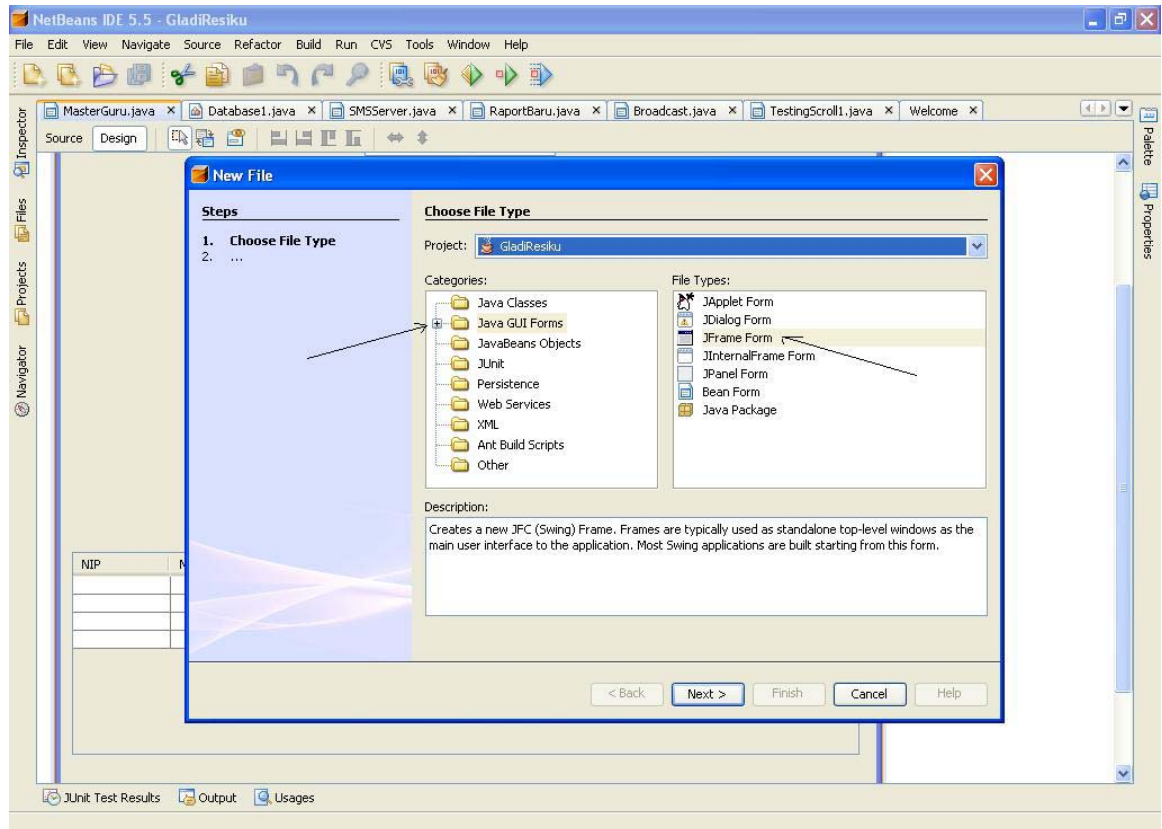
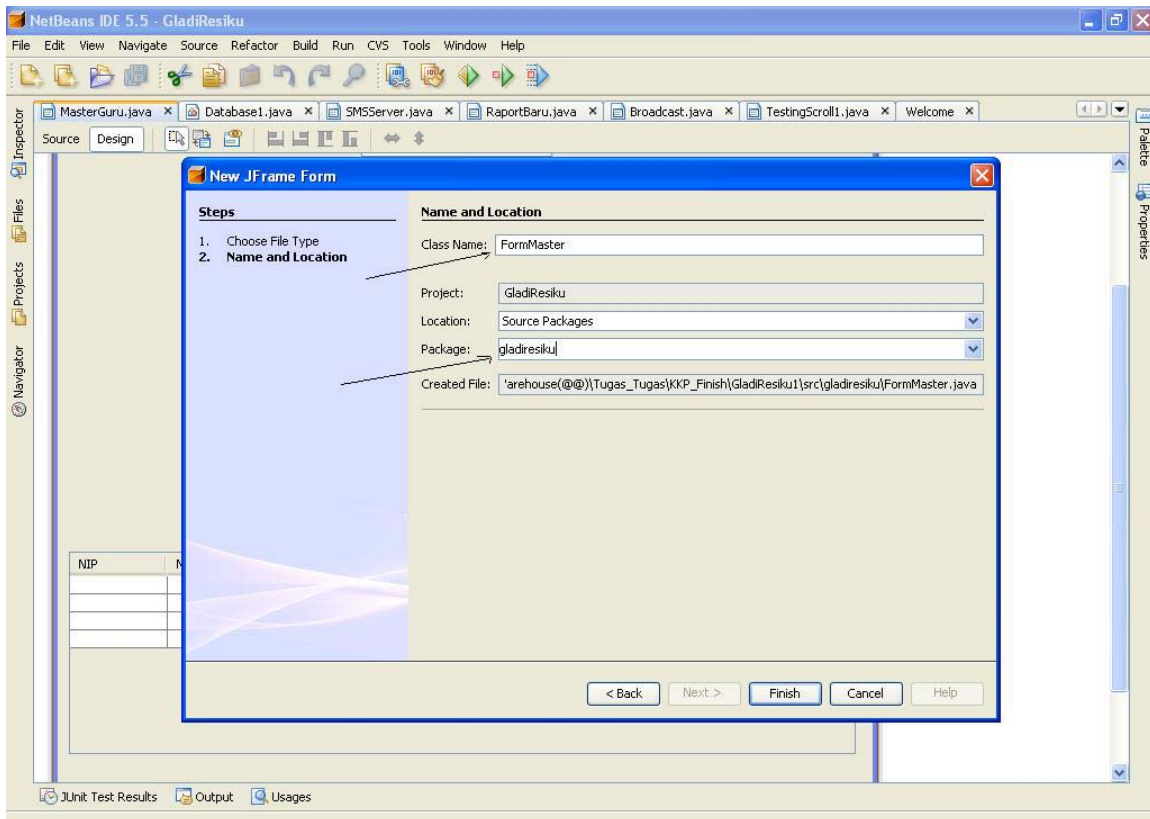


