

PEMROGRAMAN VISUAL

KONEKSI JAVA
DATABASE
(PART 1)

KONEKSI JAVA DATABASE

- Koneksi ke database pada Java ditangani oleh JDBC (Java Database Connectivity).
- JDBC merupakan salah satu API (Application Programming Interface) yang secara khusus ditujukan untuk menangani koneksi ke database.
- Antarmuka ini memungkinkan pemrogram menulis sebuah program yang dapat digunakan untuk mengakses database yang berbeda-beda, misalnya Oracle, Access, atau MySQL.

BEKERJA DENGAN JDBC

Secara umum, untuk bekerja dengan JDBC, langkah yang dilakukan adalah :

- Me-load JDBC Driver ke dalam JVM. JDBC Driver dapat dipandang sebagai library yang dibutuhkan untuk mengakses database tertentu.
- Membuat koneksi ke database yang direpresentasikan sebagai objek `java.sql.Connection`.
- Membuat objek `java.sql.Statement` yang akan digunakan untuk mengirimkan perintah SQL ke database.
- Menjalankan metode yang bersesuaian dari objek `java.sql.Statement`, seperti `executeQuery()` atau `executeUpdate()`.

MEMBUAT KONEKSI

- Melakukan koneksi ke database dapat dilakukan dengan : Meload driver dan membuat koneksi itu sendiri.
- Cara meload driver adalah dengan meletakkan file jar database driver ke dalam classpath. Kemudian load driver dengan menambahkan kode berikut:

```
Class.forName("com.mysql.jdbc.Driver");
```

MEMBUAT KONEKSI

- Nama class database driver untuk setiap DBMS berbeda-beda, nama class tersebut bisa ditemukan dalam dokumentasi driver database yang digunakan.
- Dalam contoh ini, nama class database driver dari MySQL adalah `com.mysql.jdbc.Driver`.
- Memanggil method `Class.forName` secara otomatis membuat instance dari database driver, class `DriverManager` secara otomatis juga dipanggil untuk mengelola class database driver ini.

MEMBUAT KONEKSI

- Langkah berikutnya adalah membuat koneksi ke database menggunakan database driver yang sudah di-load sebelumnya.
- Class DriverManager bekerja sama dengan interface Driver untuk mengelola driver-driver yang di-load oleh aplikasi. Dalam satu sesi bisa meload beberapa database driver yang berbeda. Ketika melakukan koneksi, JDBC Test Suite akan melakukan serangkaian tes untuk menentukan driver mana yang akan digunakan. Parameter yang digunakan untuk menentukan driver yang sesuai adalah URL. Aplikasi yang akan melakukan koneksi ke database menyediakan URL pengenalan dari server database tersebut.

MEMBUAT KONEKSI

- Sebagai contoh adalah URL yang digunakan untuk melakukan koneksi ke MySql :

```
jdbc:mysql://[host]:[port]/[schema]
```

contoh :

```
jdbc:mysql://localhost:3306/latihan
```

MEMBUAT KONEKSI

- Setiap vendor DBMS akan menyertakan cara untuk menentukan URL ini di dalam dokumentasi.
- Method `DriverManager.getConnection` bertugas untuk membuat koneksi:

```
Connection conn = DriverManager.getConnection  
("jdbc:mysql://localhost:3306/latihan");
```


MEMBUAT KONEKSI

- Dalam kebanyakan kasus juga harus memasukkan parameter username dan password untuk dapat melakukan koneksi ke dalam database. Method `getConnection` menerima Username sebagai parameter kedua dan password sebagai parameter ketiga, sehingga kode diatas dapat dirubah menjadi :

```
Connection conn = DriverManager.getConnection  
("jdbc:mysql://localhost:3306/latihan","root","");
```

MEMBUAT KONEKSI

- Jika salah satu dari driver yang di-load berhasil digunakan untuk melakukan koneksi dengan URL tersebut, maka koneksi ke database berhasil dilaksanakan. Class Connection akan memegang informasi koneksi ke database yang didefinisikan oleh URL tersebut.
- Setelah sukses melakukan koneksi ke database, kita dapat mengambil data dari database menggunakan perintah query ataupun melakukan perubahan terhadap database. bagian berikut ini akan menerangkan bagaimana cara mengambil dan memanipulasi data dari database.

