

TUTORIAL MEMBUAT APLIKASI CRUD

DENGAN PEMROGRAMAN JAVA DEKSTOP



Disusun oleh

Megandi

Me.gandi@rocketmail.com

• Pengenalan Bahasa Java

Java adalah sebuah bahasa pemrograman yang berdasarkan OOP (Object Oriented Programing) yang pada awalnya dikembangkan oleh SUN kemudian sekarang diakuisisi dan dikembangkan oleh ORACLE. Bahasa pemrograman ini digunakan untuk membuat aplikasi yang dapat berjalan di desktop komputer maupun mobile.

Java ini tidak tergantung platform/system komputer baik sourcode maupun hasil kompilasinya, jadi bisa dibilang dapat berjalan baik di windows maupun di UNIX.

Seiring dengan kemajuan zaman, pemrograman Java ini berkembang menjadi bermacam-macam, Java EE (Web app), Java/Java SE (Desktop App), Java ME (Mobile), Java FX (Next Generation).

Pada tutorial ini kita akan membahas bagaimana kita menggunakan bahasa Java untuk membuat aplikasi berbasis database dengan menggunakan sebuah framework ORM yaitu Hibernate.

• Pengenalan ORM

ORM merupakan kependekan dari Object Relational Mapping, sebuah teknik pemrograman yang memetakan sebuah objek dengan database. ORM ini akan membantu menjadi 'jembatan' antara objek yang didefinisikan dalam kode program dengan database, bagaimana objek itu disimpan, diambil, dihapus dan sebagainya. Dengan ORM ini, kita dibantu untuk melakukan aksi-aksi yang diperlukan terkait komunikasi objek ketika program dijalankan dengan database seperti menyimpan objek, mengambil data objek dari database kemudian ditampilkan, menghapus objek, mengubah objek dan sebagainya.

Ada banyak ORM yang tersedia saat ini, Di Java misalkan ada Hibernate (yang nanti akan kita pakai) dan Mylbatis (dulu namanya IBatis). Di PHP, berbagai framework seperti Laravel, CodeIgniter, Yii dan sebagainya menyediakan ORMnya sendiri-sendiri.

• Pengenalan Hibernate

Hibernate adalah sebuah framework ORM (Object Relation Mapping) dalam java. Hibernate memudahkan kita untuk bekerja dengan model. Dengan Hibernate, tabel-tabel yang ada di Database dapat direpresentasikan ke dalam class-class dalam Java. Setiap anda membuat aplikasi java yang terhubung dengan database anda pasti mengetik yang namanya query sql, connection, statement / prepare statement, resultset, dll. Dan itu dilakukan secara berulang-ulang. Nah dengan Hibernate masalah koneksi database diatur menjadi satu file yang biasanya dinamai hibernate.cfg.xml dan table dalam database kita dikelompokkan dalam satu package yang biasanya dinamai *model*. itulah sedikit kemudahan dari hibernate untuk lebih jelasnya silahkan kunjungi web resminya langsung www.hibernate.org/

• CRUD

CRUD adalah singkatan dari Create, Read, Update dan Delete. Crud atau Create Read Update Delete merupakan hal dasar bagi setiap aplikasi yang berhubungan dengan Database.

- **Create** Memasukkan (row) data baru ke dalam database
- **Read** mendapatkan informasi dari database
- **Update** Mengubah nilai di dalam database
- **Delete** Menghapus data dari database

• Software Pendukung

Software yang dibutuhkan untuk membuat aplikasi CRUD sederhana menggunakan bahasa Java Dekstop dengan konsep Hibernate MVC antara lain :

- Mysql | XAMPP
- Netbeans (Versi dalam tutorial 7.4)
- JDK

Dalam tutorial ini database yang digunakan adalah Mysql dengan menggunakan XAMPP Control Panel. Tetapi jangan khawatir bagi kalian yang tidak menguasai database jenis mysql, kalian bisa menggunakan jenis database yang lain. Langkah pembuatan dalam source code yang nanti dibentuk pun tetap sama. Itulah kelebihan dalam menggunakan konsep Hibernate MVC 😊. Oke mungkin kita harus langsung memulai langkah – langkahnya.

Langkah – Langkah Membuat Aplikasi

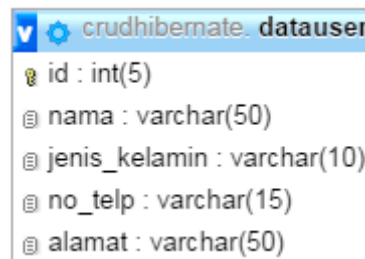
- **Konfigurasi Database**

Buat database baru dengan nama “crudhibernate” lalu buat table dengan mengikuti informasi seperti dibawah ini :

Table Name : “datauser”

Nama Field	Tipe Data	Panjang	Keterangan
id	Integer	5	Primary Key
nama	Varchar	50	-
Jenis_kelamin	Varchar	10	-
No_telp	Varchar	15	-
alamat	Varchar	50	-

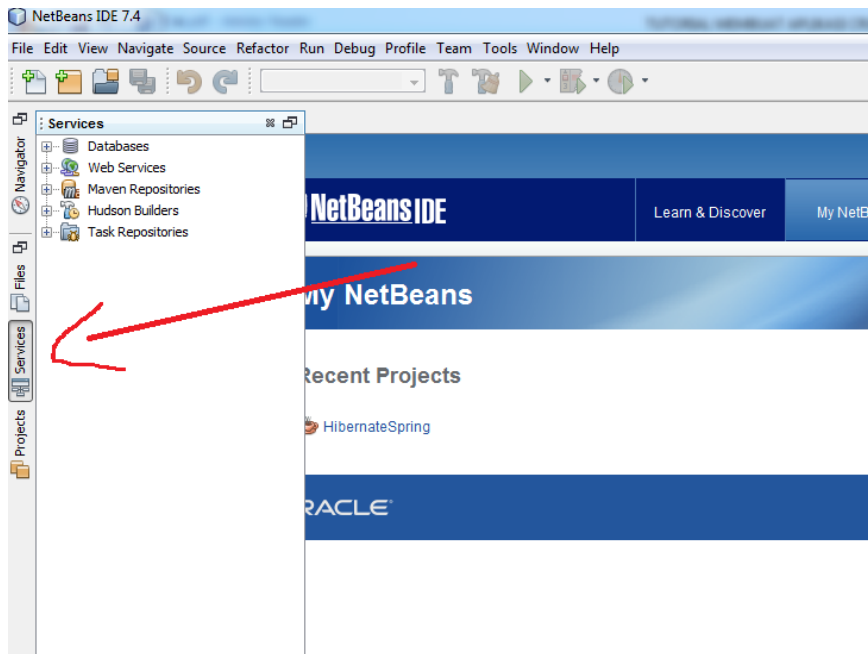
Jika kurang jelas dengan informasi table diatas, kamu bisa lihat dengan informasi database berupa gambar seperti berikut :



