



TUTORIAL APLIKASI CRUD JAVA

Dengan menggunakan MySQL

Contents

APLIKASI CRUD JAVA.....	2
PENDAHULUAN.....	2
PERSIAPAN.....	2
DATABASE.....	2
TABEL.....	2
MEMULAI PROJECT.....	3
PROJECT.....	3
MENAMBAHKAN LIBRARIES.....	3
PACKAGES.....	4
KELAS MyConnection.....	4
KELAS MahasiswaTableModel.....	5
KELAS FormMahasiswa (Design).....	6
KELAS FormMahasiswa (Source).....	8
TAMBAHAN.....	12
MENU.....	12

Wendy, S.Kom., M.Sc.

APLIKASI CRUD JAVA

PENDAHULUAN

Dalam sebuah aplikasi database, terdapat 4 buah fungsi dasar yaitu Create, Read, Update dan Delete atau yang lebih dikenal dengan istilah CRUD.

Create merupakan fungsi yang dipergunakan untuk menambahkan isi entry baru pada tabel dalam database yang sedang dikelola.

Read merupakan fungsi yang dipergunakan untuk membaca isi sebuah entry pada tabel dalam database yang sedang dikelola.

Update merupakan fungsi yang dipergunakan untuk mengubah isi sebuah entry pada tabel dalam database yang sedang dikelola.

Sedangkan delete merupakan fungsi yang dipergunakan untuk menghapus isi sebuah entry pada tabel dalam database yang sedang dikelola.

Pada tutorial ini, akan dijelaskan bagaimana membuat aplikasi CRUD Java dengan menggunakan konektor MySQL JDBC driver untuk data Mahasiswa serta menggunakan IDE Netbeans.

PERSIAPAN

Database dan tabel yang digunakan untuk aplikasi ini sudah dipersiapkan oleh teknisi lab pada server lab sehingga Anda cukup mempergunakan server tersebut.

Untuk yang hendak menggunakan perangkat pribadi bisa membuat database dan tabel yang disesuaikan dengan detail di bawah ini.

Selain itu file driver untuk konektor dapat dicopy dari server ftp lab stmik.

DATABASE

Adapun detail dbMahasiswa adalah sebagai berikut:

- Hostname : (d disesuaikan)
- Databasename : dbMahasiswa
- Username : (d disesuaikan)
- Password : (d disesuaikan)

TABEL

Adapun detail tblMahasiswa adalah sebagai berikut:

- | | | | |
|----------------|-------------|----------|----------------|
| • id | int | Not null | Auto Increment |
| • nama | varchar(50) | | |
| • nim | varchar(10) | Not null | Unique |
| • jeniskelamin | varchar(30) | | |
| • telp | varchar(15) | | |
| • programstudi | varchar(50) | | |

MEMULAI PROJECT

PROJECT

Buatlah sebuah project dengan detail berikut:

- Tipe Project Java Application
- Project Name DataMahasiswa
- Project Location (d disesuaikan)
- Use Dedicated Folder for Storing Libraries check
- Create Main Class unchecked

Catatan:

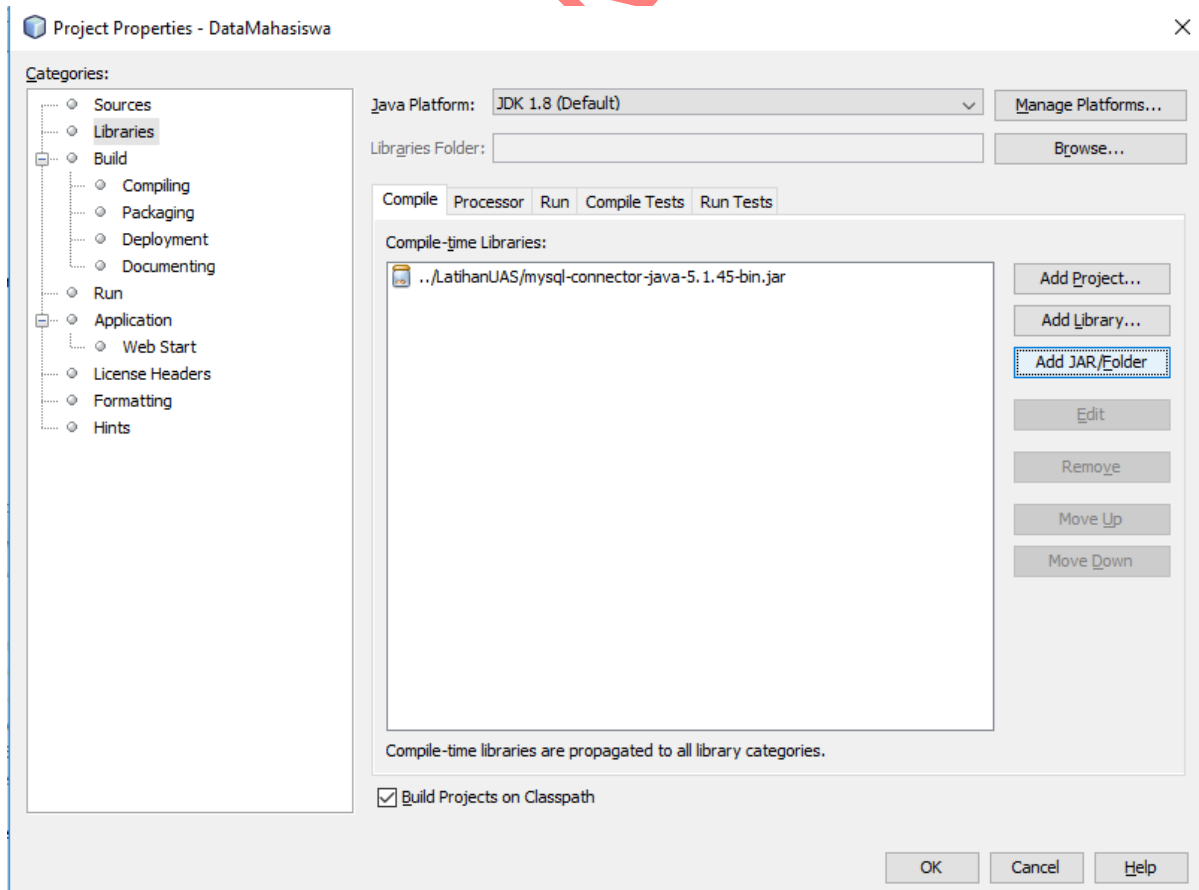
Use Dedicated Folder for Storing Libraries dicentang dengan tujuan menyimpan file library yang dipergunakan dalam folder project sehingga ketika folder project dipindahkan ke lokasi lain, libraries tersebut juga ikut dipindahkan.

Create Main Class tidak dicentang dengan tujuan main class yang hendak digunakan merupakan Jform bukan file yang digenerate oleh Netbeans.

MENAMBAHKAN LIBRARIES

Setelah selesai membuat project baru, klik kanan pada project DataMahasiswa di Projects explorer dan pilih Properties.

Pada bagian Categories pilih Libraries, kemudian pada tab Compile klik tombol Add Jar/Folder. Kemudian cari file driver MySQL Connector yang telah Anda copy dari folder FTP dan klik Open. Selanjutnya Klik tombol OK untuk mengakhiri.



PACKAGES

Buatlah sebuah package dengan nama stmik.lab

KELAS MyConnection

Buatlah sebuah Java Class bernama MyConnection pada package stmik.lab tersebut. Dimana kelas ini akan dipergunakan sebagai koneksi antara aplikasi java dengan database server. Kemudian ketikkan coding berikut:

```
package stmik.lab;
import java.sql.*;

public class MyConnection {
    private Connection myConnection;
    // Konstanta untuk menyimpan detil server
    private final String HOST = "localhost";
    private final String DATABASENAME = "dbMahasiswa";
    private final String USERNAME = "root";
    private final String PASSWORD = "";

    // Constructor kelas MyConnection
    public MyConnection() {
        init();
    }
    // method init untuk memulai kelas MyConnection
    public void init() {
        try{
            Class.forName("com.mysql.jdbc.Driver");
            myConnection=DriverManager.getConnection("jdbc:mysql://"+HOST+
                "/" +DATABASENAME, USERNAME, PASSWORD);
        }catch (Exception e) {
            System.out.println("Gagal mendapatkan koneksi database");
        }
    }
    // fungsi untuk mengembalikan koneksi
    public Connection getMyConnection(){
        return myConnection;
    }
    // fungsi untuk menutup koneksi
    public void destroy(){
        if (myConnection!= null){
            try{
                myConnection.close();
            }catch (Exception e) {}
        }
    }
}
```