:: Tutorial Membuat Form dalam Netbeans ::

1. Sebelumnya kita install dulu netbeans5.5 atau versi di atasnya dengan jdk 1.5.0_7 atau versi di atasnya.
2. lalu kalau sudah terinstall, pilih File->New Project.
Disini kita akan bikin project dan otomatis akan terbentuk Main.java
3. klo sudah kita pilih File ->New File ada banyak tipe java disitu,kita pilih Java GUI Form -> sesuai anak panah.



4. lalu isi nama Classnya dan pilih packagenya sesuai anak panah.



5. kalau sudah pilih Finish.kita akan mulai membuat Form seperti pada tampilan di bawah ini

:: Tutorial Membuat Tabel ::

1. Pertama kita membuat Table dengan Mendrag JTable pada Palette. Kita beri nama dgn klik kanan pilih **"change variable name"**, kita beri nama misal : **"table"**
2. Pilih Table klik kanan Properties, seperti gambar dibawah ini :

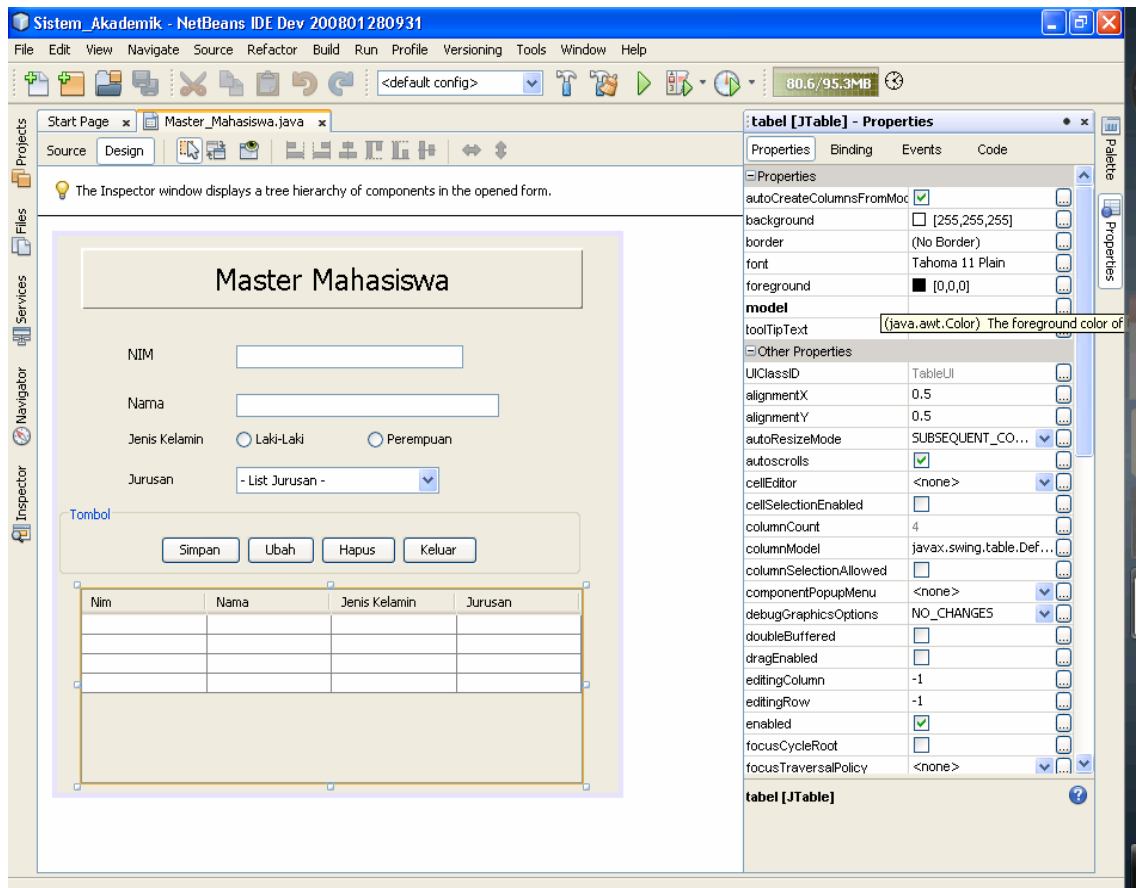
The screenshot shows a Java Swing window titled "Master Mahasiswa". It contains a form with the following fields:

- NIM:
- Nama:
- Jenis Kelamin: Laki-Laki Perempuan
- Jurusan:

