

Judul : Membuat Koneksi Database Mysql dan Perintah CRUD
Tingkat : Pemula
Email : kholidiwan@gmail.com
Mohon maaf bila masih terjadi kesalahan dalam penulisan. Silakan kritik sarannya ke email diatas.

SAMPLE CRUD JAVA & MYSQL

Melakukan operasi database pada java dan mysql CRUD (Create, Read, Update, Delete. Sebelum masuk materi ini kita memang sudah harus paham mengenai Syntax Query Sql seperti perintah:

- *Select * from nama_table* untuk menampilkan seluruh record pada database.
- *Select * from nama_table where alamat="Jakarta"* untuk menampilkan record yang mengandung yang memiliki value Jakarta pada field alamat jakarta.
- *Insert into nama_table values (record);*
dan masih banyak lagi query sql yang harus dipahami.

Adapun tools dan software yang dibutuhkan:

1. JDK
2. XAMPP
3. NetBeans

Usahakan Masing2 software merupakan versi terbarunya. Disini saya menggunakan jdk versi 1.7 dan netbeans 8.0. Disini juga dibutuhkan library *mysql-conector-java.jar* untuk menghubungkan MySql dengan Java pada Netbeans. Library dan software bisa di download gratis.

Membuat Database & Table MySQL

kita akan membuat sebuah database baru bernama kampus. Lalu kita buat table bernama mahasiswa. buat table sebagai berikut

nama database : kampus

nama table : mahasiswa

Nama field	Type
Npm	varchar (20)
nama	varchar (25)
jenkel	varchar (10)
prodi	varchar (20)
alamat	text(100)

Membuat Project Baru

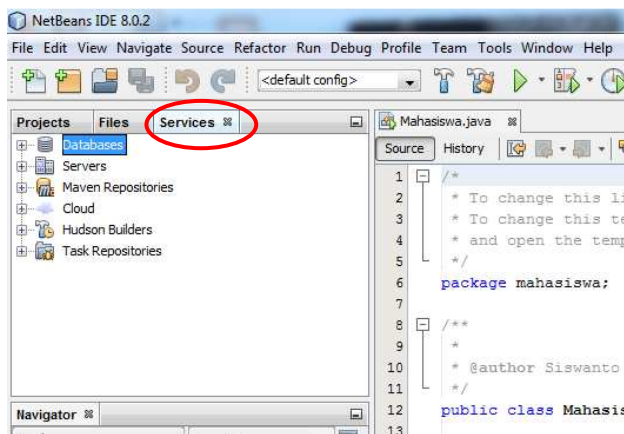
1. Buka netbeans
2. Buat project baru dengan cara klik file -> new project
3. Pilih katagori: java dan project: java application kemudian next
4. Berikan nama project: mahasiswa dan jangan lupa centang create main project. Lalu finish.

Menambahkan Library *mysql-conector-java-jar*

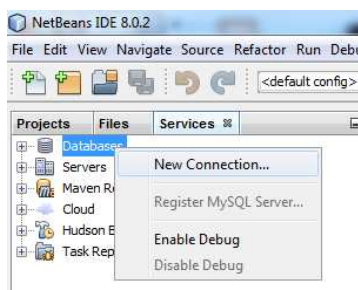
Ada beberapa cara untuk menambahkan library pada java. Menambahkan libray bisa dengan cara mengimport file library pada pada tools library di netbeans atau dengan mengcopy file library langsung pada folder ext pada java. Disini kita akan mencoba dengan cara mengcopy file library mysql-conector-java.jar ke folder ext pada directory C:\Program File\java. Langkahnya dengan cara:

1. copy file *mysql-conector-java.jar* dan pastekan di direktori C:\Program Files\Java\jdk1.7.0_80\jre\lib\ext dan C:\Program Files\Java\jre7\lib\ext.

2. pilih tab Service pada kiri netbeans.



