

# Manipulasi Database Dengan Netbeans

Oleh : Eko Zulkaryanto

<http://zulkaryanto.wordpress.com>

Sumber : <http://hendrosteven.wordpress.com/2007/10/16/manipulasi-mysql-dengan-netbeans/> dan <http://hendrosteven.wordpress.com/2007/10/18/membuat-aplikasi-database-sederhana>

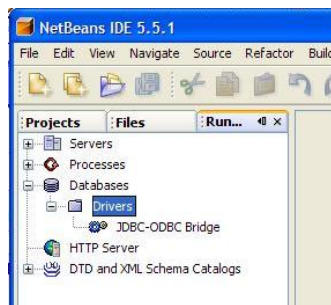
Netbeans adalah salah satu IDE Java yang sangat komplit khususnya fasilitas yang disediakan untuk mengembangkan aplikasi berbasis Java. Pada tulisan ini saya ingin menunjukkan bagaimana kita menggunakan Netbeans untuk memanipulasi database. Database yang digunakan adalah MySQL.

Pra-syarat:

1. Netbeans IDE (dalam tulisan ini saya menggunakan versi 5.5.1)
2. MySQL server (dalam tulisan ini saya menggunakan veris 5.0)

Ok! berikut langkah-langkah yang kita lakukan.

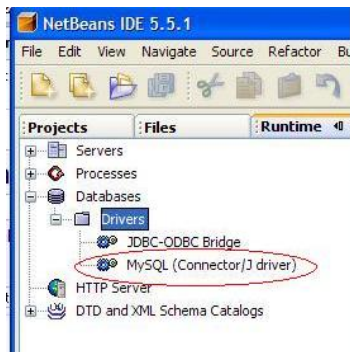
- Buatlah database dalam MySQL misalnya dengan nama dbtest. Untuk tabel kita akan buat menggunakan Netbeans.



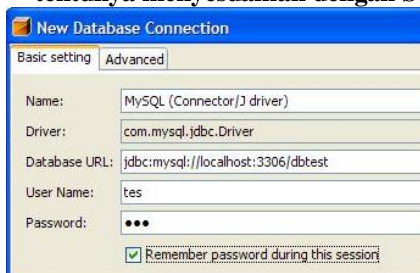
- Jalankan Netbeans, pilih tab RuntimeSelanjutnya kita akan menciptakan koneksi ke database kita. Untuk itu kita butuh MySQL konektor. Pada tulisan ini saya menggunakan versi 5.0. Silahkan download [disini](#). Ekstraklah MySQL Konektor yang telah didownload
- Selanjutnya kita daftarkan Konektor tersebut ke Netbeans
- Pada tampilan seperti gambar di atas klik kanan pada Driver, pilih New Driver.
- Pada window New JDBC Driver, pilih tombol Add
- Carilah Konektor MySQL dilokasi yang sesuai



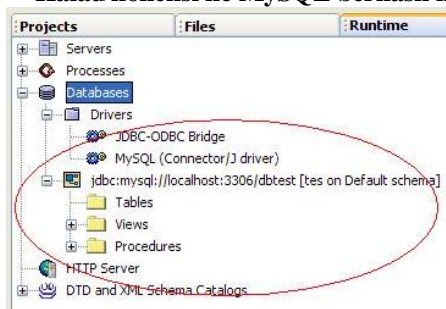
- Kemudian OK
- Maka Konektor MySQL sudah terdaftar ke Netbeans



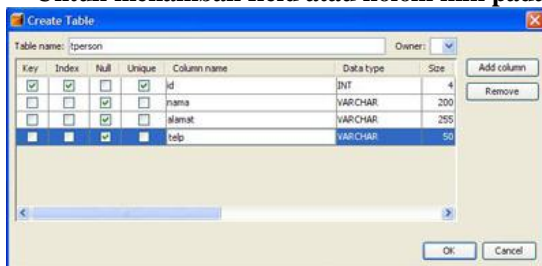
- Untuk menciptakan koneksi ke database kita (dbtest), klik kanan pada MySQL Connector seperti pada gambar di atas, dan pilih Connect Using...
- Pada Window Database Connection, masukan database url, user dan password. Data ini tentunya menyesuaikan dengan Server MySQL Anda.