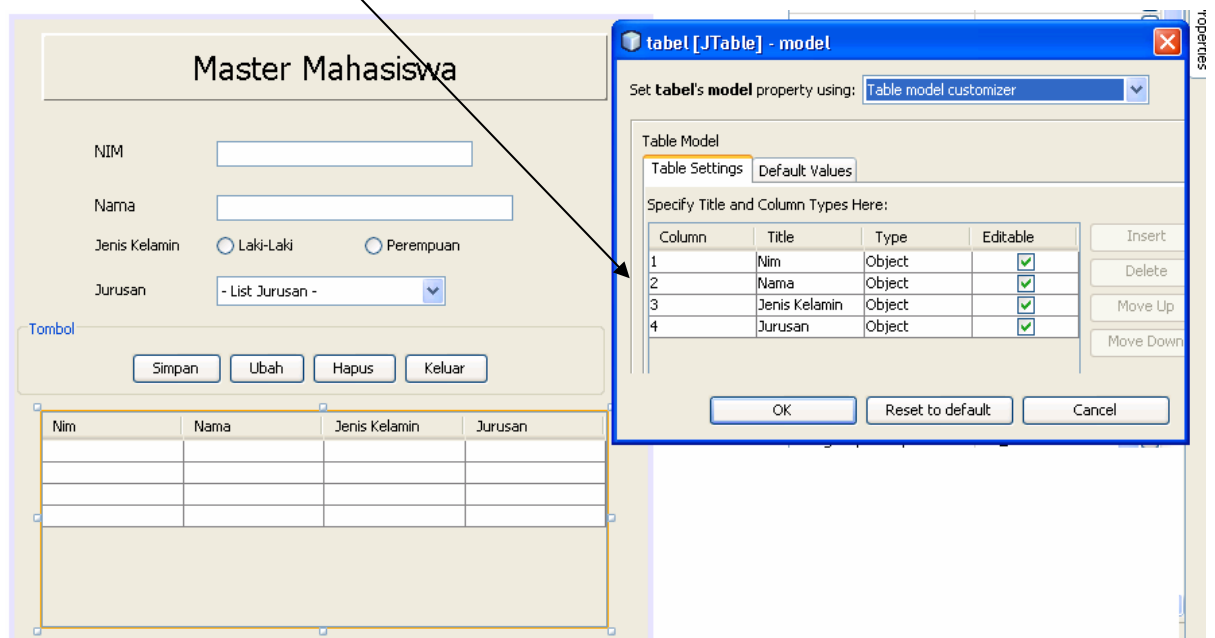
Below the form is a button bar labeled "Tombol" containing four buttons: "Simpan", "Ubah", "Hapus", and "Keluar".

At the bottom of the window is a table with the following columns: "Nim", "Nama", "Jenis Kelamin", and "Jurusan". The table has three empty rows. An arrow points to the right edge of the table, indicating the location of the JTable component in the IDE's Properties window.

*-Pilih
Inspector lalu
-pilih
JScrollPane
- pilih JTable.
Utk
Penggitan
Tablenya*



3. **kita** buat kolom dengan klik **Model lalu** isi nama kolom – kolomnya seperti gambar dibawah ini sesuai anak panah :



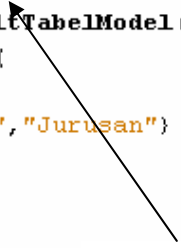
4. **Kalau** sudah klik OK tinggal kita setting di source code nya.
Pada baris paling akhir di sourcena sebelum } paling akhir kita bikin objek untuk TableModel langsung ketik coding dibawah ini :

```
private javax.swing.table.DefaultTableModel tblModel = getDefaultTabelModel();
```

5. **lalu** kita buat fungsi **getDefaultTabelModel(); dan Tabel** untuk pengaturan table nya seperti pada gambar dibawah ini, juga diketik secara manual:

```
private void Tabel(javax.swing.JTable tb,int lebar[]) {
    tb.setAutoResizeMode(tb.AUTO_RESIZE_OFF);
    int kolom=tb.getColumnCount();
    for(int i=0;i < kolom;i++) {
        javax.swing.table.TableColumn tbc = tb.getColumnModel().getColumn(i);
        tbc.setPreferredWidth(lebar[i]);
        tb.setRowHeight(17);
    }
}

private javax.swing.table.DefaultTableModel getDefaultTabelModel(){
    return new javax.swing.table.DefaultTableModel(
        new Object [][] {},
        new String [] {"Nim", "Nama", "Jenis Kelamin", "Jurusan"}
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, false , false
        };
        public boolean isCellEditable(int rowIndex, int
            columnIndex);
        return canEdit [columnIndex];
    }
};
}
```



Nama Kolom di Tabel

6. Setelah itu kita setting di **konstruktornya**, untuk pengaturan lebar kolomnya :
Tambahin source code berikut :

```
// kita modelkan tabelnya sesuai dgn objek tblModel  
tabel.setModel(tblModel);  
  
// kita kirim data ke fungsi Tabel :nama tabelnya,dan lebar tiap kolomnya  
Tabel(tabel,new int[] {120,180,120,120});
```

7. **Cara menginput data kedalam ke JTable :**

Pertama kali kita deklarasikan dulu variable array satu dimensi diluar fungsi supaya dikenal disemua fungsi yang lain.

```
String data[] = new String [4];
```

4 adalah jumlah kolom dalam tabel

Didalam fungsi `private void cmdsimpanactionperformed..`

Tambahkan coding seperti dibawah ini :

