null,"Kode produk tidak boleh kosong");
jTextKodeProduk.requestFocus();
}else if (NamaProduk.isEmpty()) {
JOptionPane.showMessageDialog(null,"Nama produk tidak boleh kosong");
jTextNama.requestFocus();
}else if (Satuan.isEmpty()) {
JOptionPane.showMessageDialog(null,"Satuan produk tidak boleh kosong");
jComboBoxSatuan.requestFocus();
}else if (Harga.isEmpty()) {
JOptionPane.showMessageDialog(null,"Harga produk tidak boleh kosong");
jTextHarga.requestFocus();
}else if (Stok.isEmpty()) {
JOptionPane.showMessageDialog(null,"Stok produk tidak boleh kosong");
jTextStok.requestFocus();
}
else if (!(new Scanner(Harga).hasNextInt())) {
JOptionPane.showMessageDialog(null,"Harga produk harus angka");
jTextHarga.requestFocus();
}else if (!(new Scanner(Stok).hasNextInt())) {
JOptionPane.showMessageDialog(null,"Stok produk harus angka");
jTextStok.requestFocus();
}else{
try{
Connection conn=(Connection)koneksi.koneksiDB();
Statement stt=conn.createStatement();
stt.executeUpdate("insert into produk(KodeProduk,NamaProduk,Satuan,Harga,Stok,CreateDate)" +

"VALUES('"+KodeProduk+"','"+NamaProduk+"','"+Satuan+"','"+Harga+"','"+Stok+"','"+CreateDate+"')");
BersihData();
}
}
}

```

```
tampilData();
SetEditOff();
JOptionPane.showMessageDialog(this,"Data
disimpan","Success",JOptionPane.INFORMATION_MESSAGE);
} catch(SQLException e){
JOptionPane.showMessageDialog(this,"Simpan
gagal\n"+e.getMessage(),"Error",JOptionPane.ERROR_MESSAGE);
}
}
}
```

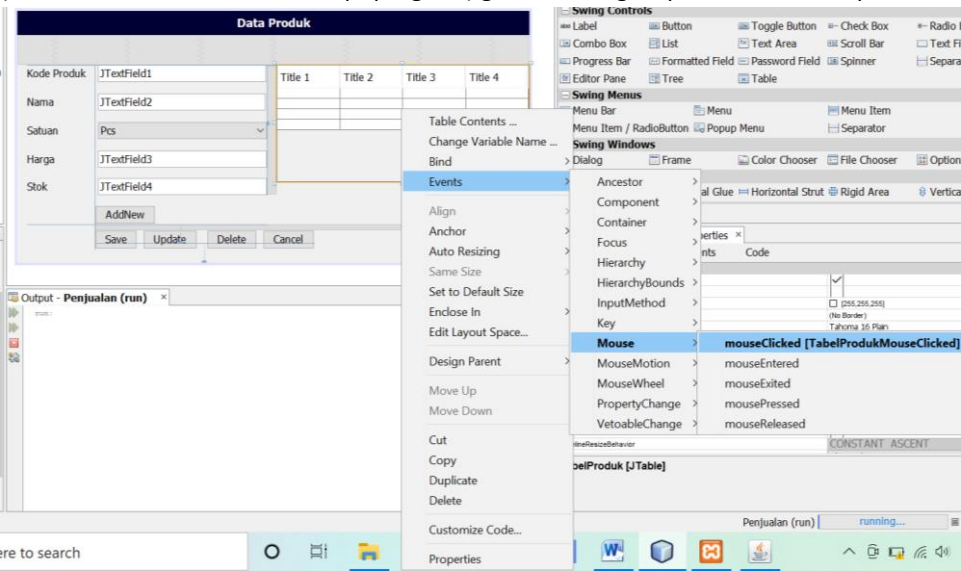
22. Tambahkan script berikut guna button save berfungsi di script button save
- ```
//menghidupkan save karena akan melakukan input data baru dan mengfungsikan button save sehingga
sementara mematikan update dan delete
jButtonSave.setEnabled(true);
jButtonUpdate.setEnabled(false);
jButtonDelete.setEnabled(false);

//setelah fungsi simpan berfungsi maka menghidupkan kembali semua komponen agar bisa digunakan
kembali
seteditOn();
```

23. Tuliskan koding guna menampilkan data ke form pada saat table di klik
- ```
//menampilkan data ke form saat data pada tabel di klik
void tblKeForm(){
jTextKodeProduk.setText(tabModel.getValueAt(TabelProduk.getSelectedRow(),0)+"");
jTextNama.setText(tabModel.getValueAt(TabelProduk.getSelectedRow(),1)+"");
jComboBoxSatuan.setSelectedItem(tabModel.getValueAt(TabelProduk.getSelectedRow(),2)+"");
jTextHarga.setText(tabModel.getValueAt(TabelProduk.getSelectedRow(),3)+"");
jTextStok.setText(tabModel.getValueAt(TabelProduk.getSelectedRow(),4)+"");