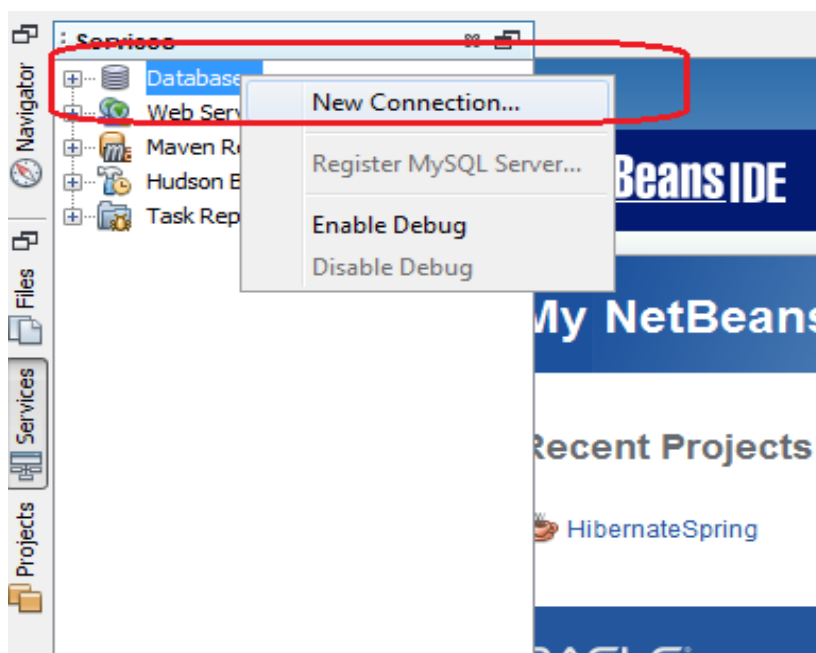
- Membuat koneksi project di Netbeans

- **Membuat koneksi**

Pilih menu services yang terdapat dalam netbeans

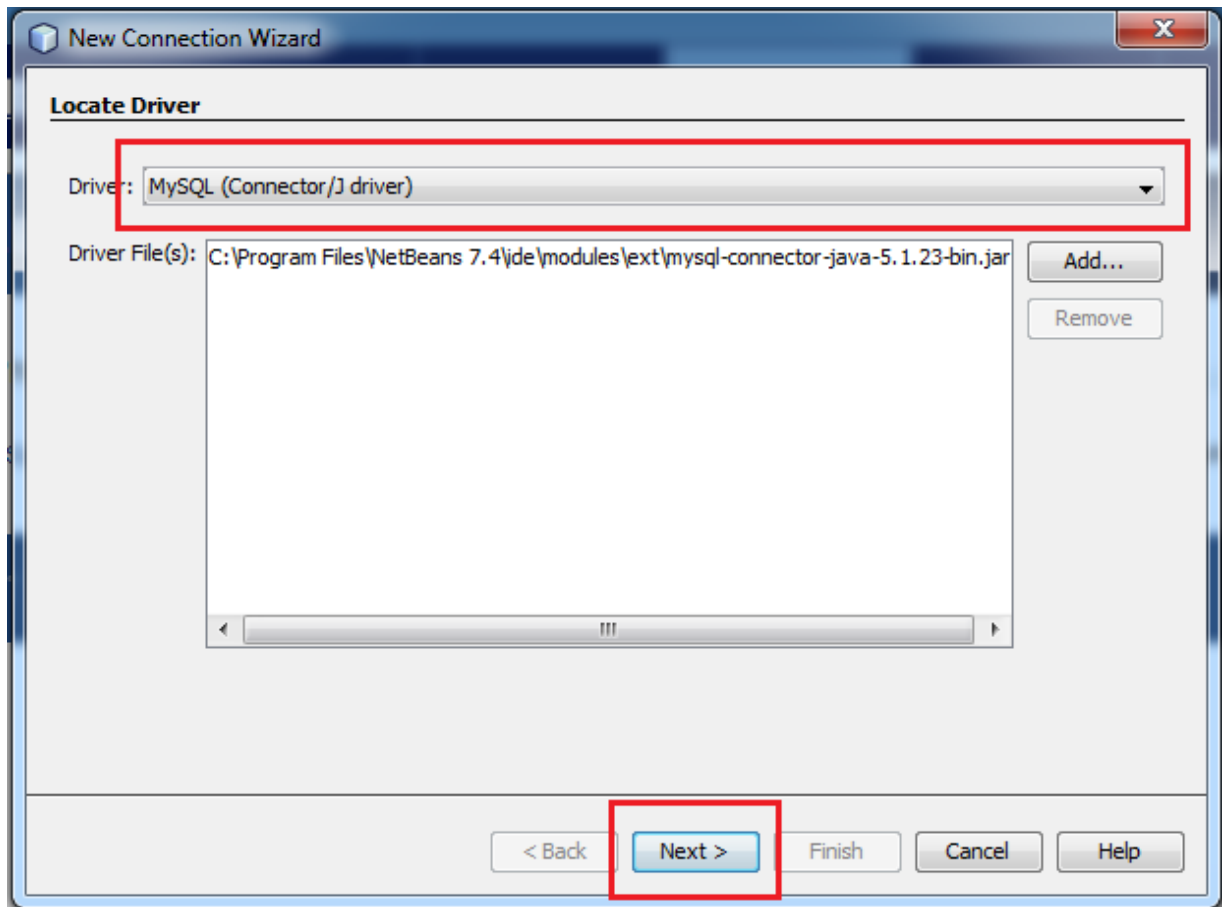


Setelah itu klik kanan pada database dan buatlah koneksi baru

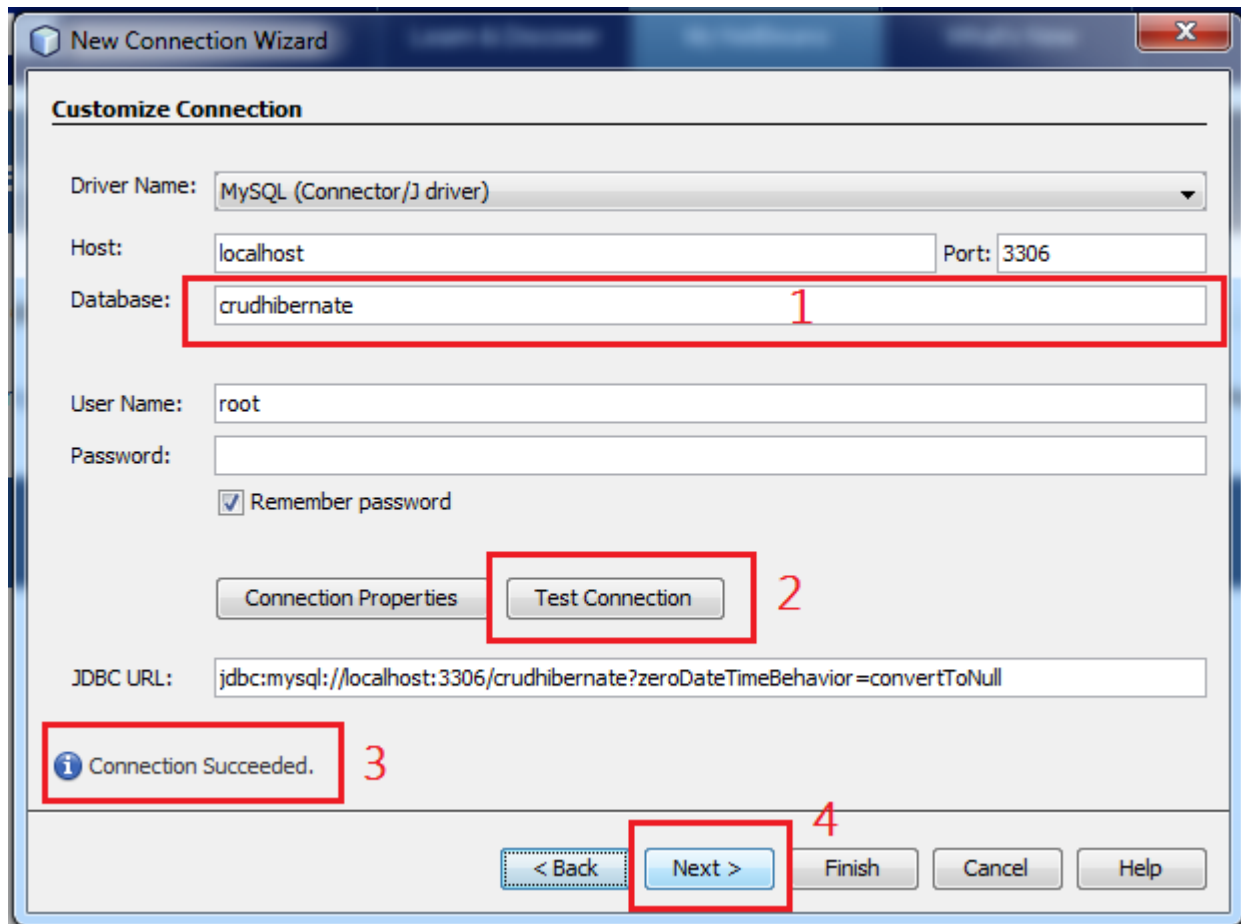


- **Pembuatan informasi koneksi database**

Pilih Driver database. Berhubungan karena database yang saya gunakan mysql maka saya pilih mysql (Connector/ J Driver).

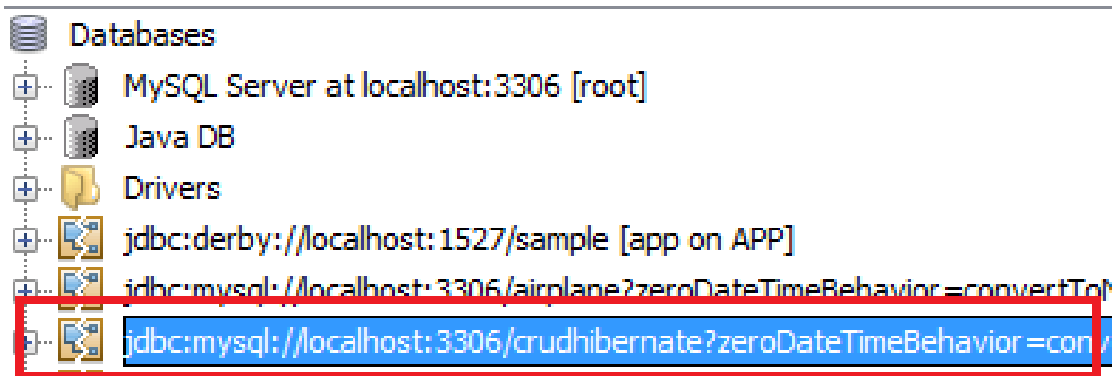


Setelah klik next maka akan muncul menu customize connection, konfigurasi seperti pada gambar



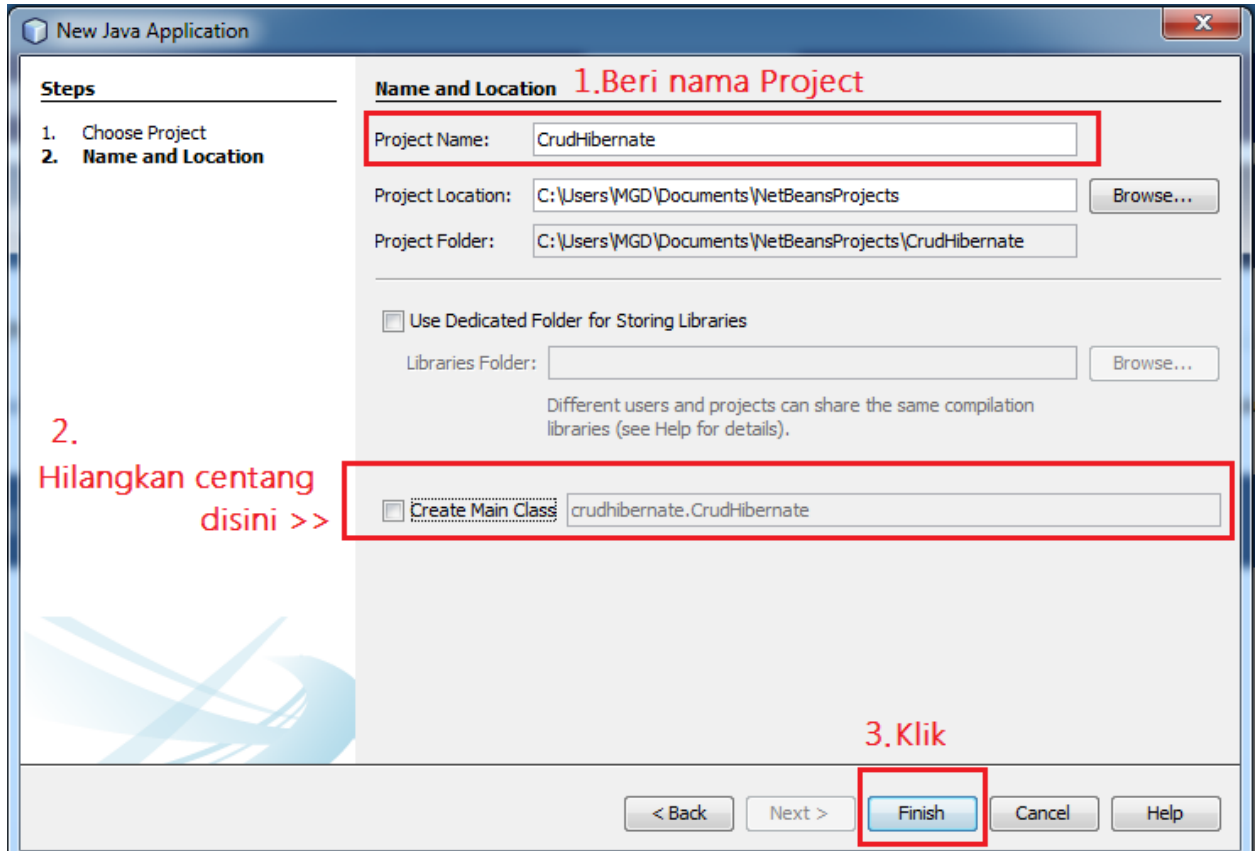
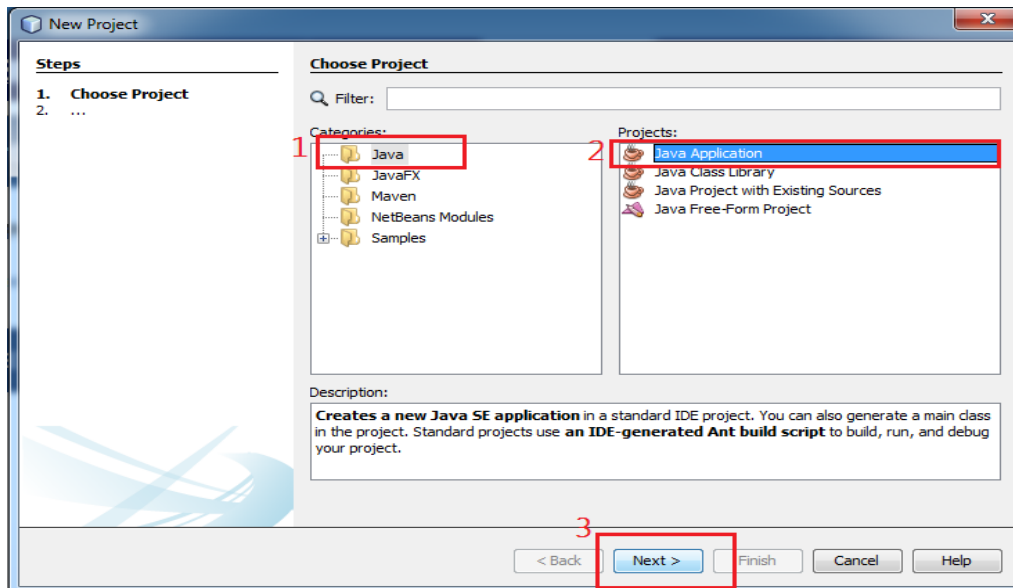
1. Kamu harus memasukan nama database, karena diawal kita membuat database dengan nama crudhibernate, maka crudhibernate lah yang kita tulis disitu.
2. Klik test Connection dan tunggu sebentar.
3. Jika muncul Connection Succeeded artinya konfigurasi yang kita lakukan benar, jika terjadi error silahkan cek kembali masalah driver database yang ada pada netbeans atau cek kembali service database yang kita miliki apakah sudah aktif atau belum.
4. Klik Next.