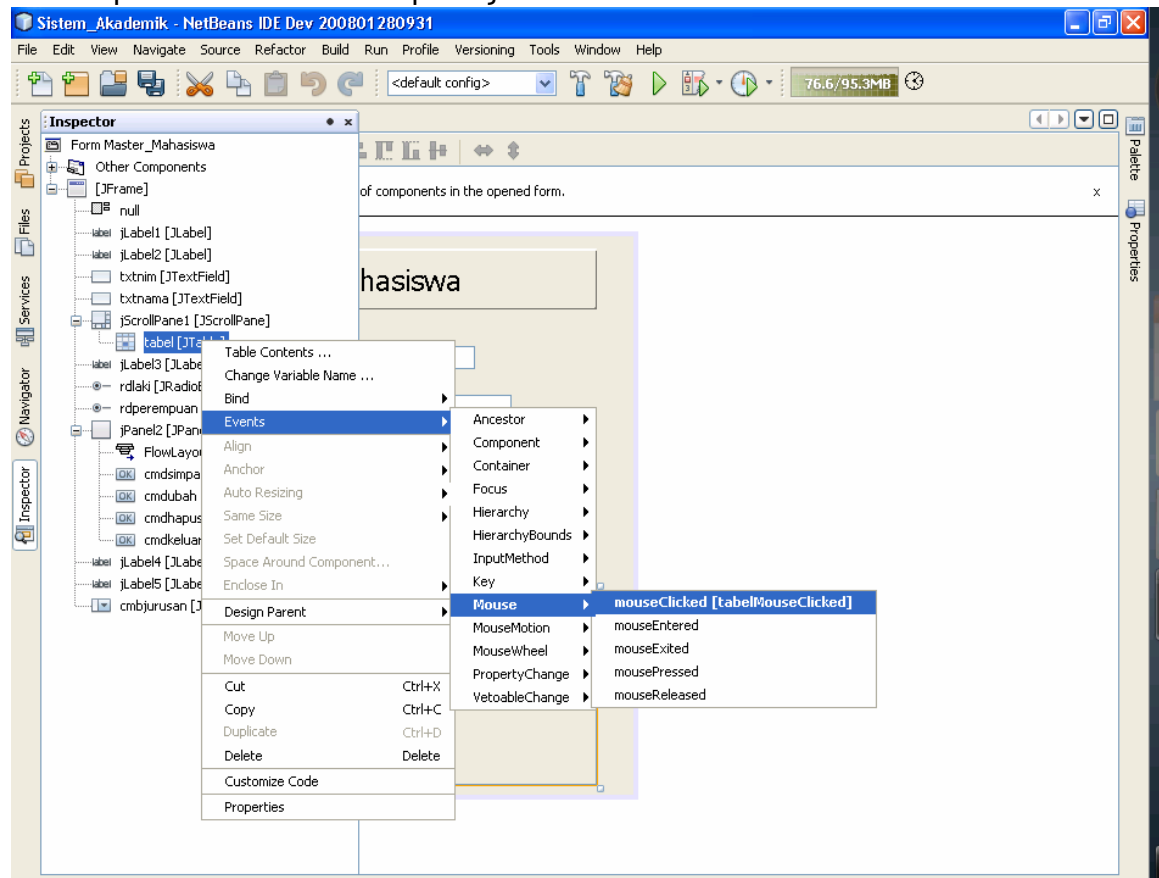
```
String jenkkel="";
if(rdlaki.isSelected())
    jenkkel="Laki-Laki";
else
    jenkkel="Perempuan";

data[0] = txtnim.getText();
data[1] = txtnama.getText();
data[2] = jenkkel;
data[3] = cmbjurusan.getSelectedItemAt().toString();
```

8. Cara Menampilkan Data dari Jtable kedalam Form di Netbeans:

Nanti kita akan mengklik 2x baris dalam JTable yang berisi data yang akan ditampilkan kedalam Form di Netbeans, caranya :

Pilih Inspector kemudian ikuti petunjuk dibawah ini :



Setelah itu akan muncul Fungsi :

```
private void tabelMouseClicked(java.awt.event.MouseEvent evt) {
}
}
```

Kita modifikasi Fungsi tersebut sehingga menjadi seperti dibawah ini :

```

private void tabelMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if(evt.getClickCount() == 2) {
        Tampil();
    }
}

int row = 0;
public void Tampil() {
    row = tabel.getSelectedRow();
    txtnim.setText(tblModel.getValueAt(row, 0).toString());
    txtnama.setText(tblModel.getValueAt(row, 1).toString());
    if(tblModel.getValueAt(row, 2).toString().equals("Laki-Laki")) {
        rdlaki.setSelected(true);
    } else {
        rdperempuan.setSelected(true);
    }
    cmbjurusan.setSelectedItem(tblModel.getValueAt(row, 3).toString());
}

```

Dengan coding diatas maka secara otomatis jika kita klik 2x baris dalam JTable maka data dalam baris terpilih akan muncul dalam Form kemudian bisa kita edit dan delete.

9. Cara Mengedit Data Dalam JTable

Setelah data muncul dalam Form untuk mengeditnya ikuti coding dibawah ini :

```

private void cmdubahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String jenkel="";
    if(rdlaki.isSelected())
        jenkel="Laki-Laki";
    else
        jenkel="Perempuan";

    data[0] = txtnim.getText();
    data[1] = txtnama.getText();
    data[2] = jenkel;
    data[3] = cmbjurusan.getSelectedItem().toString();
    tblModel.removeRow(row);
    tblModel.insertRow(row, data);
}

```

10. Cara Menghapus Data Dalam JTable

Ketikkan Coding Dibawah Ini :

```

private void cmdhapusActionPerformed(java.awt.event.ActionEvent evt) {
    tblModel.removeRow(row);
}

```

Belajar SQL(Structured Query Language)

1 Untuk belajar SQL bisa menggunakan MySQL Server di Dos untuk menjalankan Querynya atau dengan editor dari suatu MySQL yaitu MySQL Front,SQL Yog, dll.

2 SQL adalah standarisasi

SQL adalah sebuah ANSI (American National Standards Institute) yaitu standar bahasa komputer yang digunakan untuk mengakses dan memanipulasi sebuah database. Perintah SQL ini di gunakan untuk menemukan dan merubah data di dalam database. SQL bekerja dengan program database seperti MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase, dll.

