

MODUL
PRAKTKUM BAHASA PEMROGRAMAN II (BAHASA JAVA)
MEMBUAT APLIKASI SEDERHANA MENGGUNAKAN JAVA



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN
KOMPUTER PRABUMULIH**

2016

KATA PENGANTAR

Puji dan syukur kami panjatkan ke Hadirat Tuhan Yang Maha Esa. Karena berkat dan rahmat dan karunia-Nya sehingga kami dapat menyusun modul ini dengan baik serta tepat pada waktu-nya. Dalam makalah ini kami akan membahas mengenai “*Membuat aplikasi sederhana menggunakan java*”.

Modul ini disusun oleh penyusun dengan berbagai rintangan. Baik itu yang datang dari diri penyusun maupun yang datang dari luar. Namun dengan penuh kesabaran dan terutama pertolongan dari Tuhan akhirnya makalah ini dapat terselesaikan.

Harapan kami semoga modul ini membantu menambah pengetahuan dan pengalaman kita mengenai “*Membuat aplikasi sederhana menggunakan java*”. Modul ini kami akui masih banyak kekurangan karena pengalaman yang kami miliki cukup kurang. Oleh karena itu, kami harapkan kepada pembaca untuk memberi masukan-masukan yang bersifat membangun untuk kesempurnaan modul ini.

Semoga modul sederhana ini dapat dipahami bagi siapapun yang membacanya. Sekiranya laporan yang telah disusun ini dapat berguna bagi kami sendiri maupun orang yang membacanya. Sebelumnya kami mohon maaf apabila terdapat kesalahan kata-kata yang kurang berkenan dan kami mohon kritik dan saran yang membangun demi perbaikan dimasa depan.

Prabumulih, Maret 2016

Penyusun

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG.....	1
1.2 MAKSUD DAN TUJUAN.....	1
1.3 BATASAN MASALAH.....	2
1.4 SISTEMATIKA PENULISAN MODUL.....	2
BAB 2 LANDASAN TEORI.....	3
2.1 PENGERTIAN DAN DEFINISI JAVA.....	3
2.1.1 KELEBIHAN JAVA.....	3
2.1.2 KEKURANGAN JAVA.....	4
2.2 PENGERTIAN NETBEANS IDE.....	4
BAB 3 IMPLEMENTASI.....	5
3.1 MEMBUAT APLIKASI MENGGUNAKAN JAVA.....	5
BAB 4 PENUTUP.....	31
4.1 KESIMPULAN.....	31
4.2 SARAN.....	31

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era globalisasi saat ini perkembangan teknologi sangatlah maju dengan pesat dan cepat. Hal seperti ini telah membuat setiap individu terus berusaha dengan berbagai cara agar dapat mengikuti setiap perkembangan teknologi yang ada, sehingga pengetahuan mereka semakin bertambah dan maju seiring perkembangan zaman. Komputer, merupakan sebuah perkembangan teknologi yang sering dimanfaatkan untuk mencari berbagai macam informasi yang dibutuhkan oleh manusia. Bahkan sekarang ini komputer secara tidak langsung telah menjadi suatu unsur yang sangat penting dan sangat berguna bagi kehidupan manusia. Oleh karena itu teknologi komputer sudah dapat mengatasi berbagai masalah dalam pekerjaan manusia, termasuk menciptakan aplikasi yang dapat berguna untuk mempermudah manusia untuk memanfaatkan waktu dengan lebih baik.

1.2 Maksud dan Tujuan

Merujuk pada latar belakang dan rumusan masalah di atas, maka maksud dan tujuan yang kami hendak capai dalam penciptaan aplikasi sederhana ini adalah:

- 1) Mengimplementasikan materi java yang telah dipelajari.
- 2) Memberikan pemahaman lebih tentang komponen-komponen serta struktur pemrograman java.
- 3) Untuk mengasah kemampuan dalam bidang aplikasi Java.
- 4) Dapat mengenal struktur pemrograman pemrograman java.

1.3 Batasan Masalah

Dari permasalahan yang ada, aplikasi ini hanya akan menangani masalah-masalah sebagai berikut:

- 1) Aplikasi sederhana ini hanya dibuat sederhana dengan menggunakan pemrograman java.

1.4 Sistematika Penulisan Modul

Laporan tugas ini dibuat dengan sistem penulisan sebagai berikut:

BAB I PENDAHULUAN

Berisi : Latar Belakang, Maksud dan Tujuan, Batasan Masalah, Sistematika Penulisan Tugas.

BAB II LANDASAN TEORI

Berisi : penjelasan mengenai bahasa pemrograman php , html serta mysql.

BAB III IMPLEMENTASI

Berisi : bentuk real dari aplikasi yang dibuat.

BAB IV PENUTUP

Berisi : kesimpulan dan saran.

BAB II

LANDASAN TEORI

1.1 Pengertian dan Devinisi Java

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Kita lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi Java memiliki tiga komponen penting, yaitu:

- 1) Programming
- 2) Language specification
- 3) Application-programming interface
- 4) Virtual-machine specification

2.1.1 Kelebihan Java

- 1) Berorientasi objek, Memudahkan untuk mendesign dan mengembangkan program dengan cepat dan teliti, sehingga mudah digunakan. Salah satu bahasa pemrograman yang berorientasi objek secara murni.
- 2) Mirip c++, Mempunyai sintaks yang mirip dengan bahasa pemrograman c++ . Sehingga para pengguna c++ banyak yang hijrah menggunakan Java.
- 3) Multiplatform, Dapat digunakan dibanyak sistem operasi.
- 4) Perkembangan yang luas, Dari game sampai sistem operasi handphone menggunakan program java . Misalnya Handphone Sonny Ericsson dan Opera Mini 3 yang bertipe .jar (Java Archive).
- 5) Mempunyai pengumpulan sampah otomatis.

2.1.2 Kekurangan Java

- 1) Proses Compile, Mengharuskan pengguna mengcompile programnya sebelum dijalankan, berbeda dengan bahasa pemrograman python yang tidak perlu mengcompile terlebih dahulu.
- 2) Penggunaan Memori yang besar, Berbeda dengan bahasa pemrograman lain yang hanya membutuhkan memori sedikit.

2.1 Definisi Netbeans

NetBeans IDE adalah IDE open source yang ditulis sepenuhnya dengan bahasa Java menggunakan platform NetBeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi java. Semua fungsi IDE disediakan oleh modul-modul. Tiap modul menyediakan fungsi yang didefinisikan dengan baik, seperti dukungan untuk bahasa pemrograman Java, editing, atau dukungan bagi CVS. NetBeans memuat semua modul yang diperlukan dalam pengembangan java dalam sekali download, memungkinkan pengguna untuk mulai bekerja sesegera mungkin. Modul-modul juga mengizinkan NetBeans untuk dikembangkan. Fitur-fitur baru, seperti dukungan bahasa pemrograman lain, dapat ditambahkan dengan menginstal modul tambahan. Sebagai contoh, Sun Studio, Sun Java Studio Enterprise, dan Sun Java Studio Creator dari Sun Microsystem semuanya berbasis NetBeans IDE.

BAB III

IMPLEMENTASI

3.1 Membuat Aplikasi Menggunakan Java

Pada buku ini, kita akan mencoba membuat sebuah aplikasi sederhana menggunakan java yang menggunakan sistem basis data MySQL programnya sederhana, hanya membuat sebuah form dengan fasilitas CRUD (Create, Read, Update, Delete).