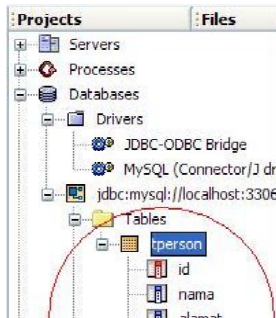
- Kemudian pilih OK 2x
- Kalau koneksi ke MySQL berhasil maka akan ditampilkan tampilan seperti berikut ini



- Selanjutnya kita akan membuat table untuk database kita.
- Klik kanan pada folder Table dan pilih Create Table
- Pada Window Create Table isilah nama table dan field-field yang diinginkan
- Untuk menambah field atau kolom klik pada tombol Add Column



- Selanjutnya pilih OK untuk mengakhiri pembuatan table
- Jika berhasil akan ditampilkan tampilan seperti gambar berikut



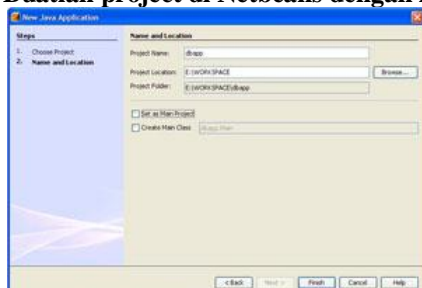
- Sampai disini kita telah dapat menambahkan table ke dalam database kita menggunakan Netbeans IDE. Selanjutnya untuk memanipulasi data dalam table tperson yang baru dibuat, kita dapat menggunakan perintah-perintah SQL langsung dari Netbeans IDE juga.
- Pada Table tPerson klik kanan dan pilih Execute Command...
- Maka pada bagian kanan dari Netbeans IDE akan ditampilkan editor tempat kita menuliskan perintah SQL kita.



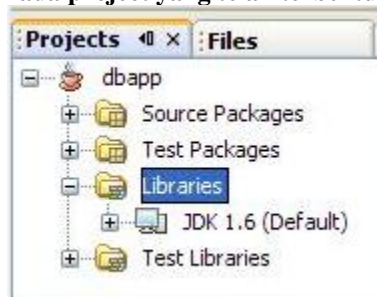
- Untuk mengeksekusi perintah SQL yang sudah kita tulis, silahkan klik pada icon Run SQL seperti pada gambar di atas.
- Jika SQL kita berhasil dieksekusi maka pada status bar Netbeans akan terdapat pesan bahwa SQL berhasil dieksekusi.
- Untuk menampilkan data yang sudah ada pada table tPerson kita dapat juga menuliskan perintah SQL pada SQL Editornya misalnya `SELECT * FROM tperson`, maka akan ditampilkan seluruh data yang ada pada table tPerson pada bagian bawah dari SQL Editornya.
- Hal yang sama juga dapat kita lakukan untuk operasi *Update* atau *Delete*.

Saya merasa perlu menunjukan mudahnya untuk membuat aplikasi database sederhana menggunakan Netbeans. Berikut langkah-langkahnya:

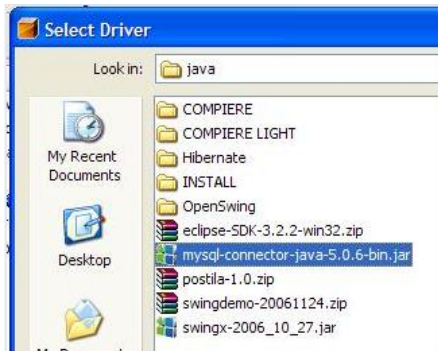
- Pada Netbeans IDE pilih menu *File-New Project...*
- Pada Window *New Project* pilih *Categories General* dan *Project Java Application*, kemudian pilih *Next*.
- Buatlah project di Netbeans dengan nama misalnya *dbapp*, kemudian pilih *Finish*



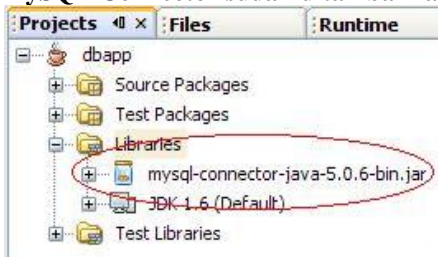
- Pada project yang telah terbentuk, klik kanan pada *Libraries* pilih *Add Jar/Folder...*



- Carilah lokasi MySQL Connector



- MySQL Connector sudah ditambahkan ke project.