Dalam banyak hal, terdapat perbedaan di dalam setiap versi dari Bahasa SQL ini, tetapi perbedaan tersebut tetap harus mengacu kepada standar ANSI. Yaitu perbedaan tersebut tidak menghilangkan perintah standar yang sudah ada sebelumnya seperti SELECT, UPDATE, DELETE, INSERT, WHERE, and others.

3 SQL sebagai tabel basis data

Sebuah database selalu terdiri dari satu atau lebih tabel. Sebuah tabel diidentifikasi dengan sebuah nama seperti "Customers" atau "Orders". Tabel terdiri dari record-record (rows) dengan data didalamnya.

Sebagai contoh dapat kita lihat tabel "Peserta" dibawah ini :

nm_dpn	nm_blg	Alamat	Kota
Parto	Listio	JL.Bungur 10	Ulujami
Sandi	Puspita	Jl.Merdeka IX	Pesanggrahan
Mira	Dahlia	Jl.Sabar 19	Pesanggrahan

Pada contoh di atas diperlihatkan satu buah table yang berisi tiga buah record dimana satu record memiliki empat buah kolom yang nantinya kita sebut sebagai Field yaitu : nm_dpn, nm_blg, alamat dan kota.

4 Komponen Sistem Basis Data

Ada beberapa komponen yang berurutan dalam Database :

1. Database
2. Tabel
3. Field (kolom)
4. Rows (Baris)
5. Data

Pembahasan SQL terdiri dari 2 bagian yaitu DML dan DDL.akan lebih dijelaskan dibawah ini

5 Data Manipulation Language (DML)

SQL (Structured Query Language) adalah sebuah syntax untuk melakukan eksekusi terhadap sebuah query. Tetapi bahasa SQL harus selalu ada syntax untuk melakukan update, insert, and delete dari pada sebuah records atau lebih.

Query dan perintah update ini bersamaan muncul dari **Data Manipulation Language (DML)** yang merupakan bagian dari SQL diantaranya :

- SELECT : mengelompokkan data terpilih dari sebuah tabel
- UPDATE : ubah/perbaiki dari sebuah tabel basis data
- DELETE : hapus data dari sebuah tabel basis data
- INSERT INTO : input data baru ke dalam sebuah tabel basis data

6 Data Definition Language (DDL)

Data Definition Language (DDL) juga merupakan bagian dari SQL yang berfungsi untuk untuk membuat dan menghapus tabel dari sebuah database. Kita juga dapat mendefinisikan indexes (keys) untuk mencirikan dan membuat hubungan antar tabel.

Yang terpenting dari perintah DDL didalam SQL adalah:

- CREATE TABLE : perintah untuk membuat tabel
- ALTER TABLE : perintah untuk mengubah tabel
- DROP TABLE : perintah untuk menghapus tabel
- CREATE INDEX : perintah untuk membuat index suatu tabel
- DROP INDEX : perintah untuk menghapus index dari suatu tabel

7 Perintah CREATE dalam SQL

CREATE digunakan untuk membuat suatu database/table dalam SQL.

> Syntax Create Database :

```
CREATE DATABASE nama_database;
```

Contoh :

Kita akan membuat database dalam SQL dengan nama system_akademik, dapat digunakan perintah CREATE seperti berikut :

```
CREATE DATABASE system_akademik;
```

> Syntax Create Tabel :

```
CREATE TABLE nama_tabel(  
    Nama_field1 tipe(panjang),  
    Nama_field2 tipe(panjang),  
    .  
    .  
    Nama_fieldn tipe(panjang),  
    Primary Key(field_key)  
);
```

Contoh :

Sebelum membuat table kita harus memilih database yang akan kita pakai caranya ketik :
USE SYSTEM_AKADEMIK;

Setelah kita melakukan perintah diatas maka ada pesan :

```
DATABASE Changed
```

Kita akan membuat database dalam SQL dengan nama system_akademik, dapat digunakan perintah CREATE seperti berikut :

```
CREATE TABLE MAHASISWA(
    Nim varchar(10),
    Nama varchar(60),
    Jenkel varchar(20),
    Jurusan varchar(30),
    PRIMARY KEY(NIM)
);
```

8 Perintah INSERT INTO

Perintah INSERT INTO digunakan untuk menambahkan data ke baris baru dalam suatu tabel.

Syntax :

```
INSERT INTO table_name VALUES (value1, value2, ..... )
```

Kita juga dapat menspesifikasikan kolom yang akan dimasukkan data terbaru

```
INSERT INTO table_name (column1, column2, .....)
VALUES (value1, value2, .....)
```

Contoh Input data ke baris baru

Tabel Peserta

nim	nama	Jenkel	Jurusan
0411500349	Pradana	Laki-Laki	Tekhnik Informatika

Perintah SQL :

```
INSERT INTO mahasiswa VALUES ('0411500349', 'Pradana', 'Laki-Laki', 'Tekhnik Informatika')
```

Contoh Input Data ke kolom spesifik

Tabel Peserta

nim	nama	Jenkel	Jurusan
0411500349	Pradana	Laki-Laki	Tekhnik Informatika

Perintah SQL

```
INSERT INTO mahasiswa (nim,Jurusan) VALUES ('0411500258', 'Sistem Informasi')
```

Hasil :

nim	nama	Jenkel	Jurusan
0411500349	Pradana	Laki-Laki	Tekhnik Informatika
0411500258			Sistem Informasi

9 Perintah SELECT dalam SQL

SELECT statemen digunakan untuk memilih data dari suatu tabel. hasil dalam bentuk tabel yang disimpan di dalam suatu tabel hasil yang disebut result-set

Syntax :

```
SELECT column_name(s) FROM table_name
```

Contoh :

Memilih beberapa kolom dengan nama "nim" dan "nama", dapat digunakan perintah SELECT seperti berikut :