Tabel yang akan kita buat sekarang adalah tabel MAHASISWA, dimana tabel tersebut memiliki beberapa kolom, yaitu NIM, NAMA, TANGGAL LAHIR, JURUSAN DAN ALAMAT. Langkah-langkah dalam pembuatan aplikasi tersebut sebagai berikut:

1) Membuat Database

Sebelum membuat program sederhana ini, hal yang pertama perlu kita buat adalah database yang akan kita gunakan. Misal kita akan membuat database UNIVERSITAS maka kita bisa menggunakan perintah:

```
CREATE DATABASE UNIVERSITAS;
```

2) Membuat Tabel

Setelah membuat database, kita terlebih dahulu perlu membuat tabel MAHASISWA. Kita dapat menggunakan perintah sebagai berikut untuk membuat tabel MAHASISWA:

```
CREATE TABLE MAHASISWA(  
    NIM VARCHAR(8) PRIMARY KEY,  
    NAMA VARCHAR(50) NOT NULL,  
    TANGGAL_LAHIR DATE NOT NULL,  
    JURUSAN VARCHAR(50) NOT NULL,  
    ALAMAT VARCHAR(500) NOT NULL,  
);
```

Sekarang kita sudah punya sebuah tabel dengan nama MAHASISWA. Saatnya kita lanjutkan membuat project Java-nya menggunakan NetBeans IDE.

3) Membuat Project

Sama seperti sebelumnya, untuk membuat suatu project dalam NetBeans IDE kita dapat membuatnya menggunakan menu File → New Project. Setelah itu pilih kategori Java dan pilih tipe projectnya Java Application.



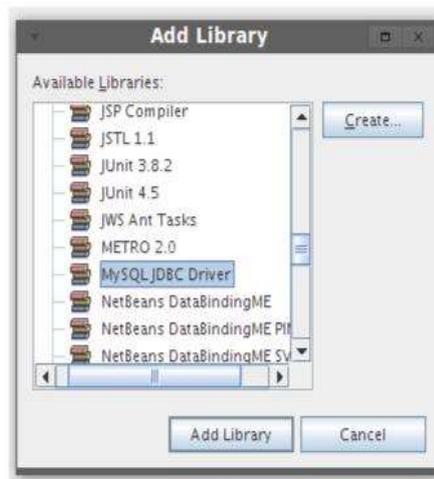
Klik next untuk melanjutkan pembuatan project.



Beri nama project dan jangan di ceklis checkbox Create Main Class. Hal ini dikarenakan kita tidak memerlukan dahulu membuat sebuah Main Class. Setelah itu klik tombol Finish sekarang kita telah membuat project Java menggunakan NetBeans IDE.

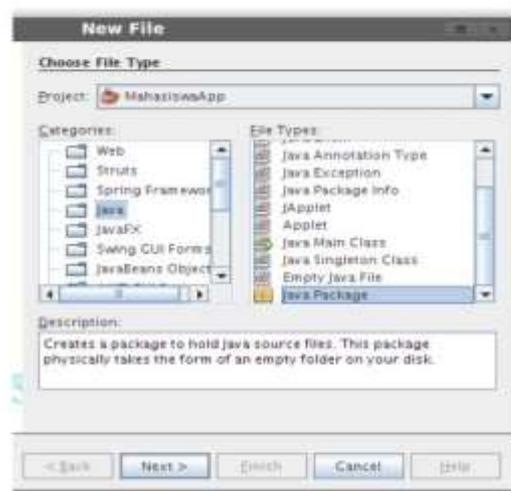
4) Membuat Koneksi MySQL

Setelah membuat project, saatnya membuat koneksi ke database UNIVERSITAS yang telah kita buat sebelumnya. Jadi hal yang pertama kita lakukan adalah menambah driver MySQL ke dalam project yang telah kita buat. Caranya klik kanan bagian Libraries project yang telah kita buat lalu pilih Add Library



Setelah keluar dialog Add Librerries, pilih MySQL JDBC Driver lalu klik tombol Add Librery untuk menambahkan MySQL Driver kedalam project.

Setelah menambah driver MySQL, sekarang saatnya membuat sebuah kelas yang akan kita gunakan untuk melakukan koneksi ke database MySQL. Tapi sebelum membuat sebuah kelas, pastikan kita membuat package dulu, caranya klik kanan bagian Source project yang telah kita buat lalu pilih New → Other.

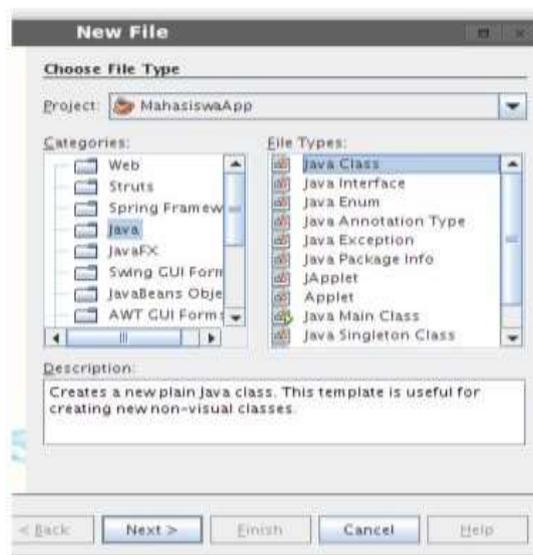


Setelah keluar dialog New File, pilih kategori Java dan jenis file Java Package. Klik next untuk melanjutkan membuat package.



Setelah itu beri nama packagenya, misalnya **program.universitas as**, setelah itu klik Finish untuk membuat packagenya.

Setelah membuat package **program.universitas as**, sekarang kita buat sebuah kelas untuk melakukan koneksi ke MySQL. Caranya klik kanan package **program.universitas as** lalu pilih New → Other.



Pilih kategori Java dan tipe Filenya Java Class, setelah itu klik tombol Next untuk melanjutkan membuat sebuah kelas.



Beri nama kelas tersebut, misal KoneksiDatabase, setelah itu klik Finish agar kelas Koneksi Database terbuat.



Sekarang, saatnya melakukan proses pengkodean. Pertama buat sebuah variabel static yang bertipe Java.sql.Connection, kita menggunakan static agar nanti aplikasi dapat mengakses koneksi secara langsung tanpa harus membuat object KoneksiDatabase.

```
packge program.universitas;
import java.sql.Connection;
public class KoneksiDatabase {
    private static Connection koneksi;
}
```

Setelah itu buat sebuah metode static `getKoneksi()`, metode ini digunakan untuk mendapatkan koneksi itu sendiri.

```
package program.universitas;
import java.sql.Connection;
public class KoneksiDatabase {
    private static Connection koneksi;
    public static Connection getKoneksi() {
        return koneksi;
    }
}
```

Sekarang untuk mengambil koneksi kita dapat langsung menggunakan perintah `KoneksiDatabase.getKoneksi()`, namun pastinya jika kita melakukan hal itu akan terjadi error, kok bisa?

Tentu karena, pada kelas `KoneksiDatabase` tersebut kita belum membuat koneksinya, jadi sebelum return koneksi, pada metode `getKoneksi()` seharusnya kita cek dulu apakah koneksinya null, jika null, maka kita deklarasikan sebuah koneksi yang baru.

```
package program.universitas;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class KoneksiDatabase {
    private static Connection koneksi;
    public static Connection getKoneksi();
    // cek apakah koneksi null
    if (koneksi == null) {
        try {
            String url = "jdbc:mysql://khannedy.server:3306 /UNIVERSITAS";
            String user = "echo";
            String pssword = "xxxxx";
            DriverManager.registerDriver (new com.mysql.jdbc.Driver());
            Koneksi = DriverManager.getConnection(url, user, pssword);
        } catch (SQLException) {
            System.out.println("Error membuat Koneksi");
        }
    }
    return koneksi;
}
```

Sekarang kita telah selesai membuat sebuah kelas untuk melakukan proses koneksi ke MySQL saatnya kita membuat Form aplikasinya.