- Cukup klik next next dan finish, setelah itu kita telah memiliki koneksi baru



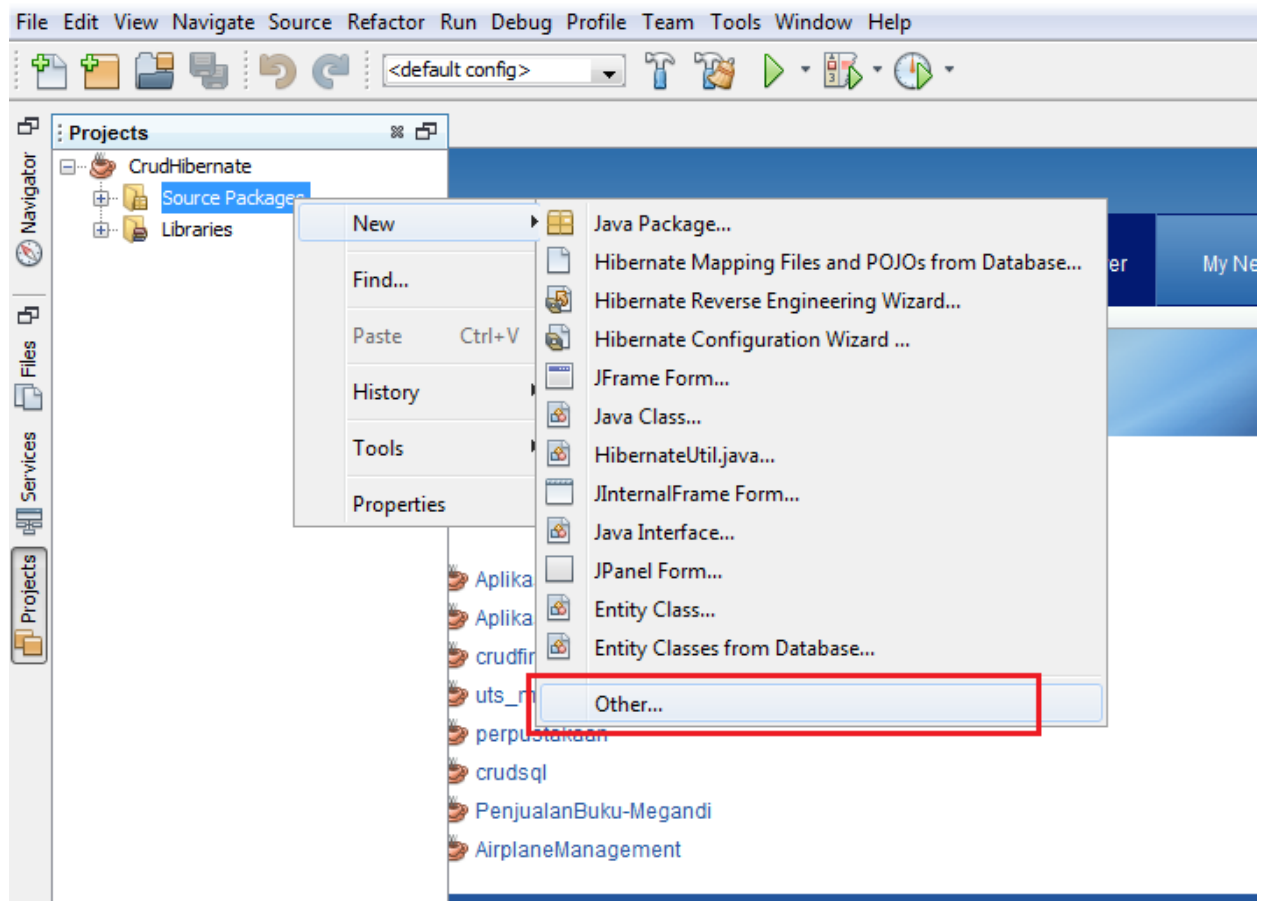
- Memulai membangun project di Netbeans

Klik File -> New Project dan pilih Java Application

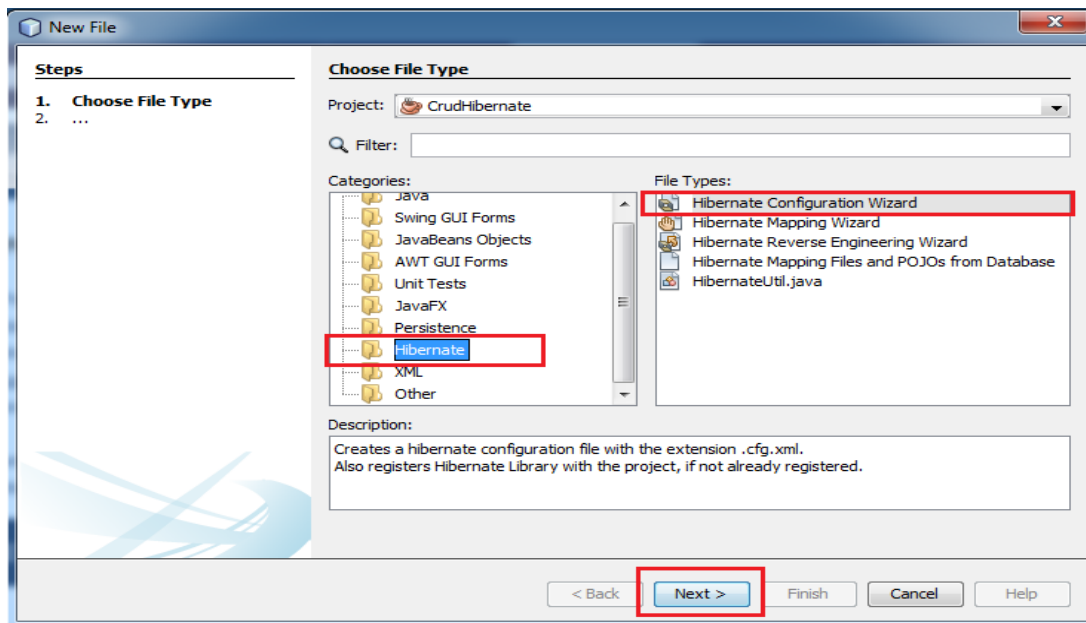


- Setelah sebelumnya kita telah membuat koneksi database ke netbeans kini kita sambungkan koneksi itu ke project kita dengan konsep hibernate.

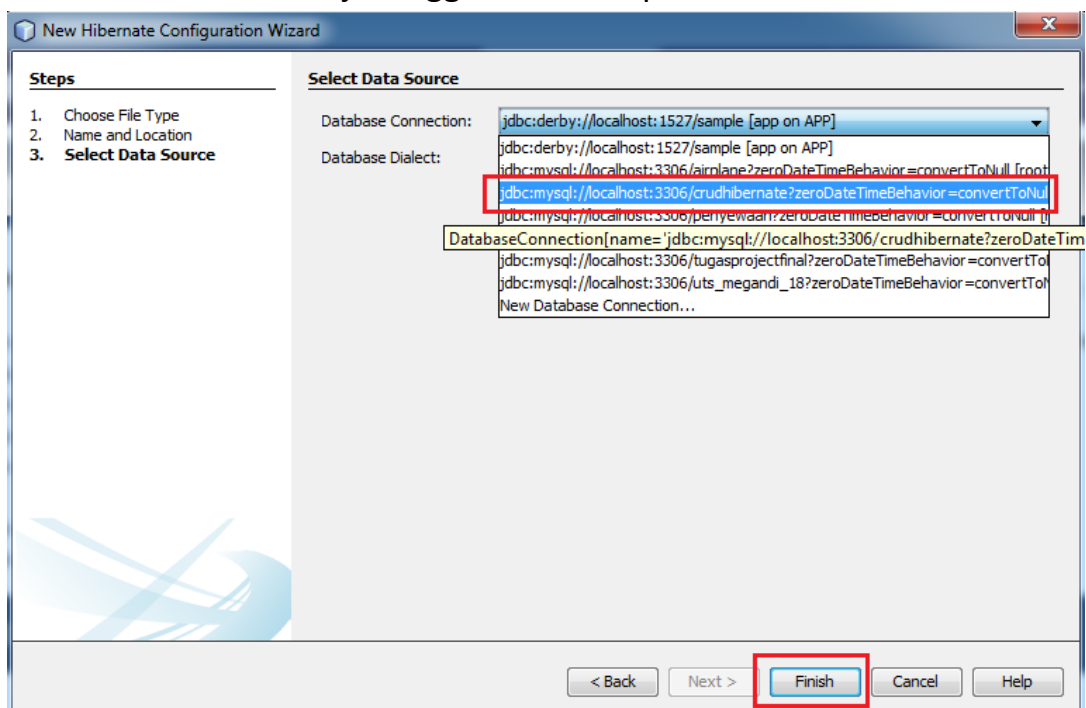
Klik kanan pada source packages lalu pilih other



Setelah itu pilih Hibernate dipilihan kategori dan Hibernate Configuration Wizard dipilihan file types.