SELECT nim, nama from mahasiswa

Tabel Peserta

nim	nama	Jenkel	Jurusan
0411500349	Pradana	Laki-Laki	Tekhnik Informatika
0411500258			Sistem Informasi

Hasil :

nim	nama
0411500349	Parto
0411500258	

A. Memilih Seluruh Kolom

Memilih semua kolom pada table Peserta, dapat digunakan tanda *, seperti contoh berikut :

SELECT * FROM mahasiswa

Hasil :

nim	nama	Jenkel	Jurusan
0411500349	Pradana	Laki-Laki	Tekhnik Informatika
0411500258			Sistem Informasi

B. Klausu Where pada SQL

Klausu Where digunakan untuk memilih data yang lebih spesifik dengan sebuah persyaratan atau lebih

Syntax :

SELECT column FROM table WHERE column operator value

Terdapat beberapa operator pada klausu WHERE yang bisa digunakan antara lain :

Operator	Deskripsi
=	Sama dengan
<>	Tidak sama dengan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar dari atau sama dengan
<=	Lebih kecil dari atau sama dengan
BETWEEN	Diantara range inklusif
LIKE	Mencari sesuai pattern

Catatan : Dalam beberapa versi dari SQL operator <> ini dituliskan seperti !=

C. Menggunakan Klausu WHERE

Mencari data dengan memilih mahasiswa yang Jurusannya "Sistem Informasi", kita dapat menambahkan WHERE clause dalam perintah SELECT sebagai berikut :

SELECT * FROM Mahasiswa WHERE Jurusan='Sistem Informasi'

Tabel Peserta

nim	nama	Jenkel	Jurusan
0411500349	Pradana	Laki-Laki	Tekhnik Informatika
0411500258			Sistem Informasi

Hasil :

nim	nama	Jenkel	Jurusan
0411500258			Sistem Informasi

SQL menggunakan tanda kutip tunggal untuk data bertipe teks (namun adakalanya database lain memperbolehkan menggunakan tanda kutip ganda). Sedangkan pada data bertipe numerik tidak diperbolehkan menggunakan tanda kutip.

Contoh :

SELECT * FROM Mahasiswa where nsms='Pradana' ' pernyataan yang benar
 SELECT * FROM Mahasiswa where nsms="Pradana" ' pernyataan yang salah (tanpa kutip tunggal)

SELECT * FROM Persons WHERE Year>1965 ' pernyataan yang benar
 SELECT * FROM Persons WHERE Year>"1965" ' pernyataan yang salah

D. Kondisi LIKE

LIKE digunakan untuk memilih data dengan kondisi yang spesifik untuk pencarian dalam sebuah kolom pada tabel.

Syntax :

SELECT column FROM table WHERE column LIKE pattern

Symbol "%" yang digunakan untuk persyaratan yang akan dicari di awal ataudiakhir dari yang dicari.

Penggunaan klausa LIKE

Perintah berikut akan menghasilkan data peserta dengan nama depan yang diawali dengan 'M' :

SELECT * From Mahasiswa WHERE nama LIKE 'P%'

Perintah berikut akan menghasilkan Peserta dengan nama yang diakhiri dengan 'a' :

SELECT * From Mahasiswa WHERE nama LIKE '%a'

Perintah berikut akan menghasilkan Peserta dengan nama yang memiliki huruf "la" :

SELECT * From Mahasiswa WHERE nama LIKE '%la%'

10 Perintah UPDATE

Perintah UPDATE digunakan untuk mengubah data dalam suatu tabel.

Syntax :

UPDATE table_name SET column_name = new_value
 WHERE column_name=some value

Tabel Peserta :

Update field untuk sebuah record

Contoh :

UPDATE Mahasiswa SET nama='Joko', Jenkel='Laki-Laki' WHERE nim=' 0411500258'

Hasil :

nim	nama	Jenkel	Jurusan
0411500349	Pradana	Laki-Laki	Tekhnik Informatika
0411500258	Joko	Laki-Laki	Sistem Informasi

11 Perintah DELETE

Perintah delete digunakan untuk menghapus baris dari suatu tabel.

Syntax :

```
DELETE FROM table_name WHERE column_name=some value
```

Tabel Peserta :

nim	nama	Jenkel	Jurusan
0411500349	Pradana	Laki-Laki	Tekhnik Informatika
0411500258	Joko		Sistem Informasi

Menghapus sebuah baris

Contoh :

```
DELETE FROM Mahasiswa WHERE nim=' 0411500349'
```

Hasil :