5) Membuat Form Aplikasi

Pada program sederhana ini, kita tidak akan membuat program berbasis terminal (command line) lagi, tapi kita akan membuat aplikasi berbasis GUI. Dalam java teknologi untuk membuat program berbasis GUI disebut Java Swing. Sekarang untuk membuat sebuah Form, kita harus membuat JFrame, caranya dengan klik kanan package **program.universitas**, lalu pilih New → Other.

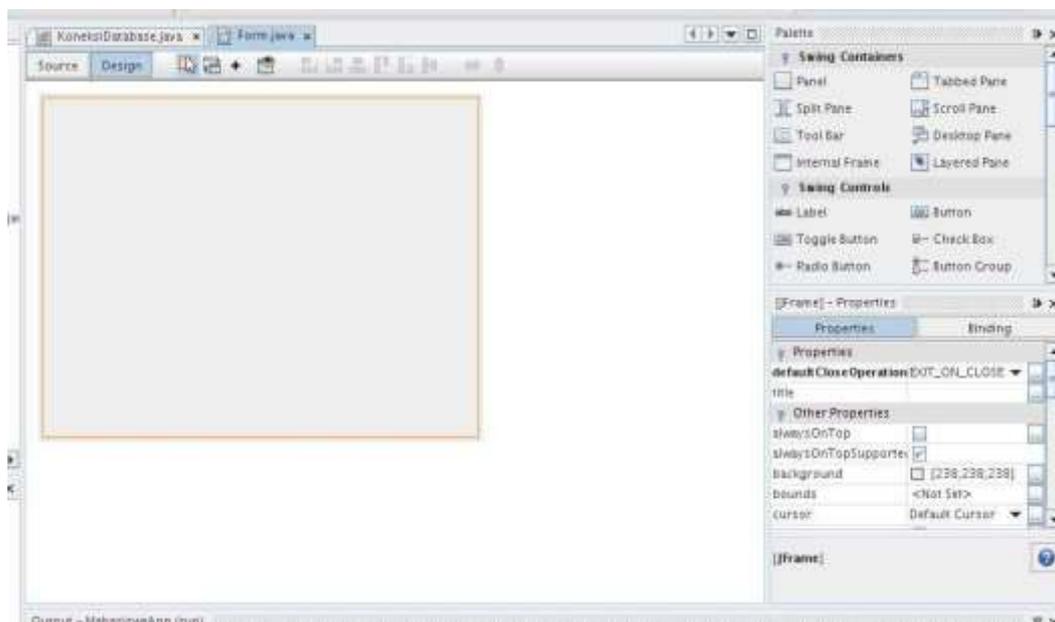


Pilih kategori Swing GUI Forms dan pilih tipe file JFrame Form. Lalu klik Next untuk melanjutkan membuat Form.

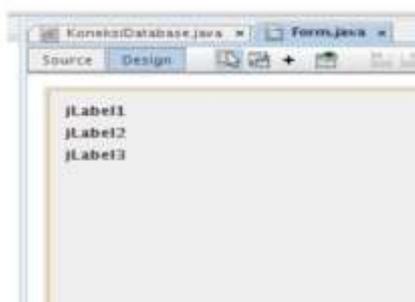
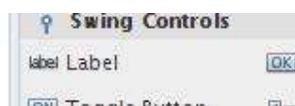


Beri nama Form tersebut, misal dengan nama Form, dengan begitu maka NetBeans akan membuat sebuah kelas dengan nama Form yang merupakan turunan dari kelas JFrame, dimana kelas JFrame ini merupakan kelas Java Swing.

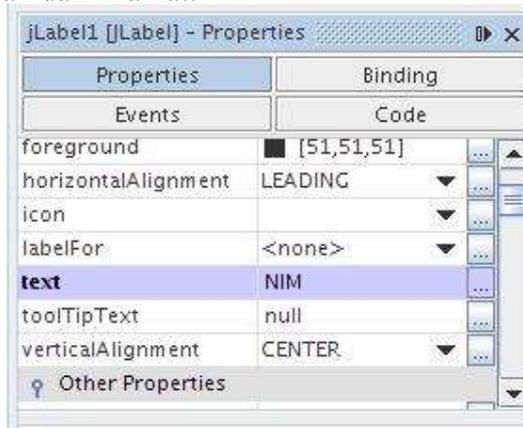
Sekarang kita dapat melihat GUI builder pada editor NetBeans dan disebelah kanannya terdapat Pallette yang merupakan komponen-komponen GUI yang ada di Java dan Properties yang merupakan editor atribut-atribut komponen yang kita klik pada GUI Builder.



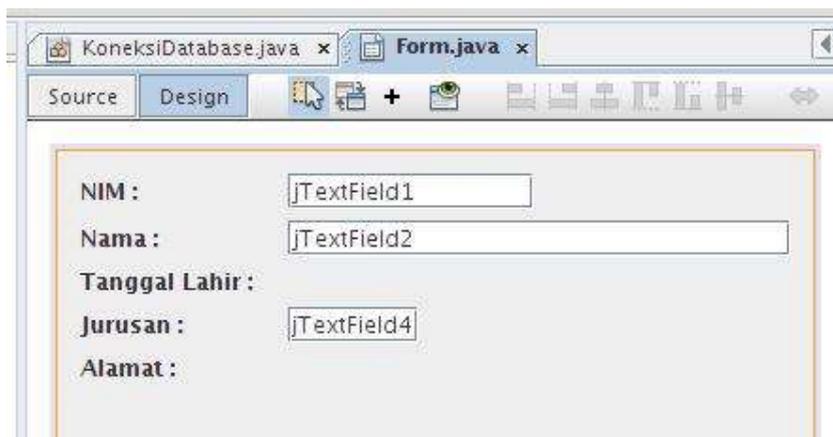
Untuk menambahkan komponen-komponen GUI lainnya, kita cukup mengklik dan mendrag salah satu komponen yang ada dalam Pallette ke dalam Form. Misal kita klik dan drag sebuah Label dari Pallette.



Untuk mengubah tulisan pada Label, kita dapat mengklik label tersebut, lalu lihat pada bagian Properties. Ubah atribut text, misal menjadi NIM, Nama, Tanggal Lahir, Jurusan dan Alamat.

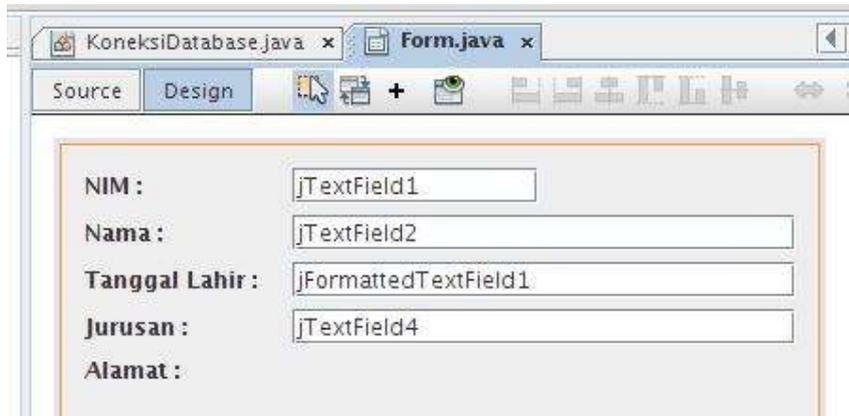


Setelah itu, klik dan drag tiga buah Text Field yang ada di Pallette ke Form, gunakan Text Field untuk Nim, Nama dan Jurusan.