//mematikan tombol save, menghidupkan button update, delete dan cancel (guna meminimalis tidak save
data kembar)
jButtonUpdate.setEnabled(true);
jButtonDelete.setEnabled(true);
jButtonCancel.setEnabled(true);
jButtonSave.setEnabled(false);
}
```

24. Memanfaatkan script tblKeForm( ) yakni dengan cara klik kanan table pilih event – Mouse – mouseClicked (double klik akan terarah ke script program) guna menangani proses saat tabel produk di klik



Setelah di double klik, implementasikan koding seperti dibawah ini

```
//menghidupkan semua komponen guna memfungsikan button update
seteditOn();
//mematikan JTextKodeProduk agar tidak bisa diedit karena bersifat unik
jTextKodeProduk.setEnabled(false);
```

```
//memanggil tblKeForm dan menampilkan data table ke textfield
tblKeForm();
```

25. Tuliskan script ke button update berikut

```
private void jButtonUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    String KodeProduk=jTextKodeProduk.getText();
    String NamaProduk=jTextNama.getText();
    String Satuan=jComboBoxSatuan.getSelectedItem().toString();
    String Harga=jTextHarga.getText();
    String Stok=jTextStok.getText();

    java.util.Date tanggal = new java.util.Date();
    java.text.SimpleDateFormat TanggalFormat= new java.text.SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    String CreateDate=TanggalFormat.format(tanggal);

    if (KodeProduk.isEmpty() ) {
        JOptionPane.showMessageDialog(null,"Kode produk tidak boleh kosong");
        jTextKodeProduk.requestFocus();
    }else if (NamaProduk.isEmpty()) {
        JOptionPane.showMessageDialog(null,"Nama produk tidak boleh kosong");
        jTextNama.requestFocus();
    }else if (Satuan.isEmpty()) {
        JOptionPane.showMessageDialog(null,"Satuan produk tidak boleh kosong");
        jComboBoxSatuan.requestFocus();
    }else if (Harga.isEmpty()) {
        JOptionPane.showMessageDialog(null,"Harga produk tidak boleh kosong");
        jTextHarga.requestFocus();
    }else if (Stok.isEmpty()) {
        JOptionPane.showMessageDialog(null,"Stok produk tidak boleh kosong");
        jTextStok.requestFocus();
    }
    else if (!(new Scanner(Harga).hasNextInt())) {
        JOptionPane.showMessageDialog(null,"Harga produk harus angka");
        jTextHarga.requestFocus();
    }else if (!(new Scanner(Stok).hasNextInt())) {
        JOptionPane.showMessageDialog(null,"Stok produk harus angka");
        jTextStok.requestFocus();
    }else{
        try{
            Connection conn=(Connection)koneksi.koneksiDB();
            Statement stt=conn.createStatement();
            stt.executeUpdate("UPDATE produk SET NamaProduk='"+NamaProduk+"', Satuan='"+Satuan+"',
Harga='"+Harga+"', "+
                "Stok='"+Stok+"' WHERE KodeProduk='"+KodeProduk+"'");
            BersihData();
            tampilData();
            SetEditOff();
            JOptionPane.showMessageDialog(this,"Data berhasil
diubah","Success",JOptionPane.INFORMATION_MESSAGE);
        } catch(SQLException e){
            JOptionPane.showMessageDialog(this,"Ubah data
gagal\n"+e.getMessage(),"Error",JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

26. Tuliskan script ke button Delete

```
private void jButtonDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    String KodeProduk=jTextKodeProduk.getText();

    if (KodeProduk.isEmpty() ) {
        JOptionPane.showMessageDialog(null,"Kode produk tidak boleh kosong");
```



```

        jTextKodeProduk.requestFocus();
    }else if(JOptionPane.showConfirmDialog(null,"Apakah anda yakin akan menghapus data ini?",

"Informasi",JOptionPane.OK_CANCEL_OPTION,JOptionPane.INFORMATION_MESSAGE)==JOptionPane.OK_OP
TION){
    try{
        Connection conn=(Connection)koneksi.koneksiDB();
        Statement stt=conn.createStatement();
        stt.executeUpdate("DELETE FROM produk WHERE KodeProduk='"+KodeProduk+"'");
        BersihData();
        tampilData();
        SetEditOff();
        JOptionPane.showMessageDialog(this,"Data berhasil di
hapus","Success",JOptionPane.INFORMATION_MESSAGE);
    } catch(SQLException e){
        JOptionPane.showMessageDialog(this,"Delete data
gagal\n"+e.getMessage(),"Error",JOptionPane.ERROR_MESSAGE);
    }
}
}

```

27. Tuliskan script ke button Cancel

```

//menangani proses saat button cancel di klik
private void jButtonCancelActionPerformed(java.awt.event.ActionEvent evt) {
    BersihData();
    jButtonSave.setEnabled(false);
    jButtonUpdate.setEnabled(false);
    jButtonDelete.setEnabled(false);
    SetEditOff();
}

```