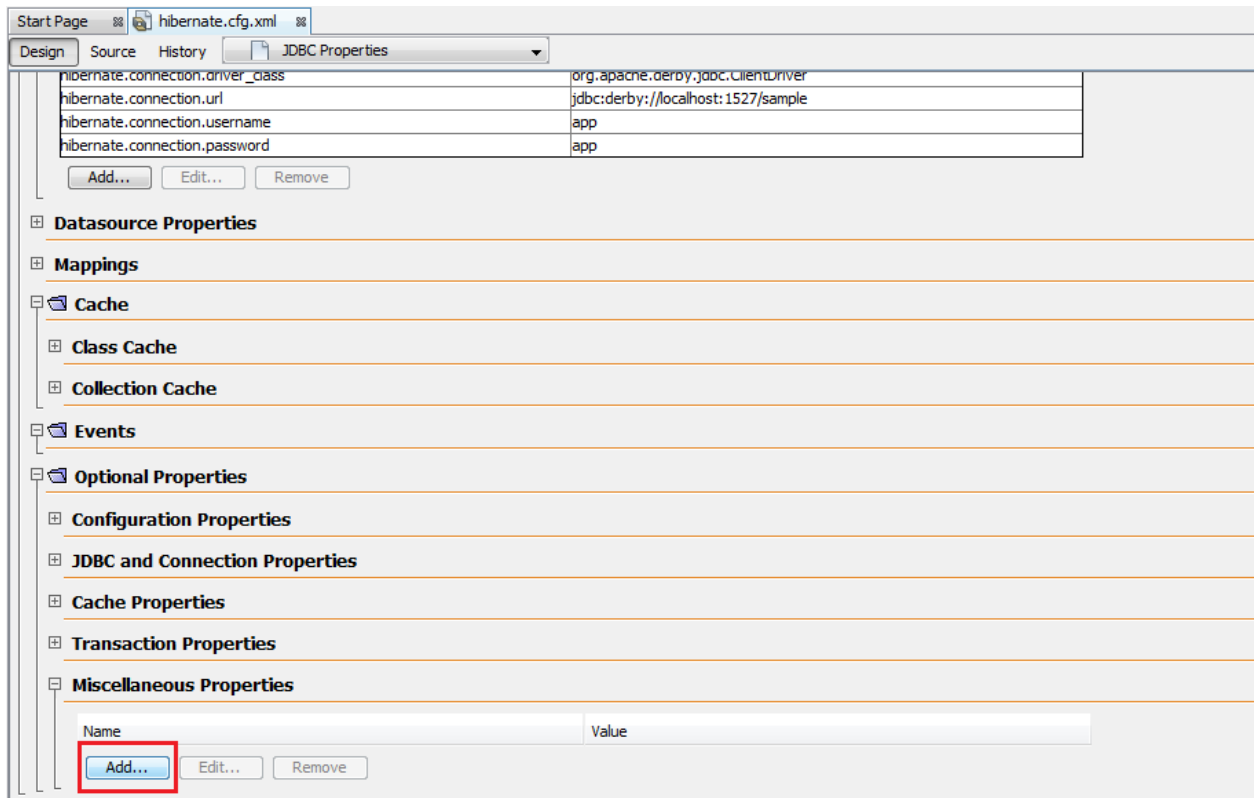


Lalu klik next – next saja hingga muncul seperti ini :

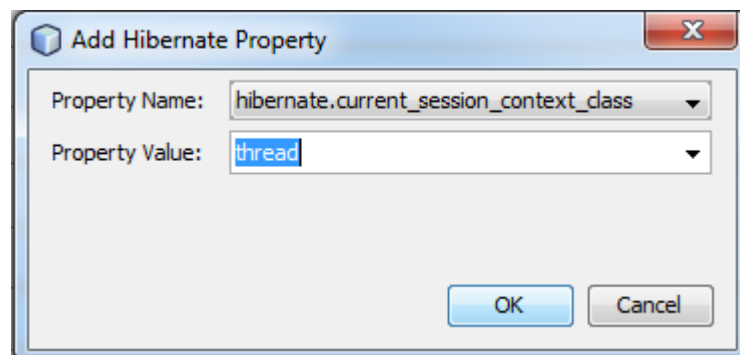


Pilih database connection yang tadi sudah kita buat yaitu `jdbc:mysql://localhost:3306/crudhibernate?zeroDate....` lalu klik finish.

Setelah itu maka akan terbuka file hibernate.cfg.xml seperti pada gambar

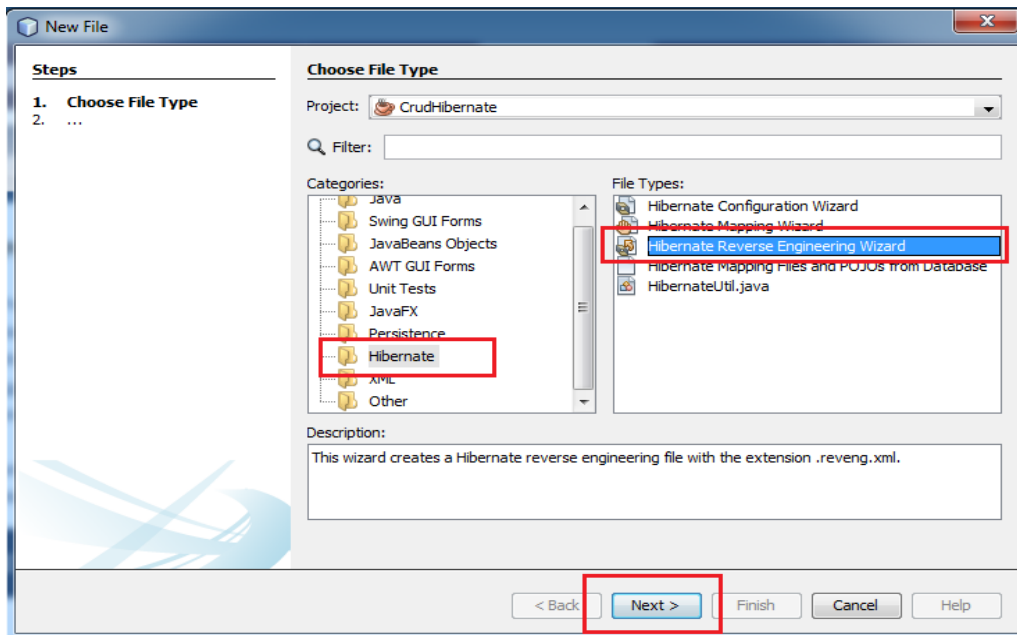


Klik add, lalu ubah property value nya menjadi thread dan klik OK. Setelah itu jangan lupa untuk menekan CTRL + S untuk menyimpannya dan close saja file hibernate.cfg.xml nya.

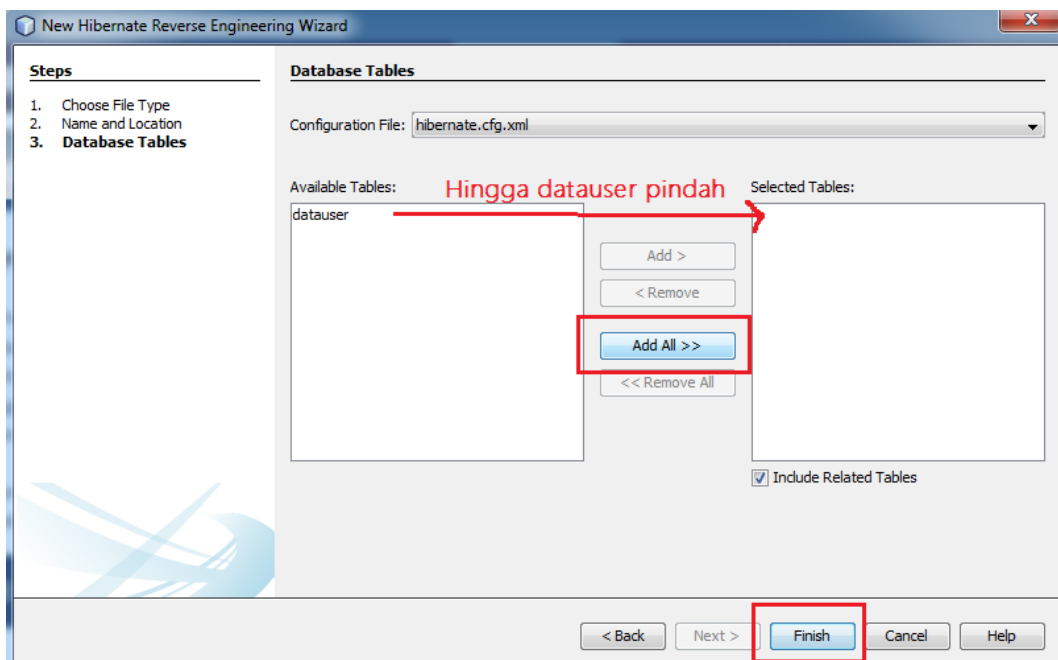


- **Menambahkan Reserve Engineering Wizard**

Lakukan yang sama seperti saat ingin menambahkan Hibernate Configuration Wizard, namun untuk langkah yang satu ini kita memilih Hibernate Reserve Engineering Wizard.

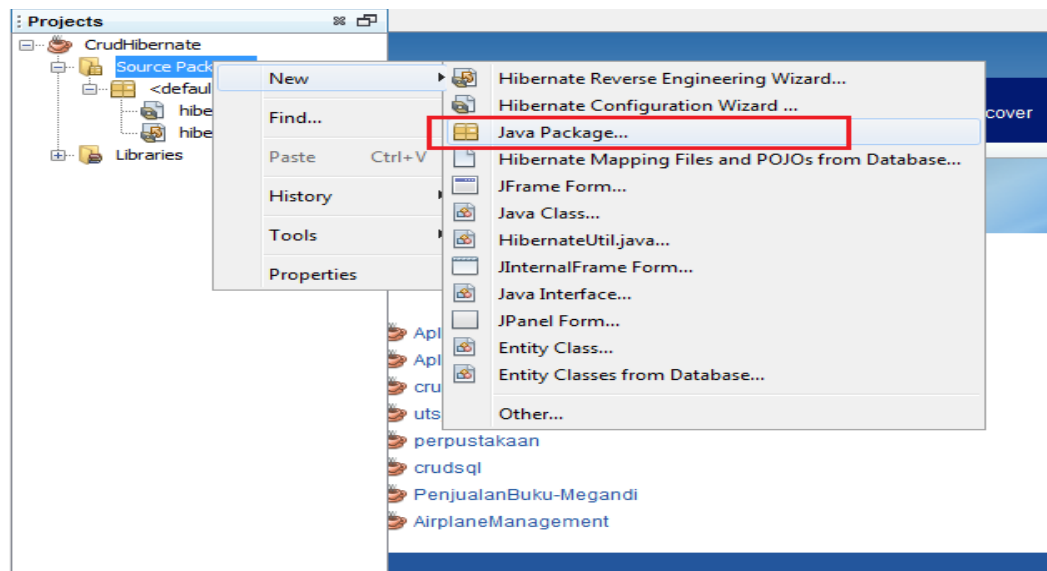


Setelah itu kita hanya perlu klik next – next saja dan pilih add all pada bagian Database Table lalu klik finish.