CONTOH PROGRAM KONEKSI DATABASE

~ DATA PASIEN ~

No. Identitas

Nama Pasien

Jenis Kelamin Laki-laki Perempuan

Alamat - No. telp

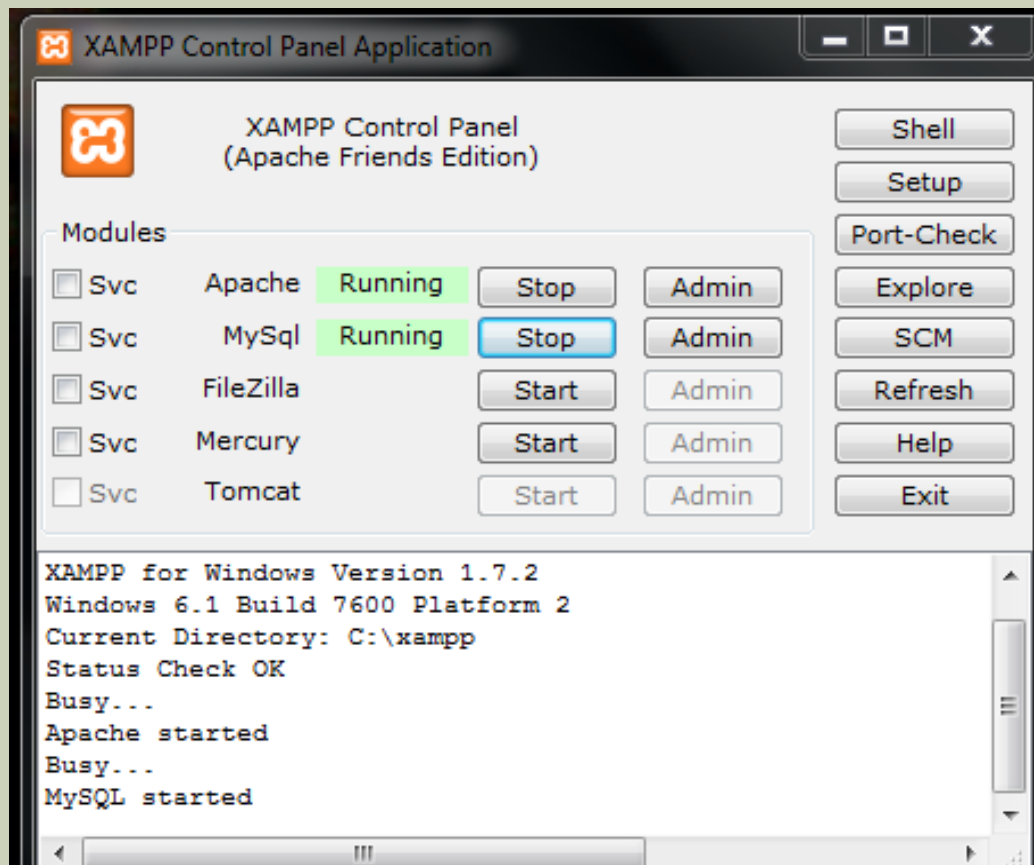
Golongan Darah

Kata kunci pencarian

No. Identitas	Nama Pasien	Jenis Kelamin	Alamat	Golongan Darah

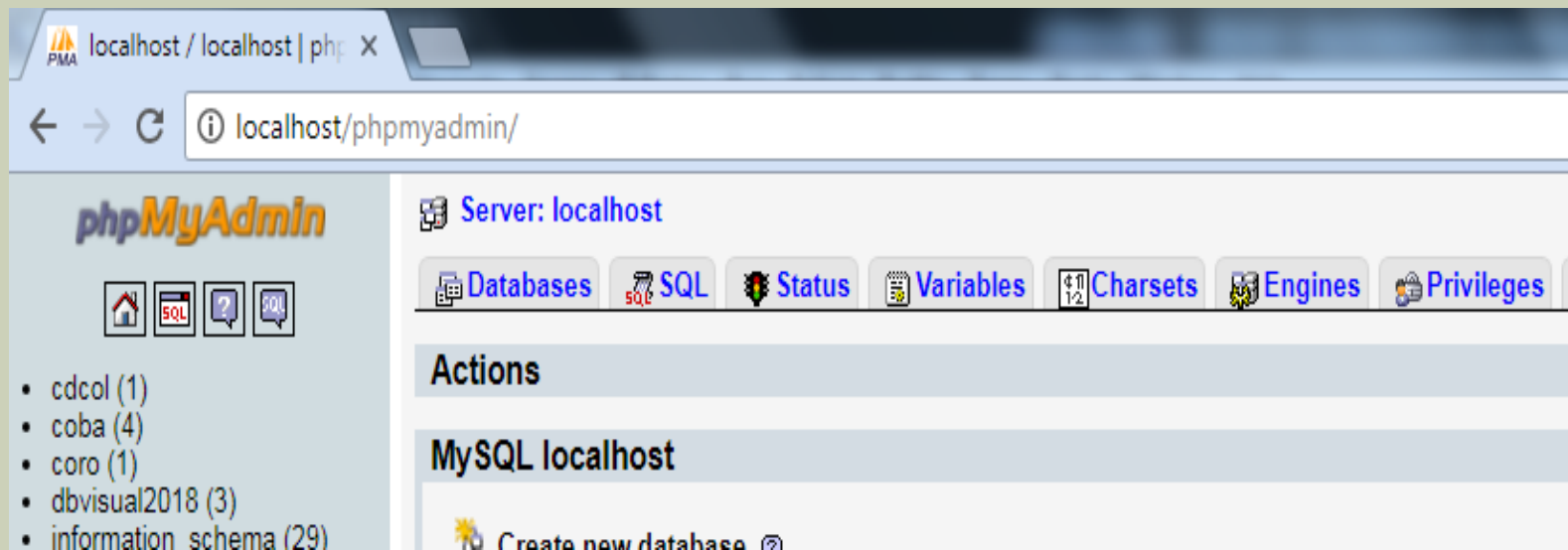
MEMBUAT DATABASE DAN TABEL

- Buka XAMPP Control Panel, Aktifkan Apache dan MySQL



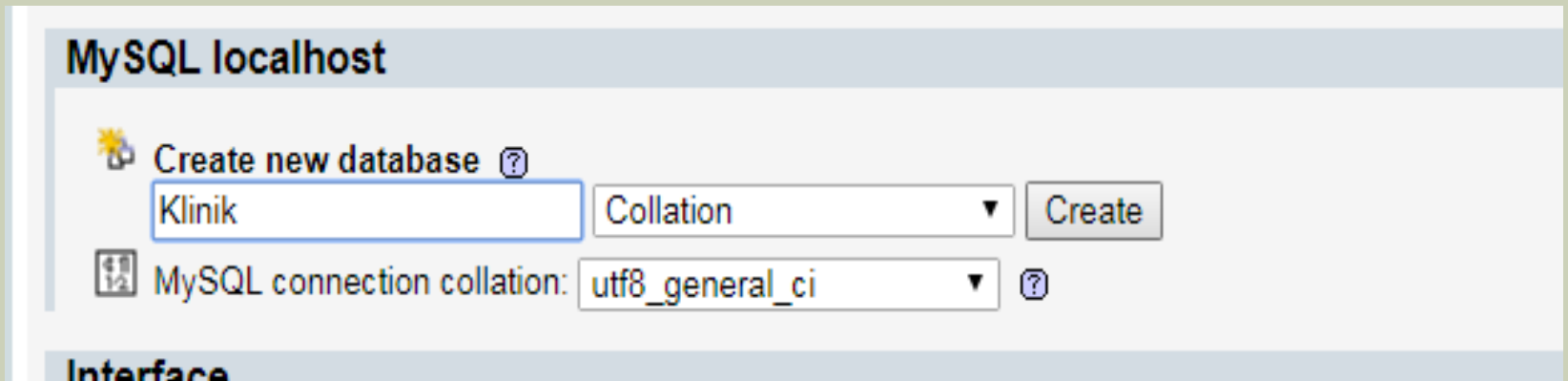
MEMBUAT DATABASE DAN TABEL

- Buka phpMyAdmin melalui browser



MEMBUAT DATABASE DAN TABEL

- Buat database dengan nama **Klinik**



The screenshot shows the MySQL localhost interface. At the top, it says "MySQL localhost". Below that, there is a section titled "Create new database" with a question mark icon. The form contains a text input field with the name "Klinik", a dropdown menu for "Collation" (which is currently empty), and a "Create" button. Below this, there is another section titled "MySQL connection collation:" with a dropdown menu showing "utf8_general_ci" and a question mark icon. At the bottom of the interface, the word "Interface" is partially visible.

MEMBUAT DATABASE DAN TABEL

- Buat tabel dengan nama **pasien** dengan jumlah kolom : 5

No tables found in database.

 Create new table on database [Klinik](#)

Name:

Number of fields:

MEMBUAT DATABASE DAN TABEL

- Buat field di dalam tabel pasien sebagai berikut :

Server: localhost ▶ Database: Klinik ▶ Table: pasien