nim	nama	Jenkel	Jurusan
0411500258	Joko	Laki-Laki	Sistem Informasi

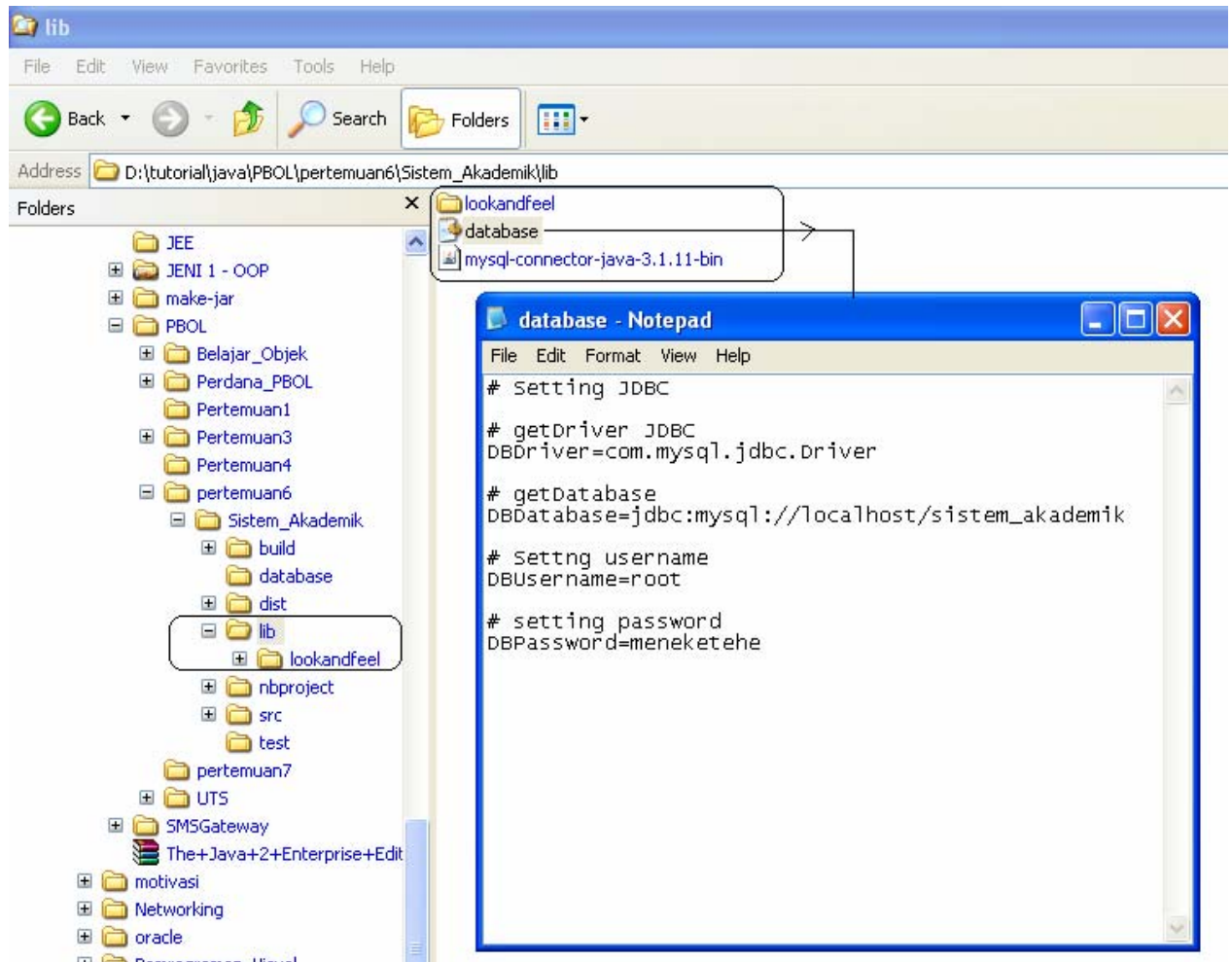
Menghapus semua baris

Contoh :

```
DELETE FROM table_name
```

Tutorial Koneksi Netbeans ke Mysql

1. Pertama kita buat file dengan extension .ini pada folder lib dengan terlebih dahulu membuat folder lib(misal : database.ini). yang misal isinya seperti dibawah ini :



2. Lalu kita buat file Database.java untuk melakukan relasi dengan file Database.ini yang isinya sbb:

```

package sistem_akademik;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Properties;
import javax.swing.JOptionPane;
/**
 *
 * @author Administrator
 */
public class Database {
    public Properties mypanel, myLanguage;
    private String strNamaPanel;
    /** Creates a new instance of Database */
    public Database() {
    }
    public String SettingPanel(String nmPanel){
        try{
            mypanel = new Properties();
            mypanel.load(new FileInputStream("lib/database.ini"));
            strNamaPanel = mypanel.getProperty(nmPanel);
        } catch(IOException e){
            JOptionPane.showMessageDialog(null, "Tidak ada koneksi", "Error", JOptionPane.INFORMATION_MESSAGE);
            System.err.println(e.getMessage());
            System.exit(0);
        }
        return strNamaPanel;
    }
}

```

Mengolah file database.ini

3. Kita tinggal memanggil class Database.java(gb diatas) jika kita ingin melakukan pemrosesan data yang berhubungan ke database. Untuk mengambil data dari file database.ini kita akan memodifikasi program seperti dibawah ini pada sebelum dan dalam Konstrktor :

```

package sistem_akademik;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
/**
 *
 * @author Administrator
 */
public class Master_Mahasiswa extends javax.swing.JFrame {

    /** Creates new form Master Mahasiswa */
    Database dbsetting;
    String driver, database, user, pass, userLogin;
    public Master_Mahasiswa() {
        initComponents();
        dbsetting = new Database();
        driver = dbsetting.SettingPanel("DBDriver");
        database = dbsetting.SettingPanel("DBDatabase");
        user = dbsetting.SettingPanel("DBUsername");
        pass = dbsetting.SettingPanel("DBPassword");
        this.setSize(600,700);
        tabel.setModel(tblModel);
        Tabel(tabel, new int[] {100,120,120,100});
        setDefaultTable();
        btngroup.add(rdlaki);
        btngroup.add(rdperempuan);
    }

```

Tambahan Import utk koneksi ke Mysql

Database dbsetting;
String driver, database, user, pass, userLogin;

Deklarasi Variabel utk membaca file ini

dbsetting = new Database();
driver = dbsetting.SettingPanel("DBDriver");
database = dbsetting.SettingPanel("DBDatabase");
user = dbsetting.SettingPanel("DBUsername");
pass = dbsetting.SettingPanel("DBPassword");