KELAS MahasiswaTableModel

Buatlah Kelas MahasiswaTableModel pada package stmik.lab. Dimana kelas ini dipergunakan sebagai model dalam menampilkan data ke tabel. Kemudian tambahkan coding berikut:

```
package stmik.lab;
import java.sql.*;
import java.util.ArrayList;
import javax.swing.table.AbstractTableModel;

public class MahasiswaTableModel extends AbstractTableModel{
    private int colNum=5;
    private int rowNum;
    private final String[] colNames={"NAMA","NIM","JENIS KELAMIN","NO
    TELP","PROGRAM STUDI"};
    private ArrayList<String[]> resultSets;

    public MahasiswaTableModel(ResultSet rs){
        resultSets = new ArrayList<String[]>();
        try{
            while(rs.next()){
                String[] row = {rs.getString("nama"),
                    rs.getString("nim"),
                    rs.getString("jeniskelamin"),
                    rs.getString("telp"),
                    rs.getString("programstudi")
                };
                resultSets.add(row);
            }
        }catch(Exception e){
            System.out.println("Exception in MahasiswaTableModel");
        }
    }
    @Override
    public Object getValueAt(int rowindex, int colindex){
        String[] row=resultSets.get(rowindex);
        return row[colindex];
    }
    @Override
    public int getRowCount(){
        return resultSets.size();
    }
    @Override
    public int getColumnCount(){
        return colNum;
    }
    @Override
    public String getColumnName(int param){
        return colNames[param];
    }
}
```

Konstruktor MahasiswaTableModel menerima input berupa ResultSet (yang diperoleh melalui eksekusi sql statement) kemudian menyimpannya kedalam variabel resultSets.

KELAS FormMahasiswa (Design)

Buatlah sebuah Kelas JFrame bernama FormMahasiswa dengan detail berikut:

- title Data Mahasiswa
- resizable unchecked

Kemudian pada JFrame tersebut tambahkan bernama JPanel bernama `pnIlsian` dan `pnITombol` dan sebuah jTable bernama `tblMahasiswa`.

Panel pnIlsian

Pada panel `pnIlsian` tambahkan komponen berikut:

Type	Nama Variabel	Keterangan
Label	<code>jLabel1</code>	Text: NAMA
Label	<code>jLabel2</code>	Text: NIM
Label	<code>jLabel3</code>	Text: JENIS KELAMIN
Label	<code>jLabel4</code>	Text: NOMOR TELP
Label	<code>jLabel5</code>	Text: PROGRAM STUDI
Text Field	<code>txtNama</code>	Text: Horizontal Size: (d disesuaikan)
Text Field	<code>txtNim</code>	Text: Horizontal Size: (d disesuaikan)
Button Group	<code>buttonGroup1</code>	
Radio Button	<code>rbLaki</code>	Text: LAKI-LAKI ButtonGroup: <code>buttonGroup1</code>
Radio Button	<code>rbPerempuan</code>	Text: PEREMPUAN ButtonGroup: <code>buttonGroup1</code>
Text Field	<code>txtTelp</code>	Text: Horizontal Size: (d disesuaikan)
Combo Box	<code>cbProgram</code>	Model: -PILIH- SISTEM INFORMASI TEKNIK INFORMATIKA

jPanel pnlTombol

Pada panel pnlTombol tambahkan komponen berikut:

Type	Nama	Keterangan
Button	btnTambah	Text: TAMBAH Horizontal Size: (d disesuaikan)
Button	btnUbah	Text: UBAH Horizontal Size: (d disesuaikan)
Button	btnSimpan	Text: SIMPAN Horizontal Size: (d disesuaikan)
Button	btnBatal	Text: BATAL Horizontal Size: (d disesuaikan)
Button	btnHapus	Text: HAPUS Horizontal Size: (d disesuaikan)
Button	btnKeluar	Text: KELUAR Horizontal Size: (d disesuaikan)

Tampilan Akhir Form

Adapun tampilan akhir form adalah sebagai berikut:

Title 1	Title 2	Title 3	Title 4

KELAS FormMahasiswa (Source)

Ada pun kelas yang perlu diimport adalah sebagai berikut:

```
import java.sql;  
import javax.swing.JOptionPane;
```

Adapun fungsi dari kelas yang diimport tersebut adalah:

- java.sql, digunakan untuk manipulasi database dan tabel
- javax.swing.JOptionPane, digunakan untuk menampilkan dialogbox

Berikut variabel global yang dipergunakan pada kelas FormMahasiswa:

```
private int pilih;  
private final MyConnection myConnection = new MyConnection();  
private Boolean baru;
```

Adapun fungsi variabel di atas adalah:

- **pilih**, untuk menampung index baris (row) yang dipilih oleh user pada tblMahasiswa
- **myConnection**, untuk memulai koneksi database
- **baru**, untuk menandai data yang dientry termasuk data baru atau bukan

Method FormMahasiswa

```
public FormMahasiswa() {  
    initComponents();  
    bersih();  
    enableForm(false);  
    btnUbah.setEnabled(false);  
    btnHapus.setEnabled(false);  
    tampilkan();  
}
```

Baris perintah di atas digunakan untuk memulai semua komponen yang ada pada FormMahasiswa, membersihkan dan men-disable form isian serta menampilkan isi tabel pada database ke tblMahasiswa.