Field	Type	Length/Values ¹	Default ²	Collation	Attributes
id	VARCHAR	30	None		
nama	VARCHAR	100	None		
jk	VARCHAR	20	None		
alamat	VARCHAR	200	None		
goldar	VARCHAR	2	None		

Table comments:

Storage Engine: MyISAM

Collation:

PARTITION definition:

Save Or Add 1 field(s) Go




































MEMBUAT DATABASE DAN TABEL







- Setelah selesai klik save

Server: localhost ▶ Database: Klinik ▶ Table: pasien

✓ Table 'Klinik'. 'pasien' has been created.

```
CREATE TABLE `Klinik`.`pasien` (
  `id` VARCHAR(30) NOT NULL,
  `nama` VARCHAR(100) NOT NULL,
  `jk` VARCHAR(20) NOT NULL,
  `alamat` VARCHAR(200) NOT NULL,
  `goldar` VARCHAR(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE = MYISAM;
```

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<u>id</u>	varchar(30)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	nama	varchar(100)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	jk	varchar(20)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	alamat	varchar(200)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	goldar	varchar(2)	latin1_swedish_ci		No	None		      

↑ Check All / Uncheck All With selected:      

MEMBUAT TAMPILAN DAN KONEKSI

- Buat project baru dengan nama AplikasiKlinik

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

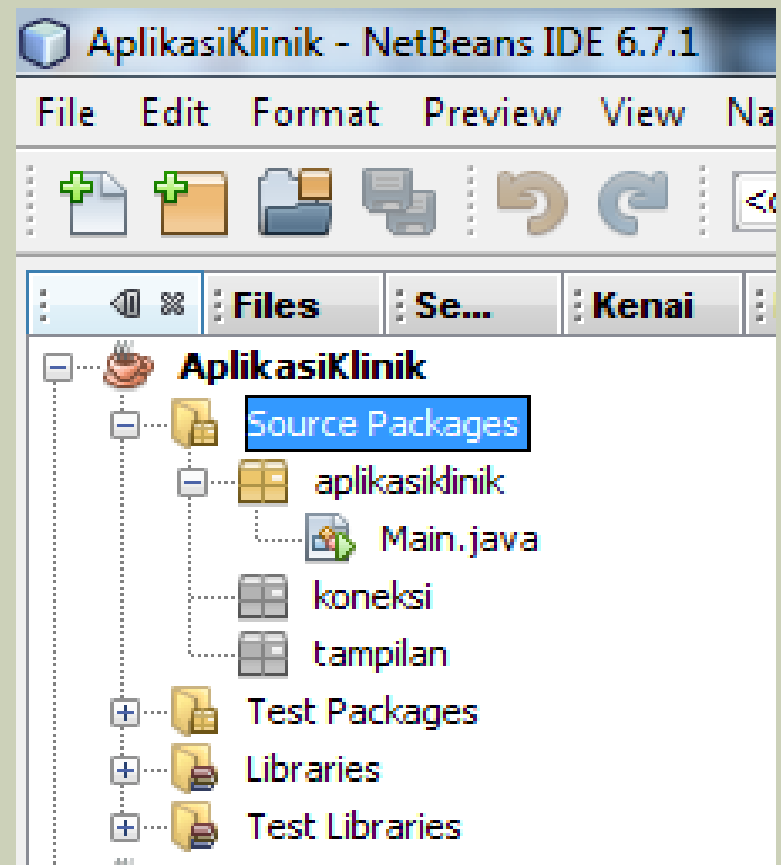
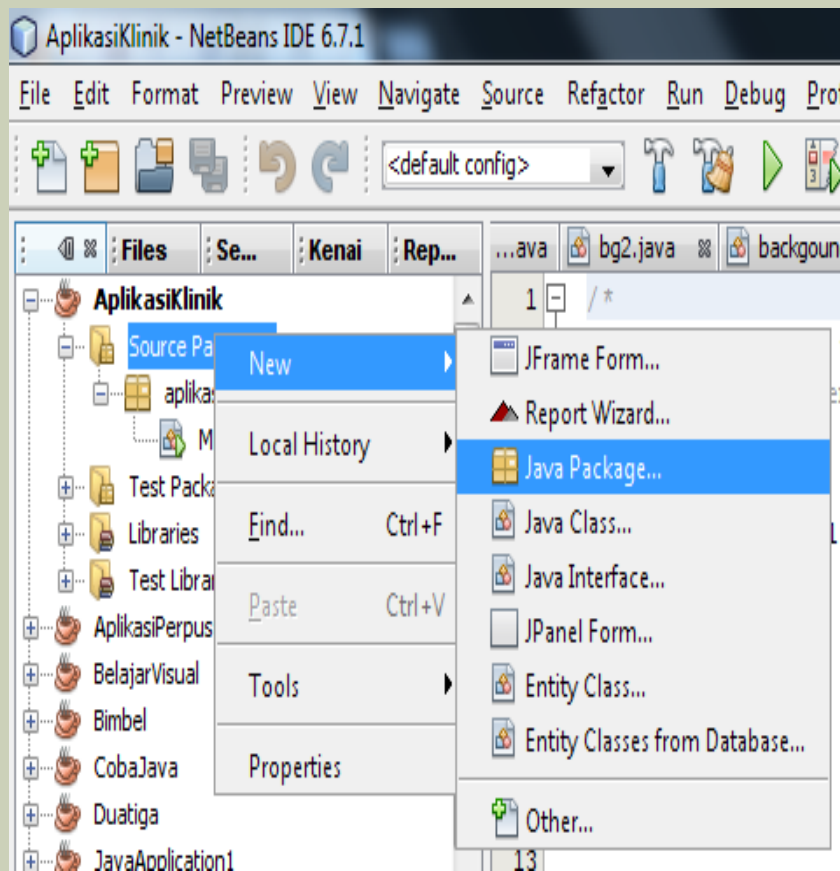
Create Main Class