Coding utk membaca file .ini

4. Lalu kita buat Function setDefaultTable(); yang isinya sbb:

```

public void setDefaultTable() {
    String stat = "";
    try {
        Class.forName(driver);
        Connection kon = DriverManager.getConnection(database, user, pass);
        Statement stt = kon.createStatement();
        String SQL = "SELECT * FROM mahasiswa ";
        ResultSet res = stt.executeQuery(SQL);
        while(res.next()) {
            data[0] = res.getString(1);
            data[1] = res.getString(2);
            data[2] = res.getString(3);
            data[3] = res.getString(4);
            tblModel.addRow(data);
        }
        res.close();
        stt.close();
        kon.close();
    } catch(Exception exc) {
        System.err.println(exc.getMessage());
    }
}

```

Nama Database

Userke Mysql

Password ke MySQL

"SELECT * FROM mahasiswa "

Query SELECT

Coding utk menjalankan Query SELECT

Menampilkan data dari MySQL

5. Setelah coding diatas selesai. Maka data dari mysql akan tertampil pada table di Form javanya seperti gambar dibawah ini:

Master Mahasiswa

NIM

Nama

Jenis Kelamin Laki-Laki Perempuan

Jurusan

Tombol

Nim	Nama	Jenis Kelamin	Jurusan
0611500247	popo	Laki-Laki	TI
0611500248	Era	Laki-Laki	SI
0611500249	Reni	Perempuan	SK
0611500250	Widia	Perempuan	SI

Data dari MySQL

Pemrosesan Data Dengan Java dan Mysql

1. Menu Input :

Untuk pembuatan menu input . klik kanan tombol "Input" lalu pilih Event ->action->Action Performed

Setelah itu buatlah coding dalam function yg baru terbentuk seperti dibawah ini :

```
private void cmdsimpanActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String jenkel="";  
    if(rdlaki.isSelected())  
        jenkel="Laki-Laki";  
    else  
        jenkel="Perempuan";  
    try {  
        Class.forName(driver);  
        Connection kon = DriverManager.getConnection(database, user, pass);  
        Statement stt = kon.createStatement();  
        String SQL = "insert into mahasiswa values('"+txtnim.getText()+"',"+  
            "'"+txtnama.getText()+"', '"+data+"'";  
        stt.executeUpdate(SQL);  
        data[0] = txtnim.getText();  
        data[1] = txtnama.getText();  
        data[2] = jenkel;  
        data[3] = cmbjurusan.getSelectedItem().toString();  
        tblModel.insertRow(0, data);  
        stt.close();  
        kon.close();  
    } catch(Exception exc) {  
        System.err.println(exc.getMessage());  
    }  
}
```

Coding utk menjalankan Query INSERT

Query utk input data ke MYSQL

2. Menu Cari

Untuk pembuatan menu cari . klik kanan tabel di Netbeans lalu pilih Event ->Mouse->Mouse Click.

Kita menggunakan metode "klik",jika tablenya diklik maka data dari table akan tertampil di Formnya.Lalu tambahkan coding dalam Fungsi tabelMouseClicked seperti coding dibawah ini :

```
private void tabelMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    if(evt.getClickCount()==2) { event jika diklik  
        Tampil(); 2x  
    }  
}  
  
int row = 0;  
public void Tampil() {  
    row = tabel.getSelectedRow();  
    txtnim.setText(tblModel.getValueAt(row,0).toString());  
    txtnama.setText(tblModel.getValueAt(row,1).toString());  
    if(tblModel.getValueAt(row, 2).toString().equals("Laki-Laki")){  
        rdlaki.setSelected(true);  
    }else{  
        rdperempuan.setSelected(true);  
    }  
    cmbjurusan.setSelectedItem(tblModel.getValueAt(row,3).toString());  
}
```

3. Menu Edit

Untuk pembuatan menu edit . klik kanan tombol "ubah" di Netbeans lalu pilih Event ->Action->ActionPerformed.

Data setelah tertampil di Form maka baru kita edit :
Untuk lebih jelasnya lihat coding sbb:

```
private void cmdubahActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        String jenkel="";  
        if(rdlaki.isSelected())  
            jenkel="Laki-Laki";  
        else  
            jenkel="Perempuan";  
        Class.forName(driver);  
        Connection kon = DriverManager.getConnection(database, user, pass);  
        Statement stt = kon.createStatement();  
        String SQL = "update mahasiswa set nama='"+txtnama.getText()+"', " +  
                    | "Jenkel='"+jenkel+"' where nim='"+txtnim.getText()+"'";  
        stt.executeUpdate(SQL);  
        data[0] = txtnim.getText();  
        data[1] = txtnama.getText();  
        data[2] = jenkel;  
        data[3] = cmbjurusan.getSelectedItem().toString();  
        tblModel.removeRow(row);  
        tblModel.insertRow(row, data);  
        stt.close();  
        kon.close();  
    } catch(Exception exc) {  
        System.err.println(exc.getMessage());  
    }  
}
```

4. Menu Delete

Untuk pembuatan menu Hapus . klik kanan tombol "Hapus" di Netbeans lalu pilih Event ->Action->ActionPerformed.

Setelah data dicari, Menu yang terakhir adalah delete (untuk menghapus data):

```
private void cmdhapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        Class.forName(driver);  
        Connection kon = DriverManager.getConnection(database, user, pass);  
        Statement stt = kon.createStatement();  
        String SQL = "delete from mahasiswa where nim='"+txtnim.getText()+"'";  
        stt.executeUpdate(SQL);  
        tblModel.removeRow(row);  
  
        stt.close();  
        kon.close();  
    } catch(Exception exc) {  
        System.err.println(exc.getMessage());  
    }  
}
```

_____ :: Semoga Bermanfaat :: _____

Created by indra

indrasela@yahoo.com or indrasela@gmail.com

web blog : <http://indrasela.blogspot.com>