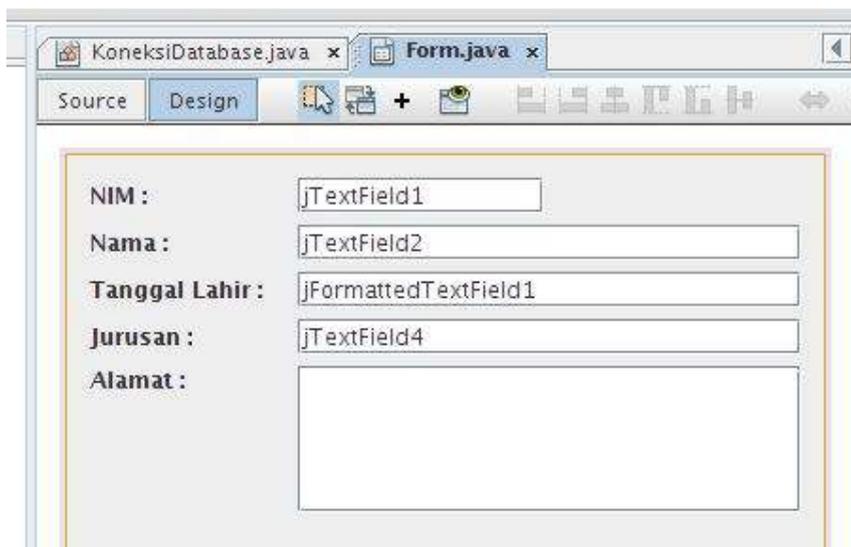


Untuk Tanggal Lahir dan Alamat kita tidak menggunakan Text Field, hal ini dikarenakan Tanggal Lahir memerlukan inputan berupa tanggal sedangkan Text Field hanya mendukung teks (string), sedangkan untuk Alamat, biasanya isi alamat itu panjang, sehingga lebih tidak cocok menggunakan Text Field, karena Text Field hanya mendukung satu baris.

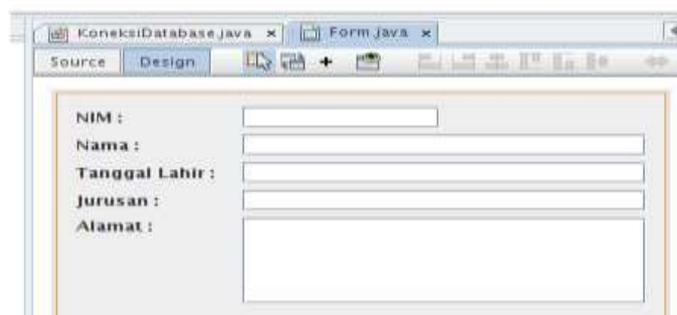
Dengan demikian, untuk Tanggal Lahir kita akan menggunakan Formatted Field, tinggal kita klik dan drag Formatted Field dari Pallette ke dalam Form.



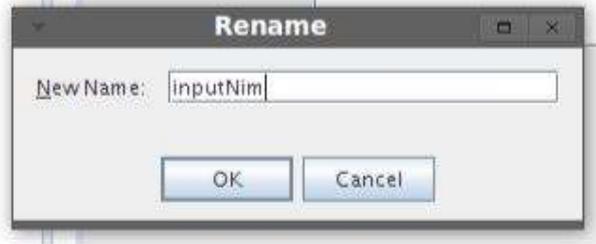
Dan untuk Alamat, gunakan komponen Text Area. Text Area hampir mirip dengan Text Field namun mendukung lebih dari satu baris.



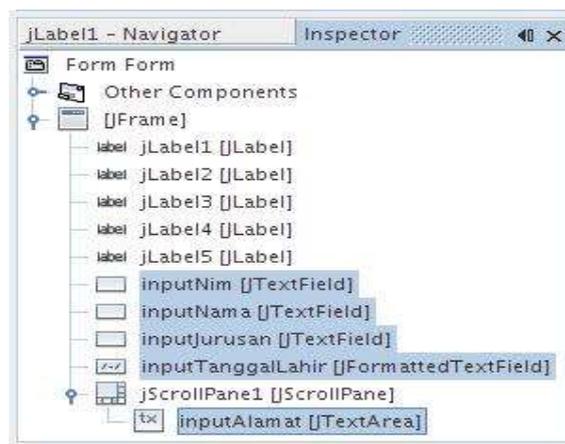
Untuk mengosongkan isi tulisan pada NIM, Nama, Tanggal Lahir, dan Jurusan, kosongkan atribut text pada setiap komponen pada Propertiesnya.



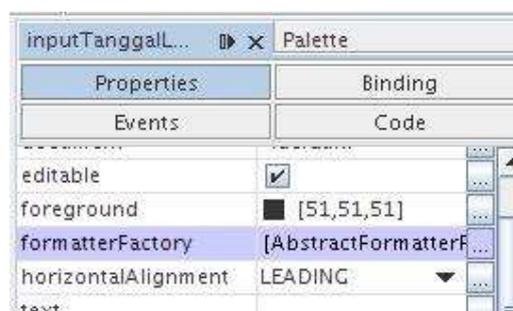
Setelah itu, sekarang saatnya kita mengubah setiap nama variabel komponennya, misal untuk Text Field NIM kita beri nama variabelnya dengan input Nim, untuk Text Field Nama dengan nama InputNama dan seterusnya, caranya dengan megklik kanan komponennya lalu pilih menu Change Variable Name.



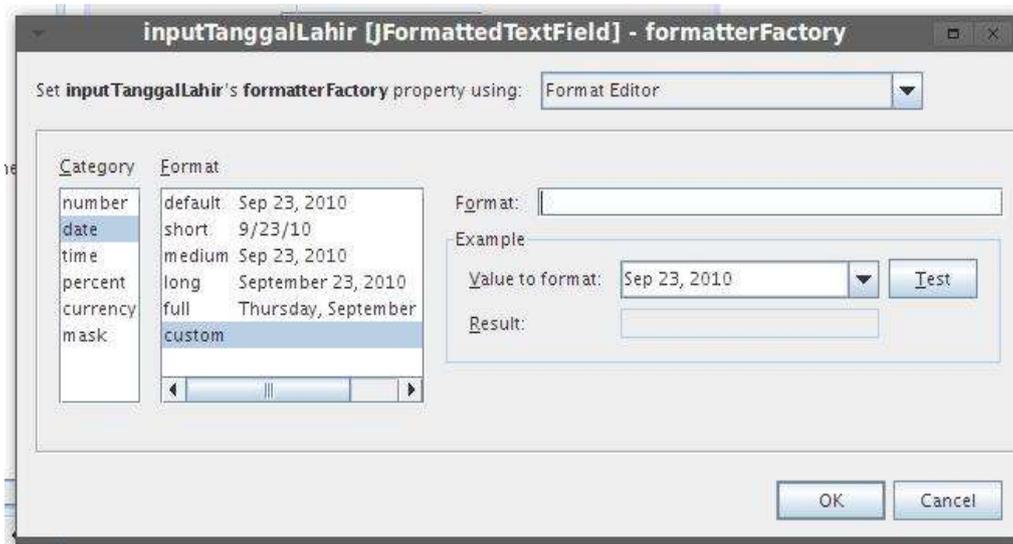
Untuk melihat seluruh nama variabelnya, kita dapat melihatnya pada bagian inspector di sebelah kiri bawah Form NetBeans.



Secara default Formatted Field seperti Text Field, dia hanya menerima teks (string), agar Formatted Field hanya menerima input berupa tanggal, maka kita perlu memberitahunya ke Formatted Fieldnya, caranya klik inputTanggalLahir, lalu pada bagian properties, cari atribut formatterFactory, ubah atribut tersebut.



Pada saat mengklik tombol [...] pada atribut formatterFactory, maka akan muncul dialog formatterFactory.