Set as Main Project

< Back Next > **Finish** Cancel Help

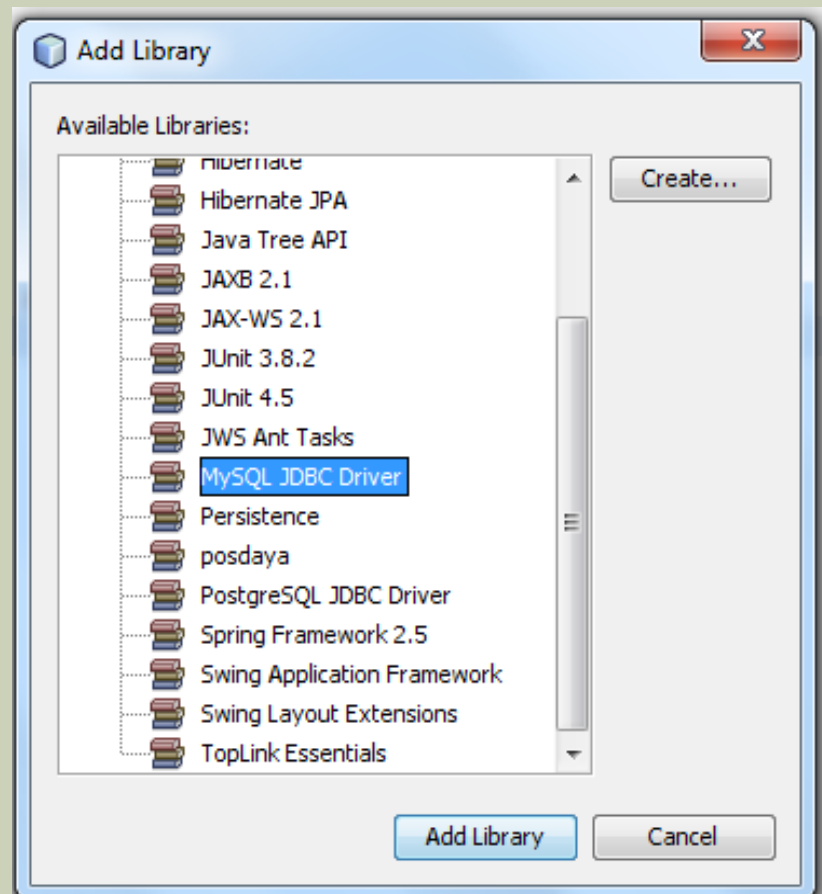
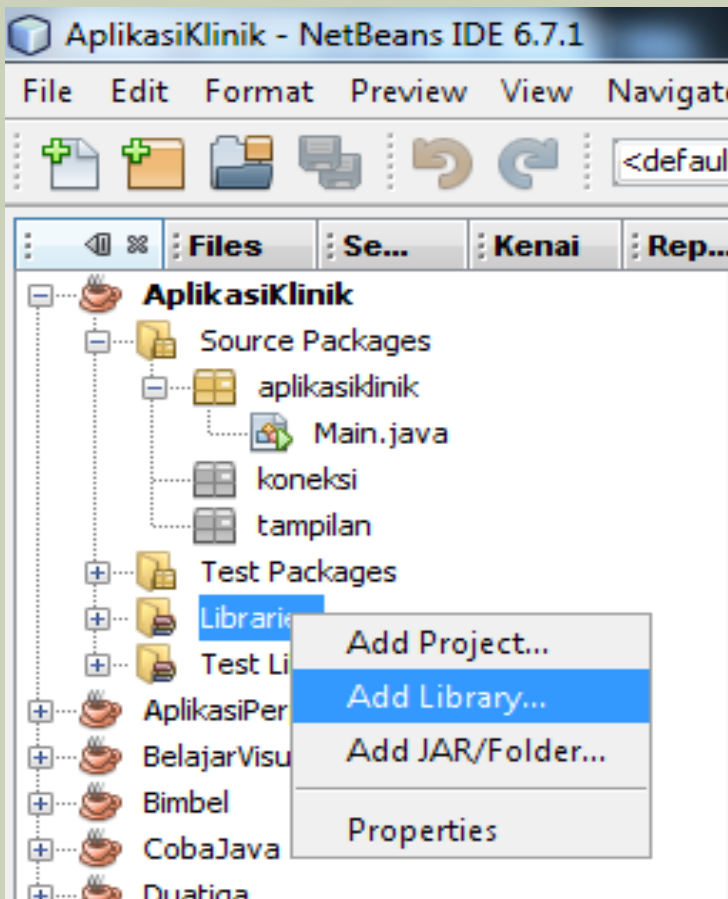
MEMBUAT TAMPILAN DAN KONEKSI