- Membuat packages di project

Setelah kita membuat serangkaian konfigurasi untuk membuat koneksi, maka kita bisa melihat di dalam source packages kita terdapat default packages yang terbentuk otomatis. Tetapi kita masih memerlukan beberapa packages lagi untuk membuat sebuah project/program yang menggunakan konsep MVC. Ikuti langkah pada gambar untuk membuat 1 buah packages baru dan lakukan hal yang sama untuk membuat packages lainnya.

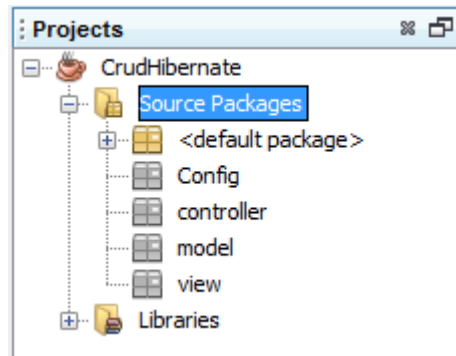


Klik kanan pada source packages -> New -> Java Packages. Jika tidak terdapat java packages anda hanya perlu memilih Other -> Java -> Java Packages.

Lalu beri nama packages dengan nama **Config**. Lakukan hal yang sama untuk membuat 3 packages lainnya dengan nama **Model**, **View** dan **Controller**.

- **Membangun masing – masing packages**

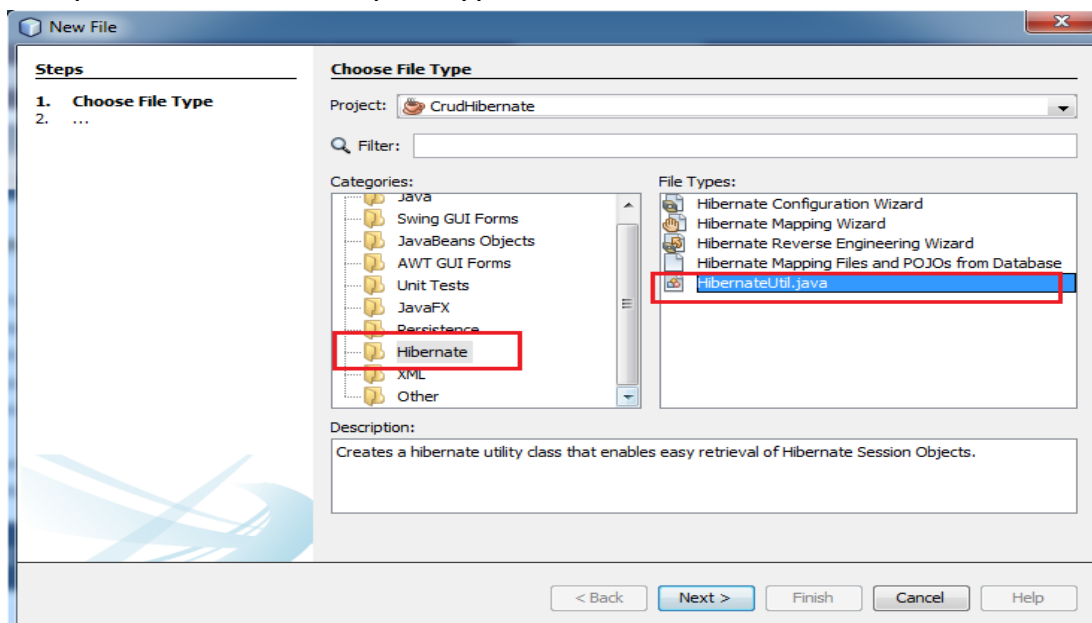
Selesai itu kini kita memiliki 4 packages, seperti pada gambar dibawah (5 jika ditambahkan dengan default) :



- **Packages Config**

Klik kanan pada Config -> New -> Other.

Lalu pilih Hibernate dan pilih type HibernateUtil

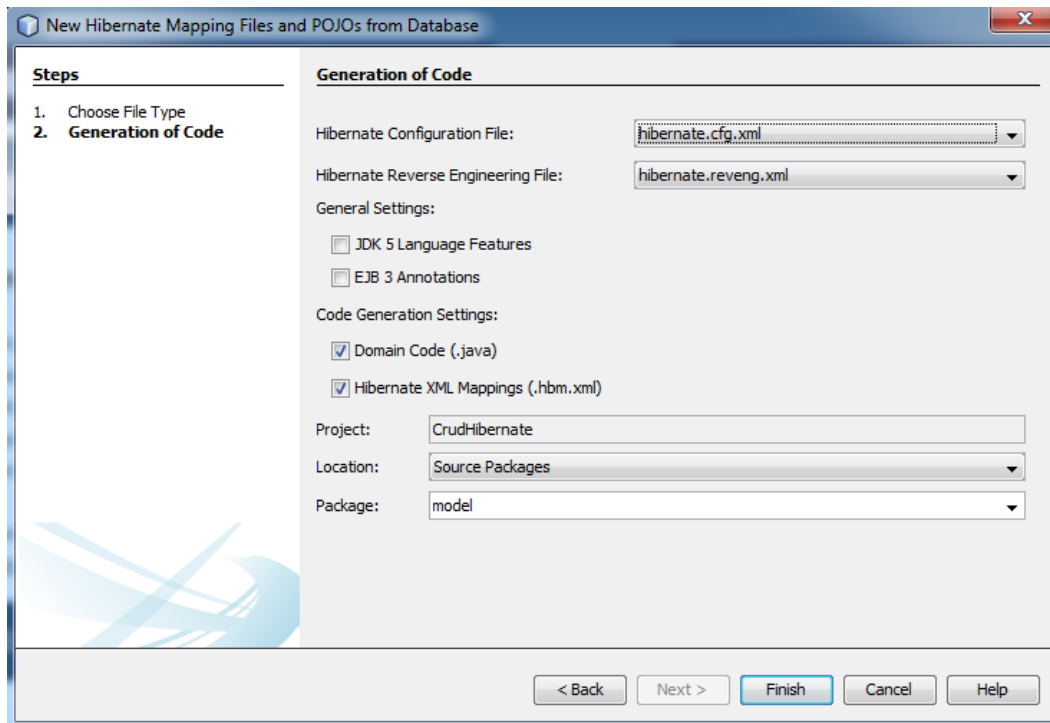


Setelah itu klik Next dan ubahlah class name menjadi HibernateUtil (Sebelumnya NewHibernateUtil), lalu klik finish.

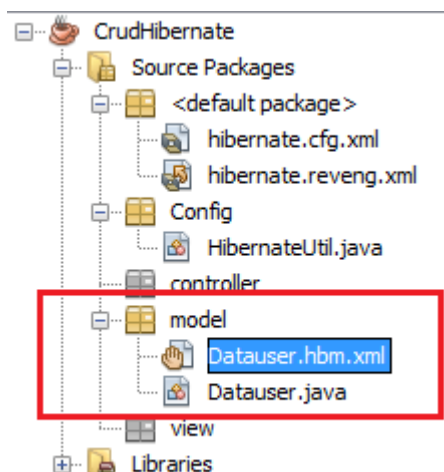
- **Packages Model**

Kembali lagi ke kategori Hibernate lalu pilih Hibernate Mapping Files and Pojos from database.

Klik next dan finish.



Setelah itu, coba cek di dalam packages model. Jika berhasil maka akan muncul Mapping file dari table yang terdapat di dalam database kita.

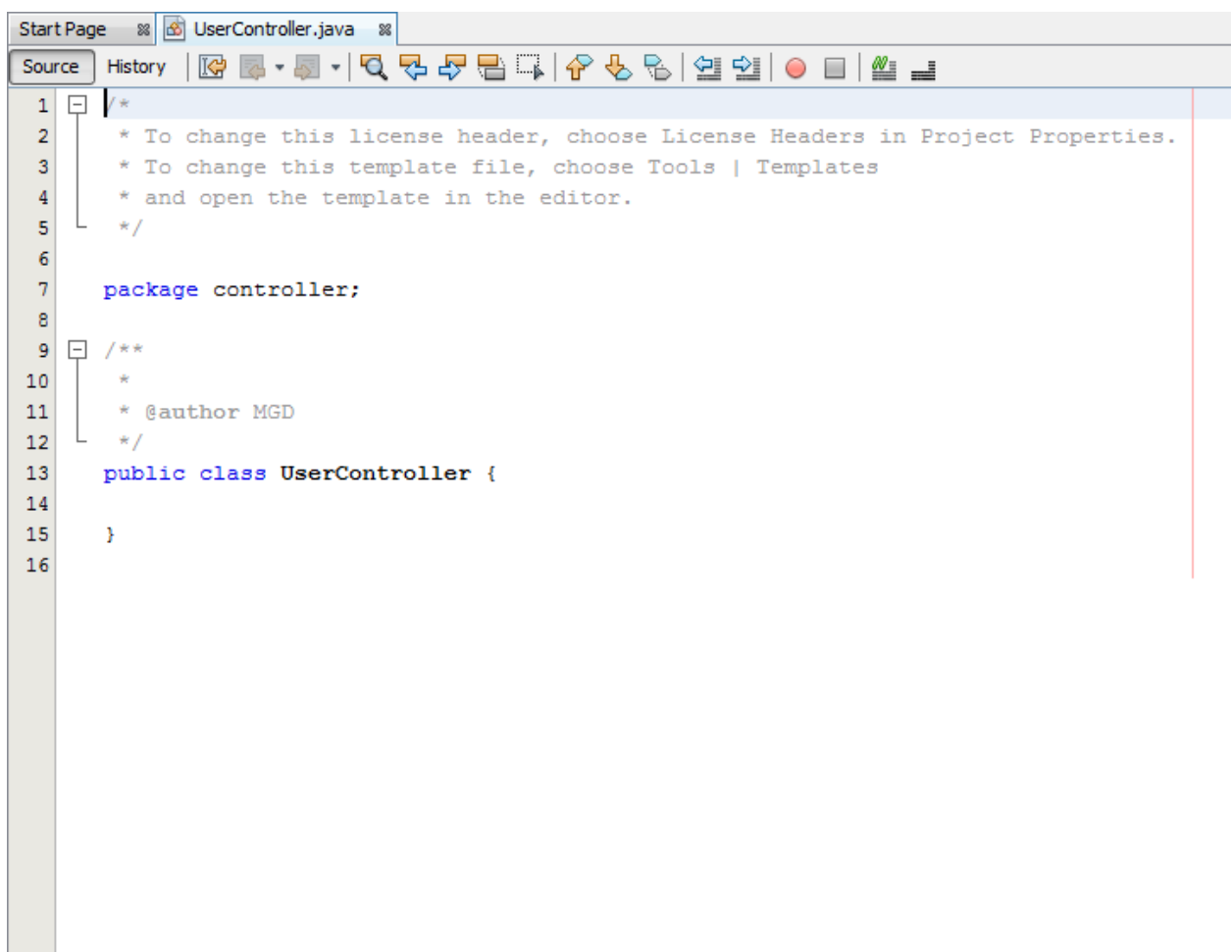


- Memulai menulis source code

- **Packages Controller**

Klik kanan pada packages Controller -> New -> Java Class.