3. klik kanan pada direktori database, kemudian pilih new connection



4. Pilih mysql (Conector/J driver), kemudian next.
5. Kemudian isikan Host: localhost, Port: 33006, Database: kampus, User Name: root, Password tidak usah diisi. Kecuali jika kita menggunakan password pada host mysql (Xampp atau lainnya).
6. Kemudian klik Test Connection, jika koneksi berhasil maka akan ada tulisan Connection Succeeded. Kemudian finish.

Membuat Class Koneksi

Class koneksi adalah class yang menghubungkan database Mysql kita dengan java. Buat sebuah class dengan nama koneksi dengan source sebagai berikut:

Class koneksi

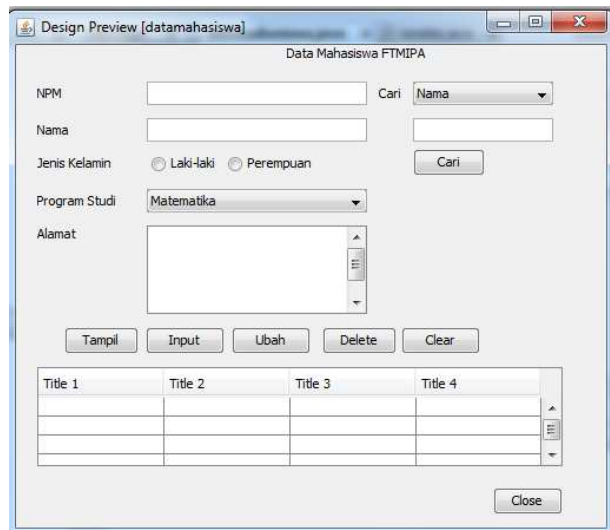
```
package chelonia;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class koneksi {
    public koneksi(){}
    Connection con=null;
    Statement st=null;
    public Connection open(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/kampus","root","");
            return con;
        } catch (SQLException sqle) {
            System.out.print("Tidak Ada Koneksi Yang Terbuka");
            return null;
        } catch (Exception e){
            javax.swing.JOptionPane.showMessageDialog(null, "Koneksi Gagal"+e.getMessage());
            System.out.print("Tidak Bisa Membuka Koneksi"+e.getMessage());
            return null;
        }
    }
    public void QUERY(String data, String option){
        try {
            if (con==null)con=open();
            if (st==null)st=con.createStatement();
            st.executeUpdate(data);
            javax.swing.JOptionPane.showMessageDialog(null, "Data Berhasil"+option);
        } catch (SQLException sqle) {
            sqle.printStackTrace();
            javax.swing.JOptionPane.showMessageDialog(null,"Data gagal"+option+"salahnya:"+sqle.getMessage());
        }
    }
}
```

Membuat JFrame Data Mahasiswa

1. Buat JFrameForm baru pada package.
2. Beri nama JFrameForm dengan nama datamahasiswa
3. Desain tampilan sebagai berikut.



4. Berikan nama variable pada komponen swing sebagai berikut:

Komponen	Nama Variable
JTextField NPM	txtnpm
JTextFiled Nama	txtnama
JRadioButton Laki-laki	rblaki
JRadioButton Perempuan	rbcewe
JComboBox Program Studi	cbojurusan
JTextArea Alamat	txtalamat
JButton Input	btninput
JButton Clear	btnbersih
JButton Ubah	btnubah
JButton Delete	tndelete
JButton Close	btnclose
JComboBox Cari	cbocari
JTextField Cari	txtcari
JButton Cari	btncari