- Sampai disini kita sudah siap untuk membuat aplikasinya. Klik kanan pada nama proejet *dbapp*, pilih *New-Java Class...*
- Pada window *New Class*, ketik *Person* pada *Class Name* dan *db.app* pada *Package* kemudian *Finish*.
- Ubahlah source *Person.java* menjadi sbb:

```

package db.app;
public class Person {
private int id;
private String nama;
private String alamat;
private String telp;
/** Creates a new instance of Person */
public Person() {
}
public int getId() {
return id;
}
public void setId(int id) {
this.id = id;
}
public String getNama() {
return nama;
}
public void setNama(String nama) {
this.nama = nama;
}
public String getAlamat() {
return alamat;
}
public void setAlamat(String alamat) {
this.alamat = alamat;
}
public String getTelp() {
return telp;
}
public void setTelp(String telp) {
this.telp = telp;
}
}

```

```
}  
}
```

- Tambahkan lagi Class lain dengan nama *PersonManager*, dan ubahlah sourcena menjadi sbb:

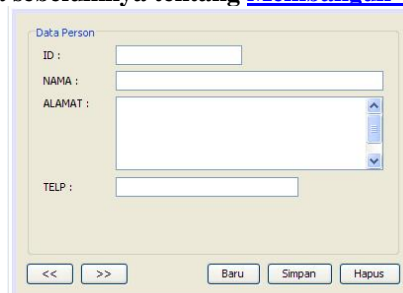
```
package db.app;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.util.ArrayList;  
import java.util.List;  
/**  
 *  
 * @author hendro  
 */  
public class PersonManager {  
    Connection con = null;  
    Statement st = null;  
    String url = "jdbc:mysql://localhost:3306/dbtest";  
    String user = "tes";  
    String pass = "tes";  
    /** Creates a new instance of PersonManager */  
    public PersonManager() {  
        try{  
            Class.forName("com.mysql.jdbc.Driver").newInstance();  
            con = DriverManager.getConnection(url,user,pass);  
            st=con.createStatement();  
        }catch(Exception ex){  
            ex.printStackTrace();  
        }  
    }  
    public List  
    getPersons(){  
        ResultSet rs = null;  
        List  
        persons = new ArrayList  
        ();  
        try{  
            rs = st.executeQuery("SELECT ID,NAMA,ALAMAT,TELP FROM tperson");  
            while(rs.next()){  
                Person p = new Person();  
                p.setId(rs.getInt(1));  
                p.setNama(rs.getString(2));  
                p.setAlamat(rs.getString(3));  
                p.setTelp(rs.getString(4));  
                persons.add(p);  
            }  
        }catch(Exception ex){  
            ex.printStackTrace();  
        }  
        return persons;  
    }  
    public int insert(Person p){  
        int result = 0;  
        try{  
            result = st.executeUpdate("INSERT INTO tperson(id,nama,alamat,telp) " +  
            "VALUES("+p.getId()+","+p.getNama()+","+p.getAlamat()+","+p.getTelp()+")");  
        }catch(Exception ex){  
            ex.printStackTrace();  
        }  
    }  
}
```

```

}
return result;
}
public int delete(Person p){
int result = 0;
try{
result = st.executeUpdate("DELETE FROM tperson WHERE ID="+p.getId()+""");
}catch(Exception ex){
ex.printStackTrace();
}
return result;
}
}
}

```

- Selanjutnya kita akan membuat user Interfacenya menggunakan Swing (Netbean matisse). Tambahkan *JFrame Form...* dengan *Class Name MyWindow* dan desainlah tampilannya menjadi sbb:
- Silahkan baca tulisan saya sebelumnya tentang [Membangun GUI dengan Netbeans](#)