Ubah nama class menjadi UserController. Setelah itu bukalah file UserController tersebut (biasanya sudah terbuka secara otomatis saat kita berhasil membuat class itu).



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7   package controller;
8
9   /**
10    *
11    * @author MGD
12    */
13   public class UserController {
14
15   }
16
```

Sekarang saatnya kita untuk mengetikkan Source Code pada Controller ini. Tapi yang perlu dicatat adalah bahwa Controller ini ialah yang menjadi penggerak dalam aplikasi CRUD yang kita buat dengan menggunakan konsep Hibernate MVC. Controller adalah penghubung antara View (Tampilan aplikasi/eksekusi program) dengan Model (penghubung ke database). Didalam Controller ini tentunya kita akan memiliki source code utama untuk menjalankan CRUD, yang dimana masing - masing dari *public static void* yang akan kita buat mewakili masing – masing perintah CRUD (*public static void* nya akan terdiri dari Insert, Update dan Delete).

**Catatan : Cukup ketikkan Source Code yang sama persis seperti tampak pada gambar. Yakinkan bahwa source code yang diketik benar – benar sama, perhatikan juga penggunaan huruf kapital, karena jika kamu salah dalam menuliskan penulisan capital maka itu akan fatal, kalian tahu bahwa java itu adalah bahasa yang case sensitive. Dan saya sengaja menggunakan bentuk gambar agar tidak ada sistem copy pasti disini 😊*

- Buat Public Static Void Insert dengan source code seperti pada gambar dibawah (abaikan source code di line 9 s/d 13, jangan dituliskan).

```
4 | * and open the template in the editor.
5 | */
6 |
7 | package controller;
8 |
9 | import Config.HibernateUtil;
10 | import model.Datauser;
11 | import org.hibernate.Session;
12 | import org.hibernate.SessionFactory;
13 | import org.hibernate.Transaction;
14 |
15 | /**
16 | *
17 | * @author MGD
18 | */
19 | public class UserController {
20 |     public static void insert(Datauser user) {
21 |         SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
22 |         Session session = sessionFactory.getCurrentSession();
23 |         Transaction tr = session.beginTransaction();
24 |
25 |         session.save(user);
26 |         tr.commit();
27 |     }
28 |
29 | }
30 |
```

Terjadi error atau peringatan? Tenang saja, itu memang pasti terjadi. Yang cukup kalian lakukan adalah melakukan fix imports dengan menekan CTRL + SHIFT + I secara barengan atau klik kanan mouse dan pilih fix imports. Maka peringatan akan hilang dan source code di line 9 s/d 13 akan terbentuk secara otomatis. Jika masih ada peringatan atau error silahkan di cek kembali penulisan source codenya.

Mengapa terdapat “Datauser user” didalam source code? Nah Datauser disini adalah Datauser yang kita import dari packages model (model.Datauser). Sudah tau kan sebelumnya bahwa yang terdapat didalam model itu ialah penghubung ke database (berupa sebuah mapping dari table). Maka disinilah gunanya kita panggil model.Datauser itu, untuk digerakan oleh controller ini.. lalu tentang “user” yang terdapat disamping Datauser ialah hanya sebagai perwakilan atau sebutan gampang nya “variable” dari Datauser itu.. agar dapat dieksekusi di source code bagian session.save(user);

- Dengan menerapkan peraturan yang sama seperti sebelumnya, silahkan buat source code seperti pada gambar dibawah untuk **public static void update**

```
19 public class UserController {
20     public static void insert(Datauser user){
21         SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
22         Session session = sessionFactory.getCurrentSession();
23         Transaction tr = session.beginTransaction();
24
25         session.save(user);
26         tr.commit();
27     }
28
29     public static void update(Datauser user){
30         SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
31         Session session = sessionFactory.getCurrentSession();
32         Transaction tr = session.beginTransaction();
33
34         Datauser users = (Datauser) session.load(Datauser.class, user.getId());
35         //bedakan user dengan users
36         session.update(user);
37         tr.commit();
38     }
39 }
```

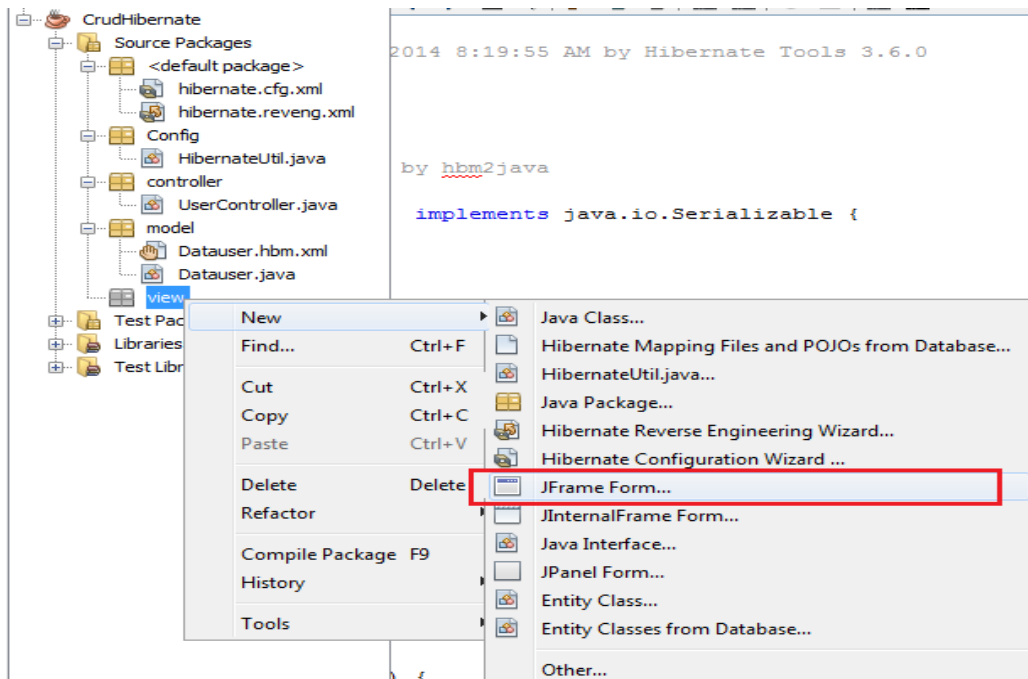
- Dengan menerapkan peraturan yang sama seperti sebelumnya, silahkan buat source code seperti pada gambar dibawah untuk **public static void Delete**