- Buat dua buah package dengan nama koneksi dan tampilan



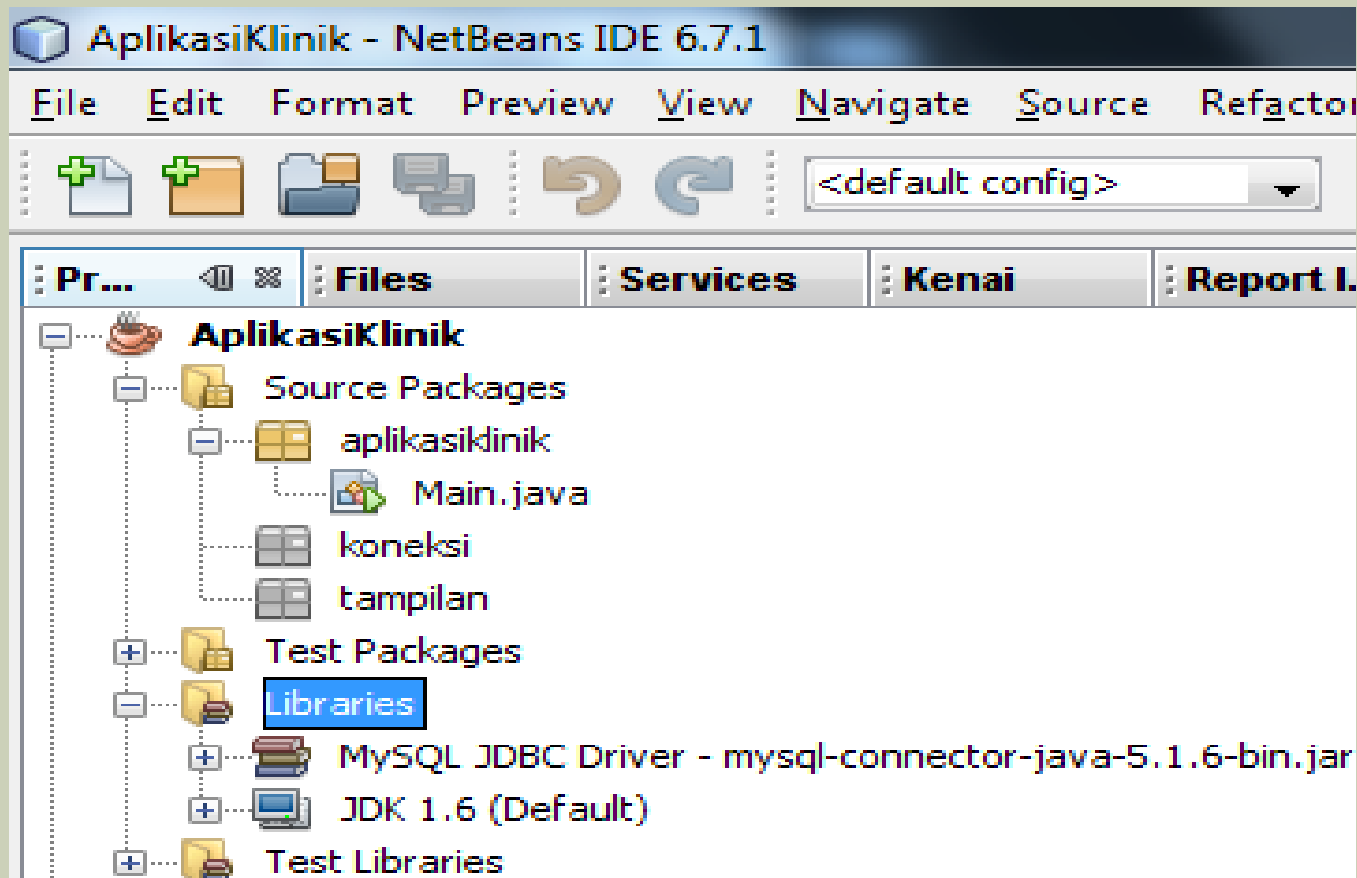
MEMBUAT TAMPILAN DAN KONEKSI

- Tambahkan driver JDBC dalam Libraries >> Add Library ... >> Pilih MySQL JDBC Driver