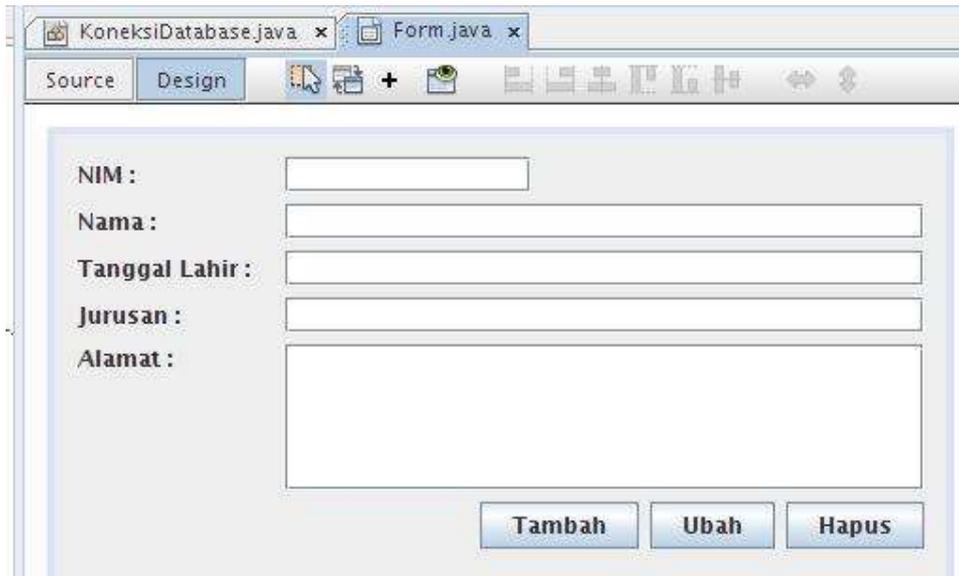
Agar Formatted Field hanya menerima input tanggal, maka ubah kategorinya menjadi date, formatnya menjadi custom, lalu pada input Format beri teks “d/M/yyyy”.



Maksud dari “d/M/yyyy” merupakan representasi tanggal/bulan/tahun dalam angka, misal jika tanggal 1 januari 2010, maka input harus 1/1/2010 dan seterusnya. Klik tombol Ok untuk mengkonfirmasi perubahan.

6) Menambah Tombol ke Form

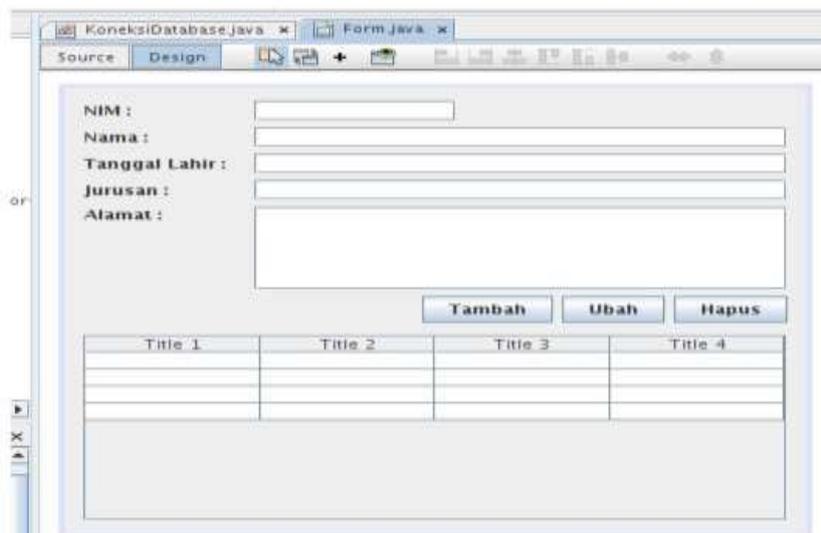
Setelah menambahkan input Form, sekarang saatnya kita menambah tombol ke dalam Form. Caranya dengan mengklik dan drag komponen Button pada Pallette ke dalam Form.



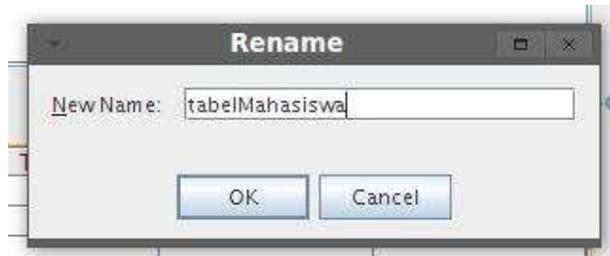
Tambahkan 3 buah tombol, Tambah, Ubah dan Hapus. Untuk mengubah teks tombolnya caranya sama seperti Label, yaitu dengan mengubah atribut text pada Properties.

7) Menambah Tabel ke Form

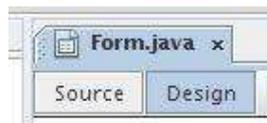
Setelah menambahkan input Form beserta tombolnya, sekarang saatnya menambahkan Tabel ke Form, caranya tinggal kita klik dan drag komponen Table dari Pallette ke Form hasilnya seperti terlihat pada gambar dibawah ini.



Jangan lupa untuk mengubah nama variabel Tabel yang tadi kita masukkan ke Form, caranya klik kanan Tabelnya lalu pilih Change Variable Name, misalnya dengan nama tabelMahasiswa.



Sekarang saatnya mengubah kolom pada tabel. Berbeda dengan komponen lain, untuk mengubah kolom pada komponen Tabel, kita memerlukan kelas lain, namanya kelas DefaultTableModel, sehingga kita perlu melakukan pengkodean, caranya masuk ke bagian source.



Setelah itu tambahkan sebuah variabel DefaultTableModel pada kelas Form tersebut.

```
1 package program.universitas;
2
3 import javax.swing.table.DefaultTableModel;
4
5 public class Form extends javax.swing.JFrame {
6
7     private DefaultTableModel model;
8
9     public Form() {
10         initComponents();
11
12         // membuat TableModel
13         model = new DefaultTableModel();
14
15         // menambahkan TableModel ke Tabel
16         tabelMahasiswa.setModel(model);
17     }
```

Untuk menambahkan kolom ke tabel, maka kita dapat menggunakan metode addColumn(nama) milik kelas DefaultTableModel. Dan saat ini kita perlu menambahkan kolom Nim, Nama, Tanggal lahir, Jurusan dan Alamat.

```
14  
15  
16 // menambahkan TableModel ke Tabel  
17 tabelMahasiswa.setModel(model);  
18  
19 model.addColumn("Nim");  
20 model.addColumn("Nama");  
21 model.addColumn("Tanggal Lahir");  
22 model.addColumn("Jurusan");  
23 model.addColumn("Alamat");  
24 }
```

8) Menambahkan Aksi

Sekarang kita telah selesai membuat Form, saatnya kita menambahkan aksi-aksi database, seperti load data dari database, menambah data ke database, mengubah data dari database dan menghapus data dari database.

9) Menambah Aksi Load Data

Saat pertama kali aplikasi muncul, maka otomatis kita harus mengambil seluruh data mahasiswa yang ada dalam tabel MAHASISWA dan ditampilkan ke dalam Table yang ada di Form. Dengan demikian, maka pertama kita perlu membuat sebuah aksi melakukan load data dari database.