```
33
34     Datauser users = (Datauser) session.load(Datauser.class, user.getId());
35     //bedakan user dengan users
36     session.update(user);
37     tr.commit();
38 }
39
40 public static void delete(String id){
41     SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
42     Session session = sessionFactory.getCurrentSession();
43     Transaction tr = session.beginTransaction();
44
45     Datauser users = (Datauser) session.load(Datauser.class, new Integer(id));
46     session.delete(users);
47     tr.commit();
48 }
49
50 }
51
```

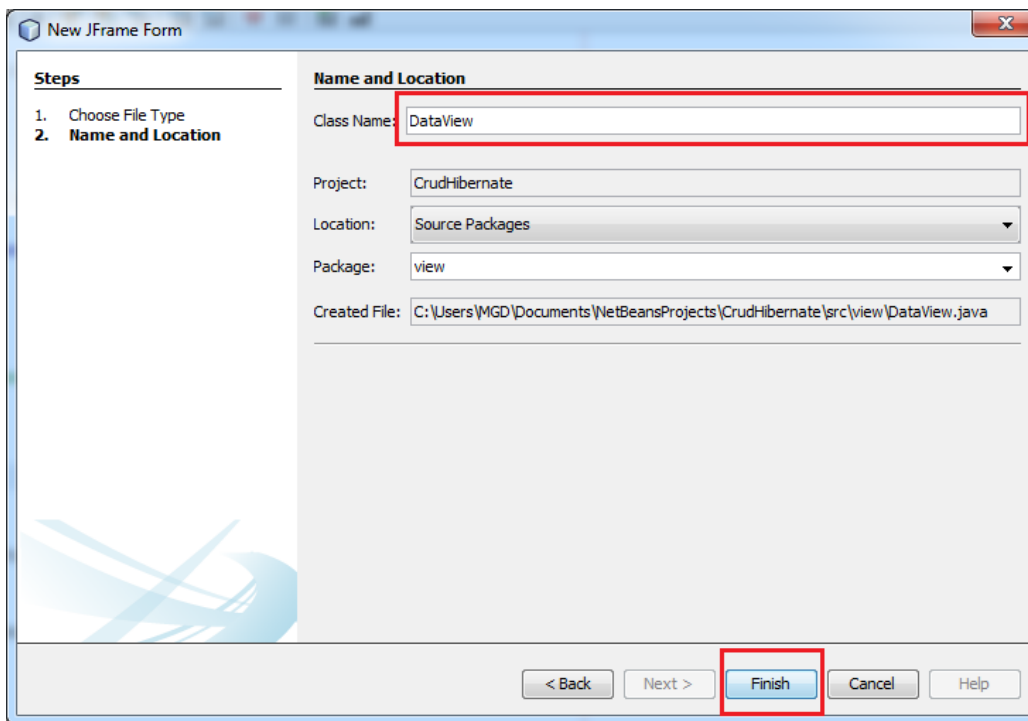
Sudah semua kan? Dari Insert, Update sampai Delete. Nah berarti tugas kita untuk membuat source code di packages *Controller* sudah selesai. Sekali lagi saya ingatkan, perhatikan penggunaan huruf capital pada source code dan jangan sampai salah menulis code untuk menghindari error dalam pembuatan source code di packages *Controller* ini dan jangan lupa untuk melakukan fix imports (CTRL + SHIFT + I). Jika tidak ada error, maka kita akan lanjut ke packages *View*.

- Packages View

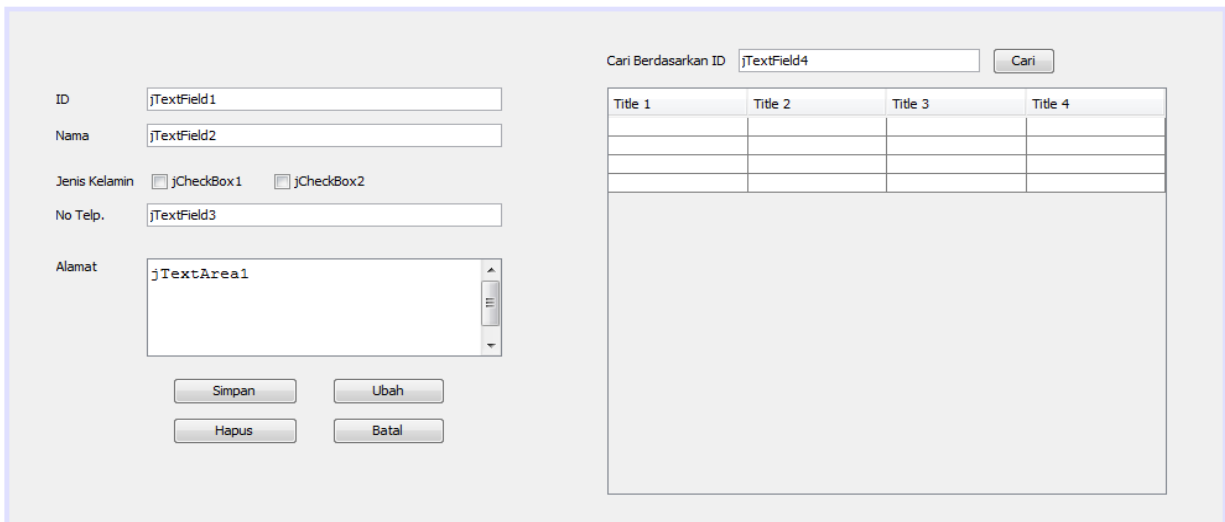
Buat sebuah JFrame baru



dan beri nama "Dataview".



- Setelah itu klik finish dan desainlah bagian JFrame tersebut hingga seperti tampak pada gambar.



Adapun komponen yang terdapat pada gambar adalah

- 3 buah jTextField
- 1 buah JTextArea
- 6 buah JLabel
- 2 buah JCheckBox
- 5 buah JButton (nama sesuai dengan gambar)
- Dan 1 buah Table

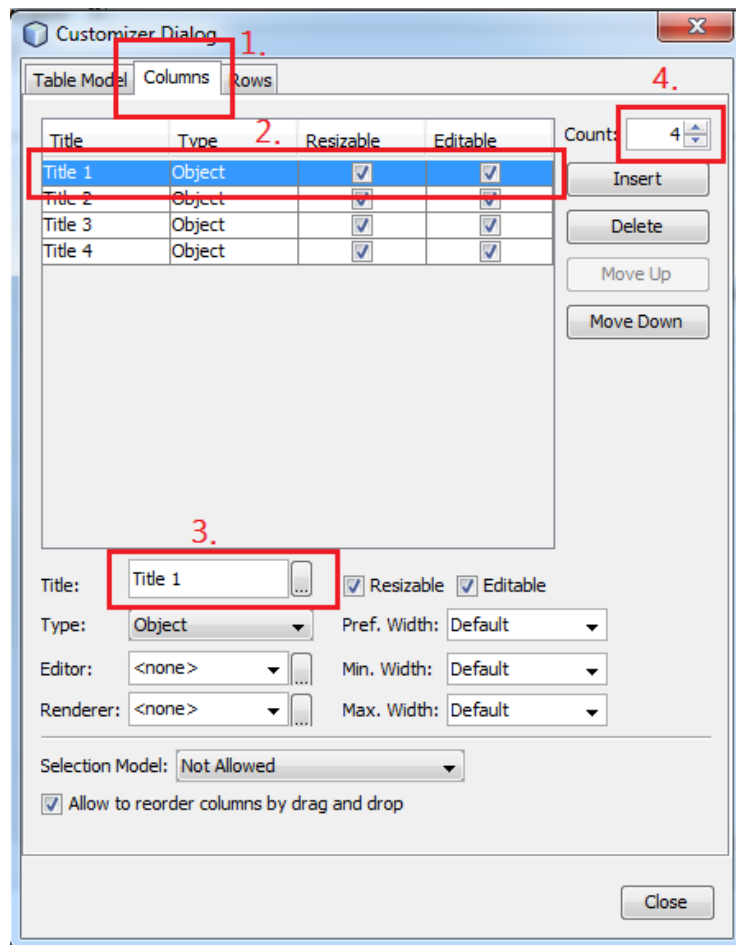
Setelah itu berilah nama variable dan text masing – masing komponen dengan klik kanan komponen -> Change Variable Name ...

Nama Komponen	Nama Text	Nama Variable
jTextField1	Kosongkan	txtid
jTextField2	Kosongkan	txtnama
jTextField3	Kosongkan	txtnotelp
jCheckBox1	Pria	cbpria
jCheckBox2	Wanita	cbwanita
JTextArea1	Kosongkan	taalamat
jTextField4	Kosongkan	txtcari
Table	-	tbluser

Ubah juga nama masing – masing dari JLabel, cukup sesuaikan dengan samping komponen yang sudah dikonfigurasi diatas..

- Konfigurasi Komponen Table

Jangan lupa untuk mengubah variable name table menjadi *tbluser*, lalu Klik kanan pada Table, pilih Table Contents dan pilih Columns



Keterangan :

1. Pilih tab Columns,
2. Klik pada row,
3. Ubah Title nya (Sesuaikan),
4. Naikan jumlah columns menjadi 5 (Karena jumlah field yg kita miliki 5).

Lakukan itu semua terhadap semua row (kecuali langkah ke 4). Hingga table menjadi seperti gambar dibawah ini.

The screenshot shows a Java Swing window with a search bar at the top labeled "Cari Berdasarkan ID" and a "Cari" button. Below the search bar is a table with five columns: "ID", "Nama", "Jenis Kelamin", "No Telp", and "Alamat". The table has three empty rows. Below the table is a large empty rectangular area. The window has a standard Mac OS X-style title bar and a scroll bar on the right side.

ID	Nama	Jenis Kelamin	No Telp	Alamat

- Konfigurasi Komponen jCheckBox

Tambahkan komponen Button Group yang ada di menu palette, letakan komponen itu secara bebas dimana saja..

Sekarang kelompokkan 2 jCheckBox yang ada (cbpria & cbwanita) ke dalam satu button group tersebut, dengan cara klik kanan di jCheckBox -> Properties, pilih buttongroup -> buttonGroup1.

- Mengetik Source Code

Pertama buat sebuah string baru guna mempermudah kita dalam memilih data terpilih. Buatlah string “datauserterpilih” diatas source code public void DataView(){}

```
23 public class DataView extends javax.swing.JFrame {
24
25     String datauserterpilih;
26     public DataView() {
27         initComponents();
28     }
29 }
```

- Source Code untuk menampilkan data

Pilih *Source View* pada project



Ketikan Source Code tepat dibawah tulisan Generate Code atau pada baris ke 213. Hapus saja source ** @param args the command line arguments*

```
13 public class DataView extends javax.swing.JFrame {
14
15
16
17     String datauserterpilih;
18     public DataView() {
19         initComponents();
20     }
21
22     /**
23      * This method is called from within the constructor to initialize the form.
24      * WARNING: Do NOT modify this code. The content of this method is always
25      * regenerated by the Form Editor.
26      */
27     @SuppressWarnings("unchecked")
28     Generated Code
213
214     /**
215      * @param args the command line arguments
216      */
217     public static void main(String args[]) {
218         /* Set the Nimbus look and feel */
219         Look and feel setting code (optional)
220
221         /* Create and display the form */
222         java.awt.EventQueue.invokeLater(new Runnable() {
```

Hapus jika mengganggu

Persiapkan untuk mengetik Source Code disini

Hapus saja.

Ketikan Source Code sesuai pada gambar dibawah. Sebelumnya, pastikan variable name yang kamu buat mengikuti informasi yang terdapat di halaman 24. Jika kamu berkreasi sendiri tidak masalah, hanya saja kamu perlu menyesuaikannya.

```
228
229 public void tampilandata(){
230     DefaultTableModel model = (DefaultTableModel) tbluser.getModel();
231     model.setNumRows(0);
232
233     SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
234     Session session = sessionFactory.openSession();
235     Transaction tr = session.beginTransaction();
236
237     Query query = session.createQuery("from Datauser");
238     List result = query.list();
239     Iterator it = result.iterator();
240
241     while (it.hasNext()){
242         Datauser user = (Datauser) it.next();
243         model.addRow(new Object []{
244             user.getId(),
245             user.getNama(),
246             user.getJenisKelamin(),
247             user.getNoTelp(),
248             user.getAlamat()
249         });
250     }
251
252 }
```

ketikan dengan teliti source code diatas, ketik saja sampai selesai dengan menghiraukan peringatan – peringatan yang ada. Setelah source code selesai diketikan hingga mirip 100%, lakukan seperti pada langkah sebelumnya yaitu tekan CTRL + SHIFT + I untuk melakukan fix imports..

Sekarang kita telah memiliki public void dengan nama “tampilandata()”, gunakan tampilandata() saat melakukan refresh / saat program pertama kali dijalankan dengan cara menaruh source code tampilandata() dibawah initComponents() atau didalam public class JFrame tersebut. Lihatlah gambar :

```
23 public class DataView extends javax.swing.JFrame {
24
25     /**
26      * Creates new form DataView
27      */
28     public DataView() {
29         initComponents();
30         tampilandata();
31     }
32
33
34     @SuppressWarnings("unchecked")
35     Generated Code
218
219     public void tampilandata() {
```

Saat public void tampilandata() kita taruh di bawah initComponents() maka secara otomatis saat program dijalankan source code pada tampilandata() akan dijalankan juga.

- **Source code untuk menampilkan data terpilih**

Tentunya kita sebagai pembuat aplikasi harus membuat sebuah aplikasi yang dapat mempermudah user dalam menggunakannya. Nah dalam menu yang satu ini, aplikasi akan memberikan kemudahan terhadap user. Saat user memilih salah satu row didalam table, data user akan tampil dengan otomatis untuk diolah agar dapat di ubah ataupun dihapus.

Klik kanan pada komponen Table -> Events -> Mouse -> mouseClicked. Ketikan source code seperti dibawah ini :