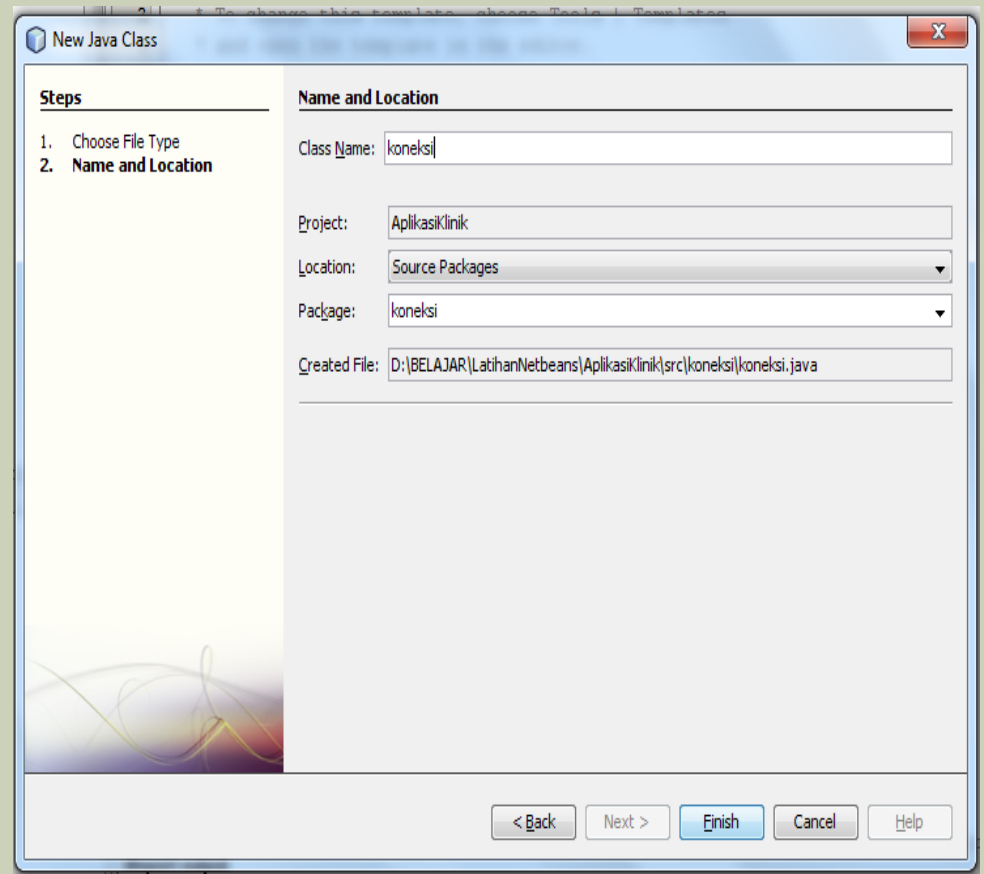
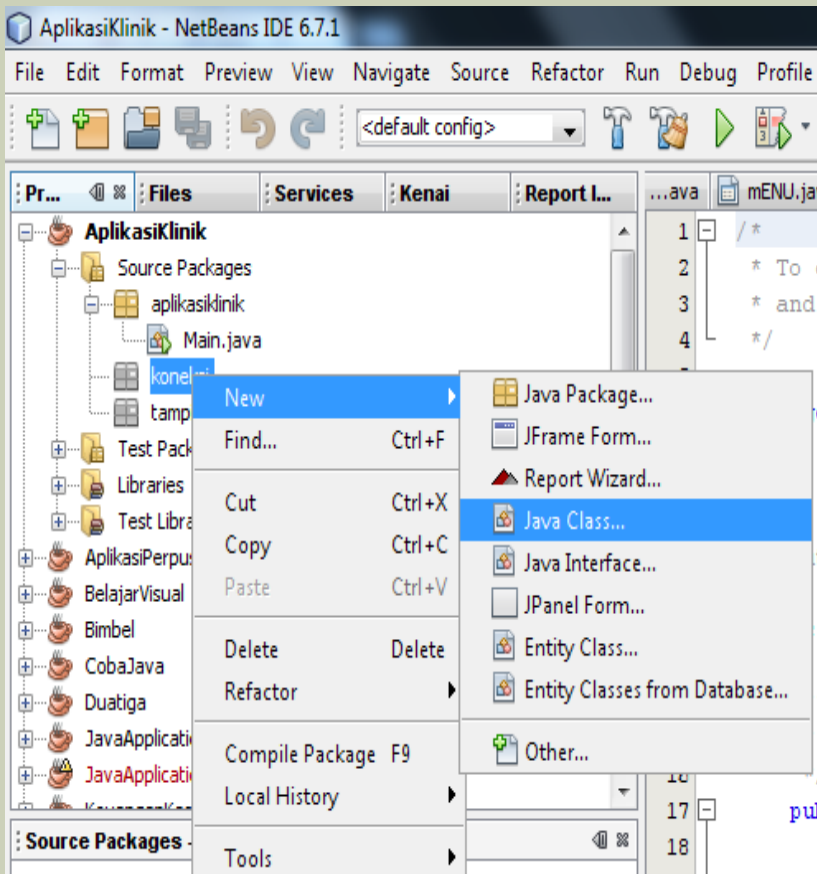
MEMBUAT TAMPILAN DAN KONEKSI

- Setelah klik add Library maka MySQL JDBC Driver akan ada di dalam Libraries project kita