Method bersih

```
private void bersih(){  
    txtNama.setText("");  
    txtNim.setText("");  
    txtTelp.setText("");  
    cbProgram.setSelectedIndex(0);  
    rbLaki.setSelected(true);  
}
```

Baris perintah di atas digunakan untuk membersihkan form isian/mengembalikan form isian ke kondisi awal.

Method enableForm

```
private void enableForm(Boolean bool) {  
    txtNama.setEnabled(bool);  
    txtNim.setEnabled(bool);  
    txtTelp.setEnabled(bool);  
    cbProgram.setEnabled(bool);  
    rbLaki.setEnabled(bool);  
    rbPerempuan.setEnabled(bool);  
    btnTambah.setEnabled(!bool);  
    btnSimpan.setEnabled(bool);  
    btnBatal.setEnabled(bool);  
}
```

Baris perintah diatas digunakan untuk men-enable atau men-disable form isian. Terkecuali untuk tombol **btnUbah**, **btnHapus** dan **btnKeluar**.

Method tampilkan

```
private void tampilkan() {
    try{
        Statement stm =
myConnection.getMyConnection().createStatement();
        ResultSet rs = stm.executeQuery("SELECT * FROM tblMahasiswa");
        MahasiswaTableModel myModel = new MahasiswaTableModel(rs);
        tblMahasiswa.setModel(myModel);
    }catch(SQLException e){
        e.printStackTrace();
    }
}
```

Baris perintah di atas digunakan untuk membuat sebuah sql statement, menampung hasil eksekusi sql statement kemudian menampilkannya pada tblMahasiswa.

Method btnKeluarActionPerformed

```
private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}
```

Baris perintah di atas digunakan untuk menghentikan aplikasi.

Method btnTambahActionPerformed

```
private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    bersih();
    enableForm(true);
    baru = true;
}
```

Baris perintah di atas digunakan untuk membersihkan dan men-enable form isian kemudian menandai bahwa data yang akan diisikan merupakan data baru.

Method btnUbahActionPerformed

```
private void btnUbahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    enableForm(true);
    btnUbah.setEnabled(false);
    btnHapus.setEnabled(false);
    txtNim.setEnabled(false);
    baru = false;
}
```

Baris perintah di atas digunakan untuk men-enable form isian kemudian men-disable tombol **btnUbah**, **btnHapus** dan **txtNim** (karena merupakan field unique) serta menandai bahwa data yang akan diisikan bukan data baru.

Method btnBatalActionPerformed

```
private void btnBatalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    bersih();
    enableForm(false);
    btnHapus.setEnabled(false);
    btnUbah.setEnabled(false);
}
```

Baris perintah di atas digunakan untuk mengembalikan kondisi form ke keadaan semula.

Method btnSimpanActionPerformed

```
private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String jk;
    if (rbLaki.isSelected()){
        jk = "LAKI-LAKI";
    }else{
        jk = "PEREMPUAN";
    }
    if (baru) {
        try {
            String sql = "INSERT INTO tblMahasiswa "
                + "(nama,nim,jeniskelamin,telp,programstudi) "
                + "VALUES ('"+
                txtNama.getText()+"', '"+
                txtNim.getText()+"', '"+
                jk+"', '"+
                txtTelp.getText()+"', '"+
                cbProgram.getSelectedItem().toString()+"')";

myConnection.getMyConnection().prepareStatement(sql).execute();
            JOptionPane.showMessageDialog(null, "Berhasil disimpan!");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, ex);
        }
    }else{
        try {
            String sql = "UPDATE tblMahasiswa SET "+
                "nama='"+txtNama.getText()+"', "+
                "jeniskelamin='"+jk+"', "+
                "telp='"+txtTelp.getText()+"', "+
                "programstudi='"+
                cbProgram.getSelectedItem().toString()+"'+
                "WHERE nim='"+txtNim.getText()+"'";

myConnection.getMyConnection().prepareStatement(sql).execute();
            JOptionPane.showMessageDialog(null, "Berhasil disimpan!");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, ex);
        }
    }
    bersih();
    tampilkan();
    enableForm(false);
    btnHapus.setEnabled(false);
    btnUbah.setEnabled(false);
}
}
```

Baris perintah di atas digunakan untuk menyimpan data baru maupun data perubahan kemudian mengembalikan form isian ke keadaan semula.