Sekarang kita buat sebuah metode dengan nama loadData() dimana metode tersebut dibuat dalam kelas Form dan dalam metode tersebut berisikan proses load data dari database.

```
public void loadData() {  
  
}
```

Sebelum melakukan proses load data dari database, maka pertama kali, kita perlu menghapus seluruh isi baris yang ada pada Table yang ada di Form. Hal ini perlu dilakukan agar saat kita akan melakukan load ulang data. Maka Tabel dikosongkan dulu. Untuk mengosongkan isi Table, kita menggunakan DefaultTableModel.

```
public void loadData() {  
    // menghapus seluruh data  
    model.getDataVector().removeAllElements();  
    // memberi tahu bahwa data telah kosong  
    model.fireTableDataChanged();  
}
```

Setelah itu baru kita melakukan load datanya, untuk mengambil data dari database, kita memerlukan Connection yang ada dalam kelas KoneksiDatabase. Setelah itu buat Statement dan ResultSet seperti biasanya.

```
public void loadData() {
    // menghapus seluruh data
    model.getDataVector().removeAllElements();
    // memberi tahu bahwa data telah kosong
    model.fireTableDataChanged();
try{
    Connection C = KoneksiDatabase.getKoneksi();
    Statement S = C.createStatement();

    String sql = "SELECT * FROM MAHASISWA";
    ResultSet r = s.executeQuery(sql);

    While(r.next()) {
        // lakukan penelusuran baris
    }

    r.close();
    s.close();
} catch(SQLException) {
    System.out.println("Terjadi Error");
}
}
```

Pada saat melakukan proses penelusuran data menggunakan ResultSet, maka kita dapat menambahkan data tersebut ke dalam Table yang ada dalam Form. Untuk menambah sebuah baris ke Table kita menambakkannya ke DefaultTableModel dengan menggunakan metode `addRow(Object[])`.

```
While(r.next()) {
    // lakukan penelusuran baris
    Object[] o = new Object[5];
    o [0] = r.getString ("NIM");
    o [0] = r.getString ("NAMA");
    o [0] = r.getDate ("TANGGAL LAHIR");
    o [0] = r.getString("JURUSAN");
    o [0] = r.getString("ALAMAT");
    model.addRow(o);
}
```

Lengkapnya metode loadData() akan berisi seperti pada kode dibawa ini.

```
public void loadData(){
    // menghapus seluruh data
    model.getDataVector().removeAllElements();
    // memberi tahu bahwa data telah kosong
    model.fireTableDataChanged();

    try{
        Connection c = KoneksiDatabase.getKoneksi();
        Statement s = c.createStatement();

        String sql = "SELECT * FROM MAHASISWA";
        ResultSet r = s.executeQuery(sql);

        while(r.next()){
            // lakukan penelusuran baris
            Object[] o = new Object[5];
            o[0] = r.getString("NIM");
            o[1] = r.getString("NAMA");
            o[2] = r.getDate("TANGGAL_LAHIR");
            o[3] = r.getString("JURUSAN");
            o[4] = r.getString("ALAMAT");

            model.addRow(o);
        }

        r.close();
        s.close();
    }catch (SQLException e){
        System.out.println("Terjadi Error");
    }
}
```

Agar metode loadData() dipanggil ketika program berjalan, maka kita perlu memanggil metode loadData() dalam konsturktor Form.

```
24     model.addColumn("Tanggal");
25     model.addColumn("Jurusan");
26     model.addColumn("Alamat");
27
28     // panggil loadData()
29     loadData();
30 }
31
32 public void loadData() {
33     // menghapus seluruh dat
34     model.getDataVector().re
35     // memberi tahu bahwa da
36     model.fireTableDataChang
```

10) Menambah Aksi Tombol Tambah

Sekarang saatnya kita menambahkan aksi tombol, dimana aksi tombol itu akan berjalan ketika tombol tambah di klik. Untuk menambahkan sebuah aksi ke tombol tambah, pertama kita masuk ke bagian Design, setelah itu tinggal klik kanan tombol Tambahnya setelah itu pilih menu **Events** → **Action** → **actionPerformed**, maka otomatis NetBeans IDE akan membuatkan sebuah metode baru untuk aksi tombol Tambah.

```

197
198 private void jButton3ActionPerformed(java.awt.event
199     // TODO add your handling code here:
200 }
201

```

Dalam metode tersebutlah kita melakukan proses penambahan data ke dalam database. Untuk menambahkan data ke dalam tabel MAHASISWA, otomatis kita memerlukan data input dari pengguna. Untuk mendapatkan data tulisan dari Text Field dan Text Area, maka kita dapat menggunakan metode `getText()`, sedangkan untuk mendapatkan tanggal dari Formatted Field, kita dapat menggunakan metode `getValue()`, namun dikarenakan `getValue()` menghasilkan Object, maka kita perlu mengkonversinya ke tanggal.

```

String nim = inputNim.getText();
String nama = inputNama.getText();
java.util.Date tanggalLahir = (java.util.Date) inputTanggalLahir.getValue();
String jurusan = inputJurusan.getText();
String alamat = inputAlamat.getText();

```

Setelah mengambil seluruh data dari input, maka baru kita menyimpannya ke dalam database MySQL. Caranya adalah dengan membuat Connection dan kelas KoneksiDatabase setelah itu membuat PropertiesStatement untuk menyimpan datanya.

```

String nim = inputNim.getText();
String nama = inputNama.getText();
java.util.Date tanggalLahir = (java.util.Date) inputTanggalLahir.getValue();
String jurusan = inputJurusan.getText();
String alamat = inputAlamat.getText();

try{
    Connection c = KoneksiDatabase.getKoneksi();

    String sql = "INSERT INTO MAHASISWA VALUES (?, ?, ?, ?, ?)";

    PreparedStatement p = c.prepareStatement(sql);

    p.setString(1, nim);
    p.setString(2, nama);
    p.setDate(3, new java.sql.Date(tanggalLahir.getTime()));
    p.setString(4, jurusan);
    p.setString(5, alamat);

    p.executeUpdate();
    p.close();
}catch(SQLException e){
    System.out.println("Terjadi Error");
}finally{
    loadData();
}

```

Pada blok finally, kita perlu memanggil metode `loadData()`, hal ini dilakukan agar setelah proses penyimpanannya data ke database, maka data akan dimuat ulang ke Table yang ada di Form.

11) Menambah Aksi Tombol Ubah

Untuk aksi tombol ubah, agak sedikit berbeda dengan aksi tombol tambah, perbedaannya adalah pertama kita harus mendeteksi baris yang sedang di klik, setelah itu baru melakukan proses pengubahan data yang di klik dengan data baru yang ada dalam input Form.

Untuk menambah aksi tombol ubah caranya sama dengan tombol tambah, tinggal klik kanan tombol ubah lalu pilih **Events** → **Action** → **actionPerformed**.

```
235  
236 private void jButton2ActionPerformed(java.  
237     // TODO add your handling code here:  
238     }  
239
```

Setelah yang telah ditulis sebelumnya, pertama kita harus mendapatkan baris yang terseleksi pada Table, jika tidak ada baris yang terseleksi, maka proses Ubah dibatalkan. Untuk mendapatkan baris yang terseleksi kita dapat menggunakan metode `getSelectedRow()` milik Table, jika returnnya -1 artinya tidak ada baris yang terseleksi.

```
int i = tabelMahasiswa.getSelectedRow();  
if(i == -1){  
    // tidak ada baris terseleksi  
    return;  
}  
  
// ambil nim yang terseleksi  
String nim = (String) model.getValueAt(i, 0);  
  
String nama = inputNama.getText();  
java.util.Date tanggalLahir = (java.util.Date) inputTanggalLahir.getValue();  
String jurusan = inputJurusan.getText();  
String alamat = inputAlamat.getText();
```