MEMBUAT TAMPILAN DAN KONEKSI

- Buat java class baru di dalam package koneksi dengan nama koneksi



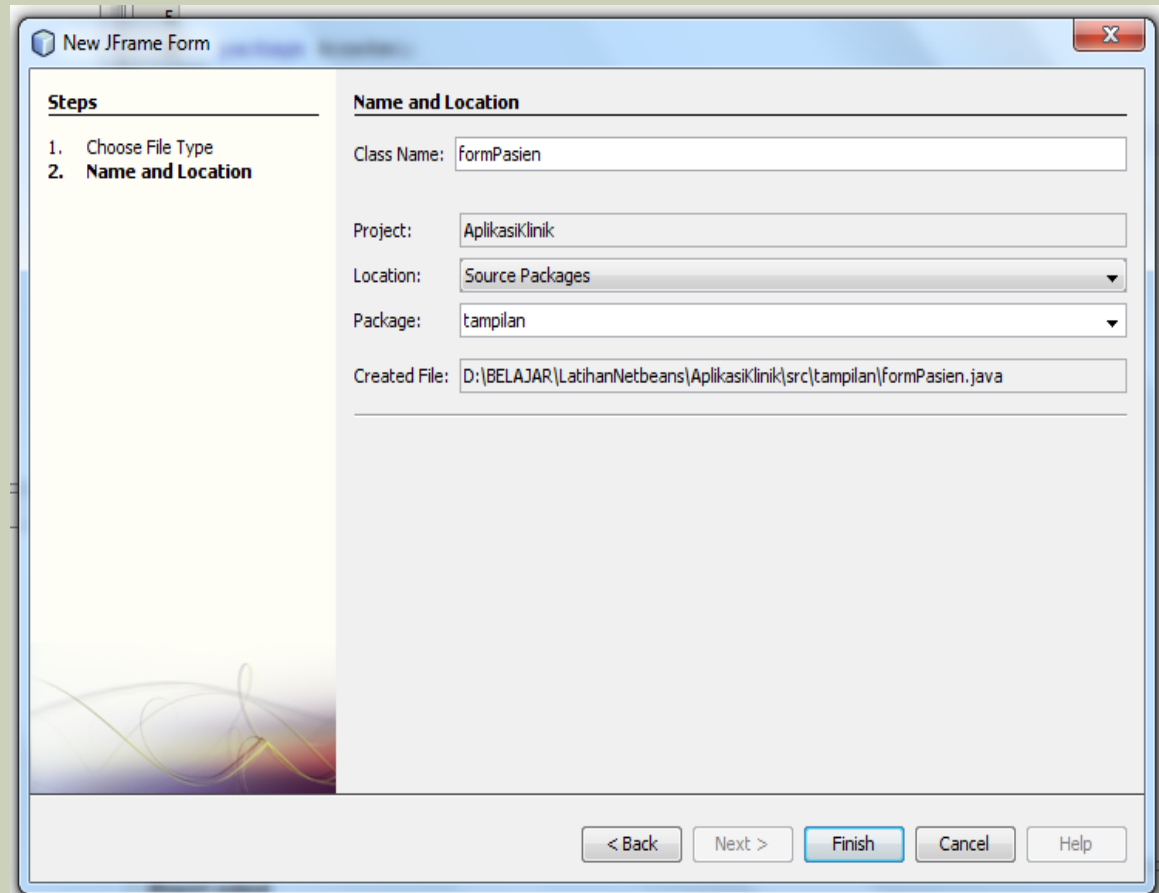
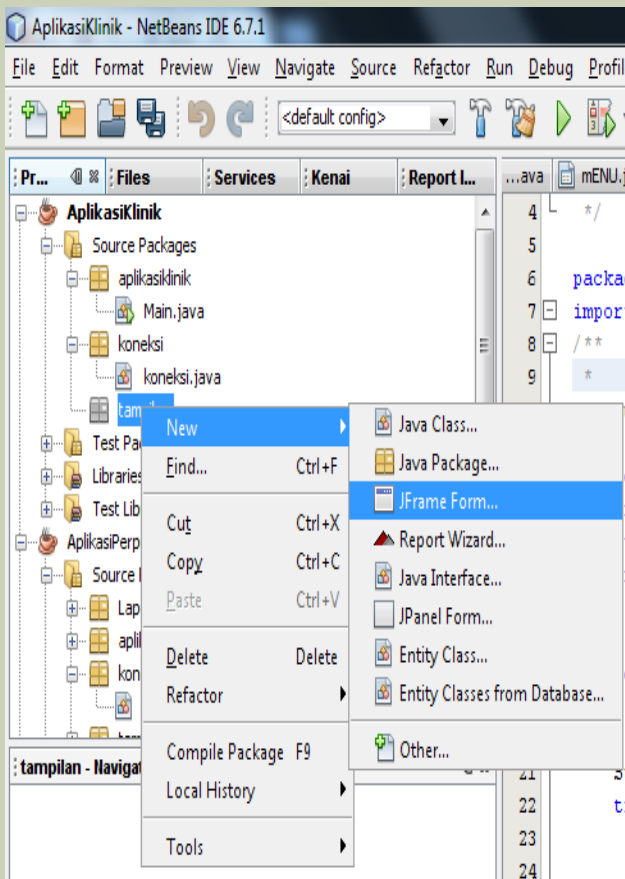
MEMBUAT TAMPILAN DAN KONEKSI

- Tulis sintaks berikut di dalam class koneksi

```
package koneksi;
import java.sql.*;
/**
 * |
 * @author funNy
 */
public class koneksi {
    private Connection koneksi;
    public Connection connect () {
        try {
            Class.forName ("com.mysql.jdbc.Driver");
            System.out.println("Berhasil Koneksi");
        }catch (ClassNotFoundException ex){
            System.out.println("Gagal Koneksi "+ex);
        }
        String url = "jdbc:mysql://localhost/Klinik";
        try {
            koneksi = DriverManager.getConnection(url,"root","");
            System.out.println("Berhasil Koneksi Database");
        }catch (SQLException ex){
            System.out.println("Gagal Koneksi Database "+ex);
        }
        return koneksi;
    }
}
```