- Untuk masing-masing component pada Form di atas diganti namanya menjadi sbb. Untuk mengganti nama masing-masing component klik kanan pada component yang diinginkan.
  - text ID -> txtId
  - text NAMA -> txtNama
  - text ALAMAT -> txtAlamat
  - text TELP -> txtTelp
  - Tombol << -> btnPrev
  - Tombol >> -> btnNext
  - Tombol Baru -> btnNew
  - Tombol Simpan -> btnSave
  - Tombol Hapus -> btnDelete
- Untuk setiap Text yang ada atur property Editablenya menjadi false (dengan melepas centangan pada Propertiesbar).
- Selanjutnya tambahkan event untuk masing-masing tombol (Baca tulisan saya mengenai [Membangun GUI dengan Netbeans](#))
- Berikut kode pada Tombol btnNew

```

private void btnNewActionPerformed(java.awt.event.ActionEvent evt) {
txtId.setText("");
txtNama.setText("");
txtAlamat.setText("");
txtTelp.setText("");
txtId.setEditable(true);
txtNama.setEditable(true);
txtAlamat.setEditable(true);
txtTelp.setEditable(true);
txtId.requestFocus();
}

```

- Selanjutnya untuk tombol btnSave

```

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
Person p = new Person();
p.setId(Integer.parseInt(txtId.getText()));
}

```

```

p.setNama(txtNama.getText());
p.setAlamat(txtAlamat.getText());
p.setTelp(txtTelp.getText());
if (pmgr.insert(p)>0) {
loadData();
currentRow = persons.size()-1;
bindData();
JOptionPane.showMessageDialog(this,"Data berhasil disimpan","Informasi",
JOptionPane.INFORMATION_MESSAGE);
txtId.setEditable(false);
txtNama.setEditable(false);
txtAlamat.setEditable(false);
txtTelp.setEditable(false);
}else{
JOptionPane.showMessageDialog(this,"Data gagal disimpan","Informasi",
JOptionPane.INFORMATION_MESSAGE);
}
}

```

- Untuk tombol btnDelete

```

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
Person p = persons.get(currentRow);
if (pmgr.delete(p)>0) {
loadData();
currentRow = currentRow-1;
bindData();
JOptionPane.showMessageDialog(this,"Data berhasil dihapus","Informasi",
JOptionPane.INFORMATION_MESSAGE);
}else{
JOptionPane.showMessageDialog(this,"Data gagal dihapus","Informasi",
JOptionPane.INFORMATION_MESSAGE);
}
}

```

- Untuk tombol btnPrev

```

private void btnPrevActionPerformed(java.awt.event.ActionEvent evt) {
if (currentRow>0) {
--currentRow;
}
bindData();
}

```

- Untuk tombol btnNext

```

private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {
if (currentRow
++currentRow;
}
bindData();
}

```

- Kemudian tambahkan beberapa variabel berikut pada Class MyWindow.java  
PersonManager pmgr = new PersonManager();  
List<Person> persons = new ArrayList<Person>();  
int currentRow = 0;

- Dan tambahkan 2 method berikut pada Class MyWindow.java

```

private void loadData() {
persons = pmgr.getPersons();
}
private void bindData() {
if (persons.size()>0) {
Person p = persons.get(currentRow);

```

```

txtId.setText(p.getId()+"");
txtNama.setText(p.getNama());
txtAlamat.setText(p.getAlamat());
txtTelp.setText(p.getTelp());
}else{
currentRow = 0;
txtId.setText("");
txtNama.setText("");
txtAlamat.setText("");
txtTelp.setText("");
}
}
}

```

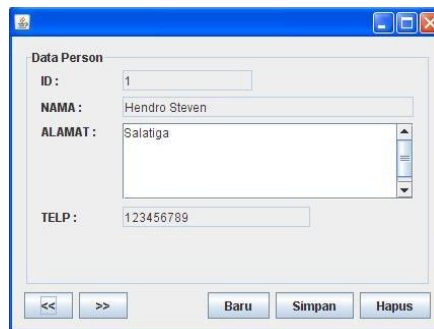
- Dan yang terakhir ubahlah Constructor Class MyWindow.java menjadi

```

public MyWindow() {
initComponents();
loadData();
bindData();
}

```

- Sampai disini kita sudah menyelesaikan aplikasi database sederhana ini, silahkan dicoba untuk dijalankan.



- Bagaimana? gak susah bukan? Tentunya ini hanya aplikasi sederhana dan belum memperhatikan efisiensi code program dan lain-lain. Tapi semoga dapat menjadi titik awal

untuk mulai belajar Java.

Selamat Mencoba...