Setelah mengambil data nim yang terseleksi data data lainnya dari input, baru kita lakukan proses ubah data yang ada di database berdasarkan nim yang baru yang terseleksi.

```
int i = tabelMahasiswa.getSelectedRow();  
if(i == -1){  
    // Tidak ada baris terseleksi  
    return;  
}  
  
// ambil nim yang terseleksi  
String nim = (String) model.getValueAt(i, 0);  
  
String nama = inputNama.getText();  
java.util.Date tanggalLahir = (java.util.Date) inputTanggalLahir.getValue();  
String jurusan = inputJurusan.getText();  
String alamat = inputAlamat.getText();  
  
try(  
    Connection c = KoneksiDatabase.getKoneksi();
```

```

String sql = "UPDATE MAHASISWA SET NAMA = ?, TANGGAL_LAHIR = ?, JURUSAN = ?,
ALAMAT = ? WHERE NIM = ?";

PreparedStatement p = c.prepareStatement(sql);

p.setString(1, nama);
p.setDate(2, new java.sql.Date(tanggalLahir.getTime()));
p.setString(3, jurusan);
p.setString(4, alamat);
p.setString(5, nim);

p.executeUpdate();
p.close();
}catch(SQLException e){
    System.out.println("Terjadi Error");
}finally{
    loadData();
}

```

12) Menambah Aksi Tombol Hapus

Untuk aksi hapus, kita tidak perlu menggunakan input Form, yang kita perlukan hanyalah baris yang terseleksi. Jika baris tidak ada yang terseleksi, maka proses penghapusan dibatalkan. Untuk menambah aksi pada tombol Hapus caranya sama seperti tombol Tambah dan Ubah, klik kanan tombol Hapus, lalu pilih menu **Events** → **Action** → **actionPerformed**.

```

278
279 private void jButton1ActionPerformed(java
280 // TODO: add your handling code here:
281 }
282

```

Setelah itu sama seperti proses Ubah, kita cek dulu apakah ada baris yang terseleksi atau tidak, jika ada ambil nim yang terseleksi, jika tidak ada maka batalkan proses tersebut.

```

int i = tabelMahasiswa.getSelectedRow();
if(i == -1){
    // tidak ada baris terseleksi
    return;
}

String nim = model.getValueAt(i, 0);

```

Setelah itu, baru kita lakukan proses penghapusan data dari database berdasarkan data baris yang terseleksi.

```
int i = tabelMahasiswa.getSelectedRow();
if(i == -1){
    // tidak ada baris terseleksi
    return;
}

String nim = (String) model.getValueAt(i, 0);

try{
    Connection c = KoneksiDatabase.getKoneksi();

    String sql = "DELETE FROM MAHASISWA WHERE NIM = ?";
```

Activate Wind

```
PreparedStatement p = c.prepareStatement(sql);

p.setString(1, nim);

p.executeUpdate();
p.close();
}catch(SQLException e){
    System.err.println("Terjadi Error");
}finally{
    loadData();
}
```

13) Menambah Aksi Baris Terseleksi

Aksi terakhir yang perlu kita tambahkan adalah aksi ketika baris Table terseleksi, misal jika baris pertama terseleksi, maka program akan menampilkan data yang terseleksi tersebut pada Form. Hal ini agar perubahan lebih mudah, karena kita tidak perlu memasukkan seluruh datanya lagi.

Untuk menambahkan aksi ketika baris terseleksi, kita dapat menggunakan aksi Mouse Click, yaitu aksi yang dijalankan ketika mouse mengklik. Caranya, klik kanan komponen Table pada Form, setelah itu pilih **Events** → **Mouse** → **mouseClicked**. Sekarang akan terbuat sebuah metode baru yang akan dipanggil ketika table di klik.

```
312
313 private void tabelMahasiswaMouseClicked(java
314 // TODO add your handling code here:
315 }
316
```

Pertama yang harus dilakukan adalah melakukan pengecekan apakah ada baris yang terseleksi, jika ada maka ambil data yang terseleksi dari DefaultTableModel setelah itu tampilkan pada Form, namun jika tidak ada baris yang terseleksi maka batalkan proses.

```
int i = tabelMahasiswa.getSelectedRow();
if(i == -1){
    // tak ada baris terseleksi
    return;
}

String nim = (String) model.getValueAt(i, 0);
inputNim.setText(nim);

String nama = (String) model.getValueAt(i, 1);
inputNama.setText(nama);

java.util.Date tanggalLahir = (java.util.Date) model.getValueAt(i, 2);
inputTanggalLahir.setValue(tanggalLahir);

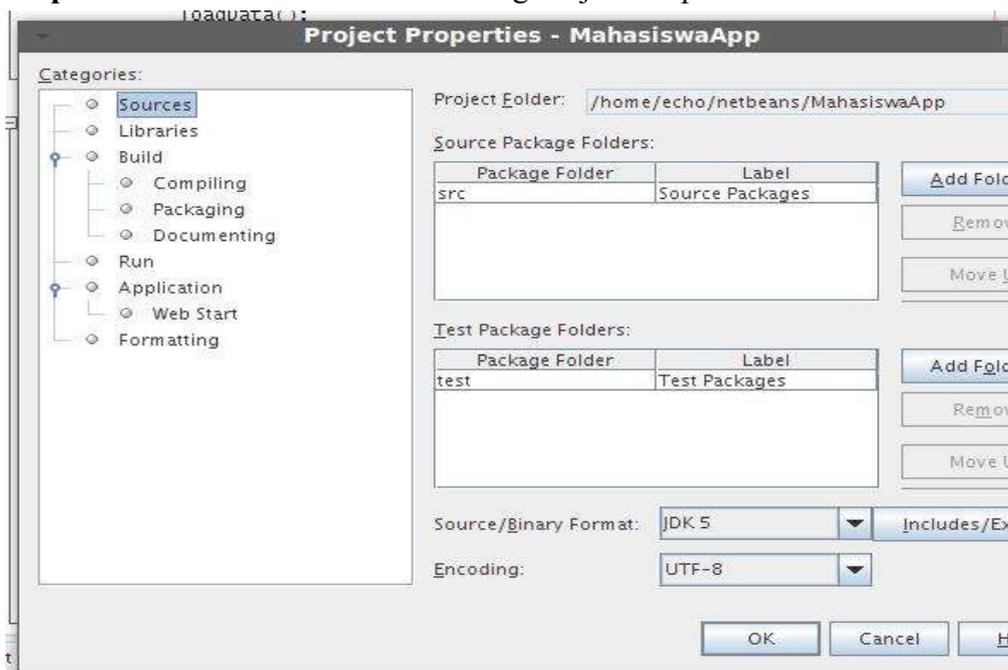
String jurusan = (String) model.getValueAt(i, 3);
inputJurusan.setText(jurusan);

String alamat = (String) model.getValueAt(i, 4);
inputAlamat.setText(alamat);
```

14) Menjalankan Program

Setelah pembuatan program telah selesai, saatnya menjalankan aplikasi. Untuk menjalankan aplikasi, pertama kita harus menentukan dahulu kelas yang akan digunakan sebagai program, dimana pada project yang telah kita buat, kelas program adalah kelas Form.

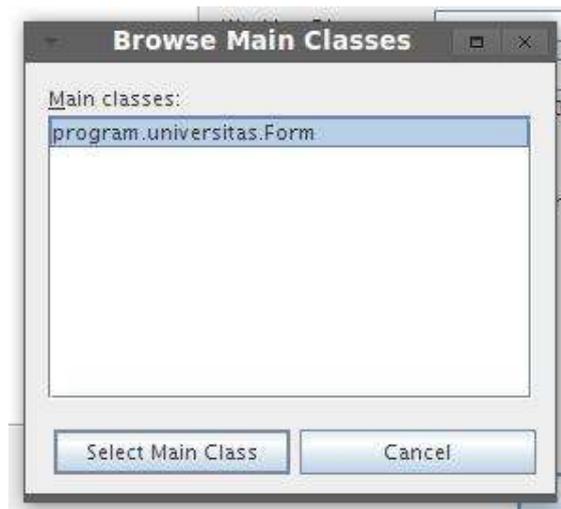
Untuk mengubah kelas program menjadi kelas Form, maka kita dapat mengubahnya dengan cara klik kanan Project yang telah kita buat, lalu pilih menu **Properties** setelah itu akan keluar dialog Project Properties.