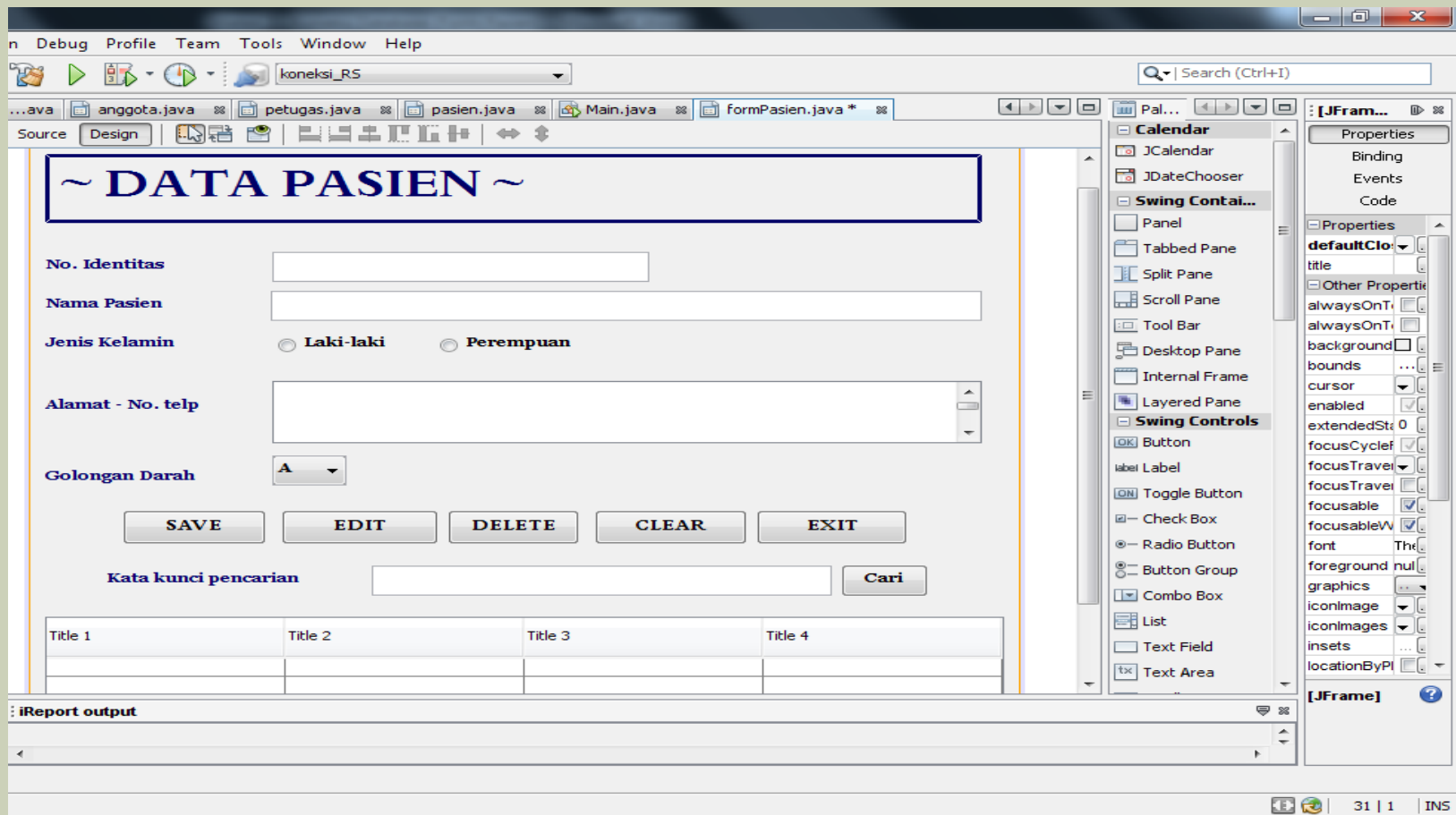
MEMBUAT TAMPILAN DAN KONEKSI

- Pada package tampilan, tambahkan JFrame Form dengan nama formPasien



MEMBUAT TAMPILAN DAN KONEKSI

- Buat tampilan formPasien sebagai berikut :



MEMBUAT TAMPILAN DAN KONEKSI

- Ubah text dan nama variabel masing-masing komponen sebagai berikut

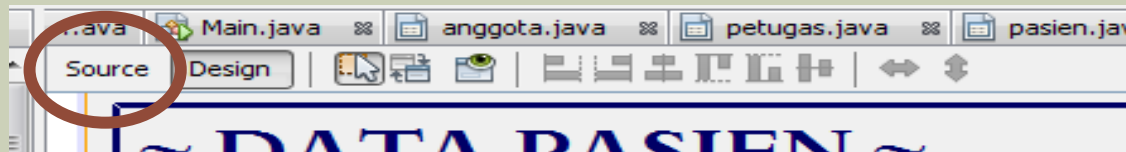
The image shows a screenshot of an IDE (likely NetBeans) with a Java Swing form titled "DATA PASIEN". The form is displayed in the "Design" view. The IDE interface includes a "Source Packages" view on the left, a "Navigator" showing the project structure, and an "Inspector" showing the component hierarchy. The form itself has a title bar with the text "~ DATA PASIEN ~". Below the title bar, there are several input fields and buttons:

- No. Identitas**: A text input field.
- Nama Pasien**: A text input field.
- Jenis Kelamin**: Two radio buttons labeled "Laki-laki" and "Perempuan".
- Alamat - No. telp**: A text input field with a vertical scrollbar.
- Golongan Darah**: A dropdown menu with the letter "A" selected.
- Buttons**: Five buttons labeled "SAVE", "EDIT", "DELETE", "CLEAR", and "EXIT".
- Kata kunci pencarian**: A text input field with a "Cari" button next to it.
- Table**: A table with 4 columns labeled "Title 1", "Title 2", "Title 3", and "Title 4".

The IDE interface also shows a "Source Packages" view on the left with "AplikasiKlinik" and "Main.java". The "Navigator" shows the project structure, and the "Inspector" shows the component hierarchy for "Form formPasiien".

MEMBUAT TAMPILAN DAN KONEKSI

- Pada source ketik sintaks berikut :



```
12 package tampilan;
13 import java.sql.*;
14 import javax.swing.JOptionPane;
15 import javax.swing.table.DefaultTableModel;
16 import koneksi.koneksi;
17
18 /**
19  *
20  * @author funNy
21  */
22 public class formPasiien extends javax.swing.JFrame {
23     private Connection conn = new koneksi().connect();
24     private DefaultTableModel tabmode;
25
26     /** Creates new form formPasiien */
27     public formPasiien() {
28         initComponents();
29     }
30 }
```

MEMBUAT TAMPILAN DAN KONEKSI

- Buat method aktif(), kosong(), dan datatable sebagai berikut :

```
protected void aktif() {  
    tid.setEnabled(true);  
    tnama.setEnabled(true);  
    talm.setEnabled(true);  
    tid.requestFocus();  
}
```

```
protected void kosong() {  
    tid.setText("");  
    tnama.setText("");  
    talm.setText("");  
    rjkl.setSelected(true);  
    cgd.setSelectedIndex(0);  
    tcari.setText("");  
}
```

MEMBUAT TAMPILAN DAN KONEKSI

```
protected void datatable(){
    Object[] Baris ={"No Identitas","Nama Pasien","Jenis Kelamin","Alamat","Golongan Darah"};
    tabmode = new DefaultTableModel(null, Baris);
    tabelpasien.setModel(tabmode);
    String sql = "select * from pasien";
    try {
        java.sql.Statement stat = conn.createStatement();
        ResultSet hasil = stat.executeQuery(sql);
        while(hasil.next()){
            String a = hasil.getString("id");
            String b = hasil.getString("nama");
            String c = hasil.getString("jkk");
            String d = hasil.getString("alamat");
            String e = hasil.getString("goldar");

            String[] data={a,b,c,d,e};
            tabmode.addRow(data);
        }
    } catch (Exception e) {
    }
}
```

MEMBUAT TAMPILAN DAN KONEKSI

- Tambahkan method datatable di bawah initComponents();

```
public formPasien() {  
    initComponents();  
    datatable();  
}
```

TUGAS

- **Buat Form Admin (terkoneksi dengan database)**
- **Sesuaikan dengan project yang Anda buat**