5. Import beberapa library, deklarasikan beberapa variable dan buat method pada masing-masing button dengan source sebagai berikut:

```
package mahasiswa;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import sun.misc.Cleaner;

public class katagoribarang extends javax.swing.JFrame {
    String header [] ={"NPM","Nama","Jenis Kelamin","Program Studi","Alamat"};
    Connection con;
    Statement st;
    koneksi kon=new koneksi();
    private DefaultTableModel tabMode;
    String query;

    public datamahasiswa() {
        initComponents();
    }

    private void clear(){
        txtnpm.setText("");
        txnama.setText("");
        txtalamat.setText("");
        txtnpm.requestFocus();
    }

    private void removeTable(){
        try {
            for (int t=tabMode.getRowCount();t>0;t--)
                {tabMode.removeRow(0);}
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    private void tampilDataTabel(){
        removeTable();
        try {
            con=kon.open();
            st=con.createStatement();
            query="SELECT*from mahasiswa";
            ResultSet rs= st.executeQuery(query);
            while(rs.next()){
                String a=rs.getString("npm");
                String b =rs.getString("nama");
                String c =rs.getString("jenkel");
                String d =rs.getString("prodi");
                String e =rs.getString("alamat");
                String data[]={a,b,c,d,e};
                tabMode.addRow(data);
            }
        } catch (SQLException sqle) {
            JOptionPane.showMessageDialog(null, "Data gagal
ditampilkan"+sqle);
        }
    }
}
```

```

private void Search(){
    removeTable();
    try {
        con=kon.open();
        st=con.createStatement();
        query="select*from mahasiswa where "+cbocari.getSelectedItem()+" like '%"+txtcari.getText()+"% '";
        ResultSet rs=st.executeQuery(query);
        while(rs.next()){
            String snpm = rs.getString("npm");
            String snama = rs.getString("nama");
            String sjenkel = rs.getString("jenkel");
            String sprodi=rs.getString("prodi");
            String salamat=rs.getString("alamat");
            String data []={snpm,snama,sjenkel,sprodi,salamat,};
            tabMode.addRow(data);
        }
    } catch (SQLException sqle) {
        JOptionPane.showMessageDialog(null, "data gagal masuk tabel"+sqle.getMessage());
    }
}

```

```

private void input(){
    String kelamin="";
    if (rblaki.isSelected()){
        kelamin ="Laki-laki";
    }
    else if (rbcewe.isSelected()){
        kelamin ="Perempuan";
    }
    try {
        kon.QUERY("insert into mahasiswa (npm,nama,jenkel,prodi,alamat) values ('"+txtnpm.getText()+"','"+
            txtnama.getText()+"','"+
            kelamin+"','"+
            cbojurusan.getSelectedItem()+"','"+
            txtalamat.getText()+"')", "simpan");
    } catch (Exception sqle) {
        JOptionPane.showMessageDialog(rootPane, "Data Gagal di Simpan"+sqle.getMessage());
    }
}

```

```

private void update(){

    String kelamin="";
    if (rblaki.isSelected()){
        kelamin ="Laki-laki";
    }
    else if (rbcewe.isSelected()){
        kelamin ="Perempuan";
    }
    try {
        kon.QUERY("update mahasiswa set nama='"+txtnama.getText()+"',jenkel='"+
            kelamin+"',prodi='"+
            cbojurusan.getSelectedItem()+"',alamat='"+
            txtalamat.getText()+"'where npm='"+
            txtnpm.getText()+"' ", "Ubah");

        tampilDataTabel();
        clear();
    } catch (Exception sqle) {
        JOptionPane.showMessageDialog(rootPane, "data gagal diubah"+sqle.getMessage());
    }
}

```

```

private void ShowDataTabelKeText(){
    txtnpm.setText(String.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 0)));
    txtnama.setText(String.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1)));
    cboprodi.setSelectedItem(String.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 3)));
    txtalamat.setText(String.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 4)));
    txtnpm.setEditable(true);
    txtnama.setEditable(true);
    cboprodi.setEditable(true);
    txtalamat.setEditable(true);
}

```

```

private void Delete(){
    if (JOptionPane.showConfirmDialog
        (this,"Apakah Anda Yakin Akan Menghapus Data Ini???", "Konfirmasi",
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE)==
        JOptionPane.YES_OPTION)
    {
        try {
            kon.QUERY("delete from mahasiswa where npm='"+txtnpm.getText()+"' ","hapus");
            tampilDataTabel();
            clear();
        } catch (Exception sqle) {
            JOptionPane.showMessageDialog(rootPane,"data gagal dihapus"+sqle.getMessage());
        }
    }
}

```