Method btnHapusActionPerformed

```
private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if (JOptionPane.showConfirmDialog(null, "Yakin dihapus?",  
"Peringatan",  
        JOptionPane.YES_NO_OPTION)==JOptionPane.YES_OPTION) {  
        try {  
            String sql = "DELETE FROM tblMahasiswa WHERE "+  
                "nim='"+txtNim.getText()+"'";  
  
myConnection.getMyConnection().prepareStatement(sql).execute();  
            JOptionPane.showMessageDialog(null, "Berhasil dihapus!");  
        } catch (SQLException ex) {  
            JOptionPane.showMessageDialog(null, ex);  
        }  
        bersih();  
        enableForm(false);  
        btnUbah.setEnabled(false);  
        btnHapus.setEnabled(false);  
        tampilkan();  
    }  
}
```

Baris perintah di atas digunakan untuk menanyakan kepada user apakah yakin ingin menghapus data terpilih. Jika user memilih Yes, maka data akan dihapus dan form isian dikembalikan ke kondisi semula.

Method tblMahasiswaClicked

```
private void tblMahasiswaMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    pilih = tblMahasiswa.getSelectedRow();  
    txtNama.setText(tblMahasiswa.getValueAt(pilih, 0).toString());  
    txtNim.setText(tblMahasiswa.getValueAt(pilih, 1).toString());  
    if (tblMahasiswa.getValueAt(pilih, 2).toString().equals("LAKI-  
LAKI"))  
        rbLaki.setSelected(true);  
    else  
        rbPerempuan.setSelected(true);  
    txtTelp.setText(tblMahasiswa.getValueAt(pilih, 3).toString());  
    cbProgram.setSelectedItem(tblMahasiswa.getValueAt(pilih, 4));  
    btnHapus.setEnabled(true);  
    btnUbah.setEnabled(true);  
}
```

Baris perintah di atas digunakan untuk menampilkan data baris yang dipilih oleh user ke form isian kemudian men-enable tombol **btnUbah** dan **btnHapus**.

Method txtNimKeyTyped

```
private void txtNimKeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    if (txtNim.getText().length() >= 10) {  
        evt.consume();  
    }  
}
```

Baris perintah di atas digunakan untuk membatasi entry pada **txtNim** hanya sebanyak 10 karakter.

Method txtTelpKeyTyped

```
private void txtTelpKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (txtNim.getText().length() >= 15) {
        evt.consume();
    }
}
```

Baris perintah di atas digunakan untuk membatasi entry pada `txtNim` hanya sebanyak 15 karakter.

Method formWindowClosing

```
private void formWindowClosing(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    myConnection.destroy();
}
```

Baris perintah di atas digunakan untuk menutup koneksi ketika user menutup aplikasi.

TAMBAHAN

MENU

Pada tampilan Design Form Mahasiswa, tambahkan sebuah MenuBar. Secara default netbeans akan membuat dua buah menu, `jMenu1` (File) dan `jMenu2` (Edit). Pada tutorial ini, kita hanya memerlukan `jMenu1` (File), sehingga `jMenu2` (Edit) dapat kita hapus melalui klik kanan kemudian pilih delete.

Selanjutnya tambahkan dua buah MenuItem pada `jMenu1` (File) yaitu `menuAbout` dan `menuExit` kemudian ubah pada bagian property ubah text kedua menu tersebut menjadi About dan Exit.

Untuk membuat shortcut dari kedua MenuItem yang telah kita tambahkan, pada bagian shortcut (sebelah kanan) dari `menuAbout`, klik dua kali untuk menampilkan jendela Accelerator. Pada isian Key Stroke isikan tombol F1. Sedangkan untuk `menuExit` isikan tombol Esc.

Setelah itu, untuk menambahkan baris perintah yang dijalankan saat MenuItem dipilih, klik kanan pada `menuAbout` kemudian pilih Events>Action>actionPerformed. Begitu juga untuk `menuExit`.

`menuAbout` digunakan untuk menampilkan sebuah dialogbox yang berisi tentang Aplikasi CRUD ini. Sedangkan `menuExit` digunakan untuk mengakhiri aplikasi.

Berikut baris perintah untuk kedua MenuItem tersebut:

```
private void menuAboutActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null, "Aplikasi CRUD Java\n"+
        "Entry Data Mahasiswa\n\n"+
        "Pemrograman Berbasis Objek\n"+
        "Oleh: Wendy.S.Kom.,M.Sc.");
}

private void menuExitActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```