```

215 private void tbluserMouseClicked(java.awt.event.MouseEvent evt) {
216     DefaultTableModel model = (DefaultTableModel) tbluser.getModel();
217     datauserterpilih = model.getValueAt(tbluser.getSelectedRow(), 0).toString();
218     txtid.setText(model.getValueAt(tbluser.getSelectedRow(), 0).toString());
219     txtnama.setText(model.getValueAt(tbluser.getSelectedRow(), 1).toString());
220     txtnotelp.setText(model.getValueAt(tbluser.getSelectedRow(), 3).toString());
221     txtalamat.setText(model.getValueAt(tbluser.getSelectedRow(), 4).toString());
222     if (model.getValueAt(tbluser.getSelectedRow(), 2).equals("Pria")){
223         cbpria.setSelected(true);
224     } else if (model.getValueAt(tbluser.getSelectedRow(), 2).equals("Wanita")){
225         cbwanita.setSelected(true);
226     }
227 }

```

Seperti biasa, jangan lupa untuk memperhatikan penggunaan huruf capital dalam source code. Program saat dijalankan :

Cari Berdasarkan ID Cari

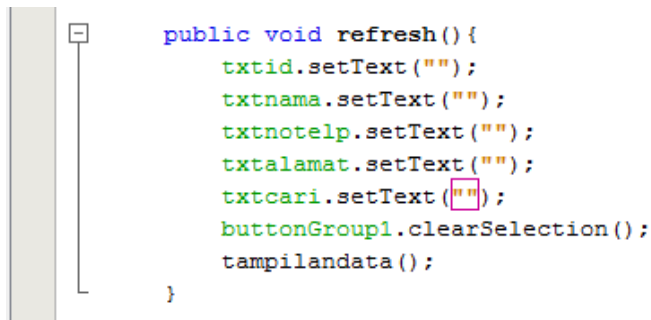
ID:
 Nama:
 Jenis Kelamin: ☒ Pria ☐ Wanita
 No Telp.:
 Alamat:

ID	Nama	Jenis Kelamin	No Telp	Alamat
1	megandi	Pria	0812345678	Jakarta

Klik row data yang ada, maka data akan muncul di form pengolahan data

- **Source code untuk refresh**

Buatlah sebuah public void dengan nama refresh() letakan source code seperti sebelumnya, yaitu dibawah Generated code. Ketikan Source code seperti gambar :

A screenshot of a code editor window. On the left, there is a vertical toolbar with a small square icon at the top and a vertical line with a horizontal tick at the bottom. The main area of the editor contains the following Java code:

```
public void refresh() {  
    txtid.setText("");  
    txtnama.setText("");  
    txtnotelp.setText("");  
    txtalamat.setText("");  
    txtcari.setText("");  
    buttonGroup1.clearSelection();  
    tampilanda();  
}
```

Public void refresh ini berguna untuk meload dan membersihkan txtfield dari text – text yang tersisa.

- **Source Code Menambahkan Data**

Pada Button *Simpan* ketikan source code seperti gambar dibawah ini :

```
247
248 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
249     Datauser user = new Datauser();
250     user.setId(new Integer(txtid.getText()));
251     user.setNama(txtnama.getText());
252     user.setNoTelp(txtnotelp.getText());
253     user.setAlamat(txtalamat.getText());
254     if (cbpria.isSelected()){
255         user.setJenisKelamin("Pria");
256     } else if (cbwanita.isSelected()){
257         user.setJenisKelamin("Wanita");
258     }
259     UserController.insert(user);
260     JOptionPane.showMessageDialog(null, "Anda berhasil menambahkan data",
261     "Informasi", JOptionPane.INFORMATION_MESSAGE);
262     refresh(); // fungsi refresh yang sebelumnya sudah dibuat
263 }
264
```

Jangan lupa kan peran controller disini pada baris ke 259.

- **Source Code Mengubah Data**

Pada Button *Ubah* ketikkan source code seperti gambar dibawah ini :

```
private void BtnubahActionPerformed(java.awt.event.ActionEvent evt) {  
    Datauser user = new Datauser();  
    user.setId(new Integer(datauserterpilih));  
    user.setNama(txtnama.getText());  
    user.setNoTelp(txtnotelp.getText());  
    user.setAlamat(txtalamat.getText());  
    if (cbpria.isSelected()){  
        user.setJenisKelamin("Pria");  
    } else if (cbwanita.isSelected()){  
        user.setJenisKelamin("Wanita");  
    }  
    UserController.update(user);  
    JOptionPane.showMessageDialog(null, "Anda berhasil mengubah data",  
        "Informasi", JOptionPane.INFORMATION_MESSAGE);  
    refresh(); // fungsi refresh yang sebelumnya sudah dibuat  
}
```

- **Source Code Menghapus Data**

Pada Button *Hapus* ketikkan source code seperti gambar dibawah ini :

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    UserController.delete (datauserterpilih);  
    JOptionPane.showMessageDialog (null, "Anda berhasil menghapus data",  
    "Informasi", JOptionPane.INFORMATION_MESSAGE);  
    refresh();  
}
```

- **Source Code Untuk Refresh / Batal**

Pada Button *Batal* hanya cukup ketikkan source code seperti gambar dibawah ini :

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    refresh();  
}
```

Pada Button *batal* kita hanya memerlukan pemanggilan public void yang sudah kita buat tadi, yaitu refresh();

- **Source Code Untuk Mencari Data**

Pada Button *Cari* ketikkan source code seperti gambar dibawah ini :

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    DefaultTableModel model = (DefaultTableModel) tbluser.getModel();  
    model.setNumRows(0);  
  
    SessionFactory sessionFactory = HibernateUtil.getSessionFactory();  
    Session session = sessionFactory.openSession();  
    Transaction tr = session.beginTransaction();  
  
    Query query = session.createQuery("from Datauser u where u.id='"+txtcari.getText()+"'");  
    List result = query.list();  
    Iterator it = result.iterator();  
  
    if (it.hasNext()){  
        Datauser user = (Datauser) it.next();  
        model.addRow(new Object[]{  
            user.getId(),  
            user.getNama(),  
            user.getJenisKelamin(),  
            user.getNoTelp(),  
            user.getAlamat()  
        });  
        txtid.setText(String.valueOf(user.getId()));  
        txtnama.setText(user.getNama());  
        txtnotelp.setText(user.getNoTelp());  
        txtalamat.setText(user.getAlamat());  
        if (user.getJenisKelamin().equals("Pria")){  
            cbpria.setSelected(true);  
        } else {  
            cbwanita.setSelected(true);  
        }  
    } else {  
        JOptionPane.showMessageDialog(null, "ID Tidak ada",  
            "Informasi", JOptionPane.ERROR_MESSAGE);  
        refresh();  
    }  
}
```

- Menjalankan Aplikasi

Setelah semua langkah – langkah diatas dilakukan dengan benar, sekarang saatnya kamu untuk mencoba menjalan aplikasi. Klik kanan pada source code (dimana saja) -> Run File atau dengan menekan Shift + F6.

The screenshot shows a Java Swing application window with a light gray background. On the left side, there is a form with the following fields and controls:

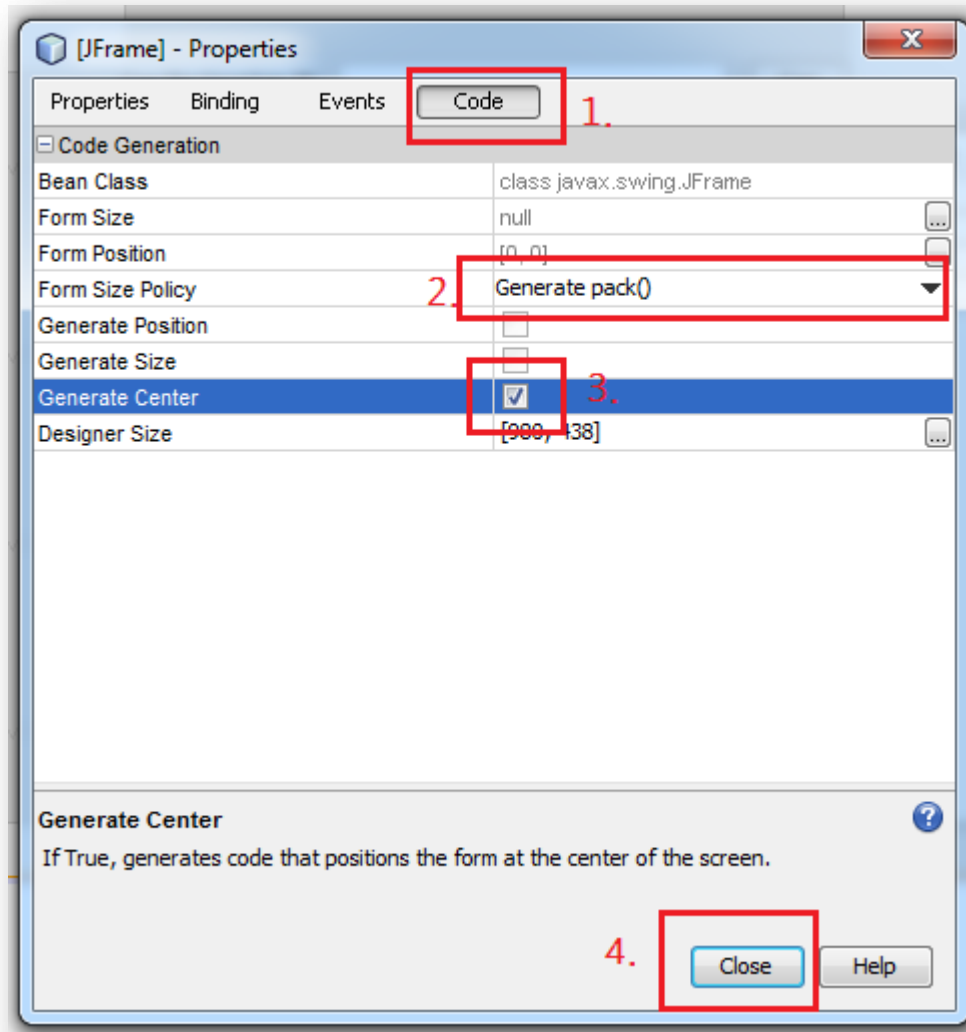
- ID**: A text input field.
- Nama**: A text input field.
- Jenis Kelamin**: Two radio buttons labeled **Pria** and **Wanita**.
- No Telp.**: A text input field.
- Alamat**: A text area with a vertical scrollbar.
- At the bottom of the form are four buttons: **Simpan**, **Ubah**, **Hapus**, and **Batal**.

On the right side of the window, there is a search section and a table:

- Search Section**: Labeled "Cari Berdasarkan ID", it includes a text input field and a **Cari** button.
- Table**: A table with 5 columns: **ID**, **Nama**, **Jenis Kelamin**, **No Telp**, and **Alamat**. It contains one data row:

ID	Nama	Jenis Kelamin	No Telp	Alamat
1	Megandi	Pria	08123456	Jakarta

Aplikasi berjalan berada di pinggir desktop? Tenang saja. Kamu bisa mengubah posisinya menjadi berada di tengah desktop dengan cara yang sangat mudah. Klik Form (di Design) lalu pilih properties dan pilih tab code, lihatlah gambar :



Keterangan :

- Pilih tab code,
- Pilih Generate Pack() di Form Size Policy,
- Ceklis atau Centang Generate Center,
- Klik Close.

- Penutup

Apabila kamu mengikuti tutorial dengan benar dan aplikasi dapat berjalan dengan baik, maka selamat kamu telah berhasil membuat sebuah aplikasi pengolahan data sederhana menggunakan bahasa Java melalui metode/framework Hibernate MVC ☺

Jika ada yang ingin ditanyakan atau sekedar memberi masukan / saran, silahkan saja langsung menghubungi saya melalui alamat email saya : me.gandi@rocketmail.com. Semoga Modul Tutorial ini dapat bermanfaat bagi kita semua, Terima Kasih.

Best Regards,

Megandi