Pada bagian categories, pilihlah menu Run, untuk mengubah kelas yang akan dijalankan sebagai kelas Program.



Pada input Main Class, klik tombol Browse untuk menampilkan daftar kelas yang dapat dijadikan sebagai program. Maka akan keluar dialog pemilihan kelas.



Pilih kelas Form yang telah dibuat tadi, setelah itu klik tombol Select Main Class, untuk mengkonfirmasi perubahan, klik tombol Ok. Sekarang kelas Form akan menjadi kelas yang dijalankan jika project dijalankan.

Untuk menjalankan aplikasi, klik kanan project yang telah kita buat, setelah itu pilih menu **Run**, maka otomatis program akan berjalan. Pada komputer saya, program terlihat seperti pada gambar dibawah ini.

A screenshot of a software application window. On the left side, there are five labels with corresponding input fields: "NIM :", "Nama :", "Tanggal Lahir :", "Jurusan :", and "Alamat :". The "Alamat :" field is a larger text area. To the right of these fields are three buttons: "Tambah", "Ubah", and "Hapus". Below the buttons is a table with five columns: "Nim", "Nama", "Tanggal Lahir", "Jurusan", and "Alamat". The table is currently empty.

Sekarang kita dapat menambah data, mengubah data dan menghapus data yang telah kita masukkan.

A screenshot of the same software application window, but now with data entered into the form. The input fields contain: "NIM : 10106032", "Nama : Tono Sumarno", "Tanggal Lahir : 12/5/1988", "Jurusan : Teknik Industri", and "Alamat : Dago - Bandung". The buttons "Tambah", "Ubah", and "Hapus" are still present. The table below now contains two rows of data:

Nim	Nama	Tanggal Lahir	Jurusan	Alamat
10106031	Eko Kurniaw...	1988-12-29	Teknik Infor...	Kalijati - Sub...
10106032	Tono Sumarno	1988-05-12	Teknik Industri	Dago - Band...

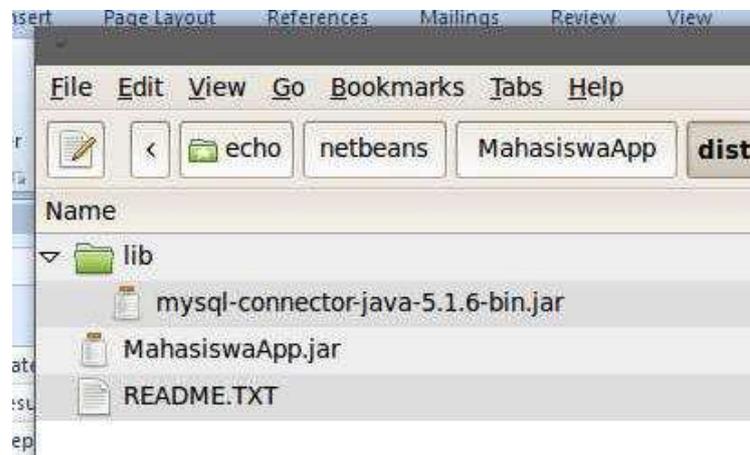
15) Mendistribusikan Program

Saat ini kita telah menyelesaikan program yang kita buat, saatnya mendistribusikan program yang kita buat ke komputer lain. Hal ini perlu dilakukan agar untuk menjalankan program kita tidak perlu menggunakan NetBeans IDE lagi.

Langkah-langkah pertama adalah build project yang telah kita bangun tadi agar menjadi file yang siap didistribusikan. Caranya klik kanan project-nya lalu pilih **Clean and Build**. Setelah selesai maka akan tercipta sebuah folder **dist** pada folder project yang telah kita buat.



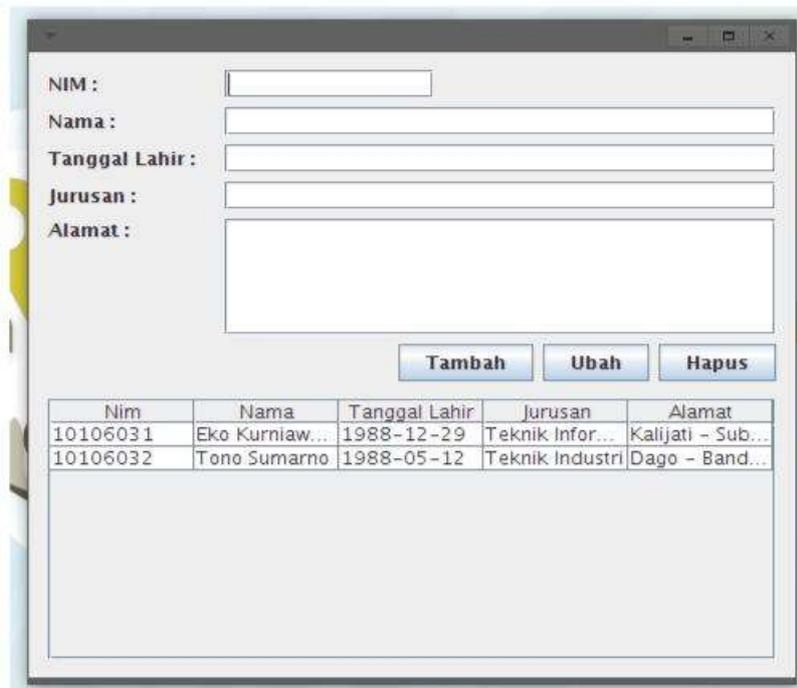
Dalam folder dist tersebut sebuah file JAR yang bernama MahasiswaApp, sesuai dengan nama project yang kita buat. Selain itu akan ada folder lib yang berisikan library-library yang dibutuhkan oleh program yang kita buat, misal driver untuk MySQL.



Untuk menjalankan dari luar NetBeans, kita hanya perlu menjalankan menggunakan terminal atau command menggunakan perintah:

```
Java -jar MahasiswaApp.jar
```

maka program akan berjalan seperti berjalan pada NetBeans IDE.



Nim	Nama	Tanggal Lahir	Jurusan	Alamat
10106031	Eko Kurniaw...	1988-12-29	Teknik Infor...	Kalijati - Sub...
10106032	Tono Sumarno	1988-05-12	Teknik Industri	Dago - Band...

BAB IV

PENUTUP

4.1 Kesimpulan

Kesimpulan dari semua pembahasan di atas, dapat diambil. Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Kita lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi Java memiliki tiga komponen penting, yaitu:

- 1) Programming
- 2) Language specification
- 3) Application-programming interface
- 4) Virtual-machine specification

NetBeans IDE adalah IDE open source yang ditulis sepenuhnya dengan bahasa Java menggunakan platform NetBeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi java. Semua fungsi IDE disediakan oleh modul-modul. Tiap modul menyediakan fungsi yang didefinisikan dengan baik, seperti dukungan untuk bahasa pemrograman Java, editing, atau dukungan bagi CVS. NetBeans memuat semua modul yang diperlukan dalam pengembangan java dalam sekali download, memungkinkan pengguna untuk mulai bekerja sesegera mungkin.

4.2 Saran

Dari semua pembahasan diatas, modul ini ditujukan bagi seorang yang ingin menjadi programmer dan ingin memulai membuat suatu aplikasi yang sederhana melalui Java NetBeans IDE.