

Tutorial Membuat Program CRUD dengan Socket Programming (Protokol TCP)

Bismillaahir-rohmaanir-rohiim.

Pada dokumen ini saya akan membuat tutorial membuat program CRUD dengan menggunakan socket programming. Semoga tutorial ini dapat membantu dan memberikan pemahaman kepada para pembaca. Aamiin.

Dalam program ini kita akan membuat dua buah program inti, yaitu program untuk server dan program untuk client. Dimana program client berfungsi sebagai user interface dan menerima data inputan, yang selanjutnya data yang masuk tersebut diubah kedalam bentuk Query SQL dan dikirimkan kepada server. Sedangkan program server berfungsi untuk menerima 'permintaan' dari client dan menjalankan Query tersebut.

Sebelumnya ada beberapa hal yang perlu dipersiapkan terlebih dahulu, yaitu sebuah laptop yang di dalamnya terinstal:

- Netbeans
- Dan XAMPP

Note :

Dalam tutorial ini saya akan membuat **program CRUD untuk Data Mahasiswa**. Silahkan kalian sesuaikan program ini dengan program yang akan kalian buat.

Index Tutorial

Langkah demi langkah	2
#Persiapan Awal	2
#Class Config : Membuat konfigurasi untuk mengkoneksikan program dengan database	4
#Class Server : Membuat Class Server TCP.....	4
#Jframe Client : Membuat Tampilan Jframe Form Client	6
#Kode Program Jframe Form Client	8
Cara Menjalankan Program	11

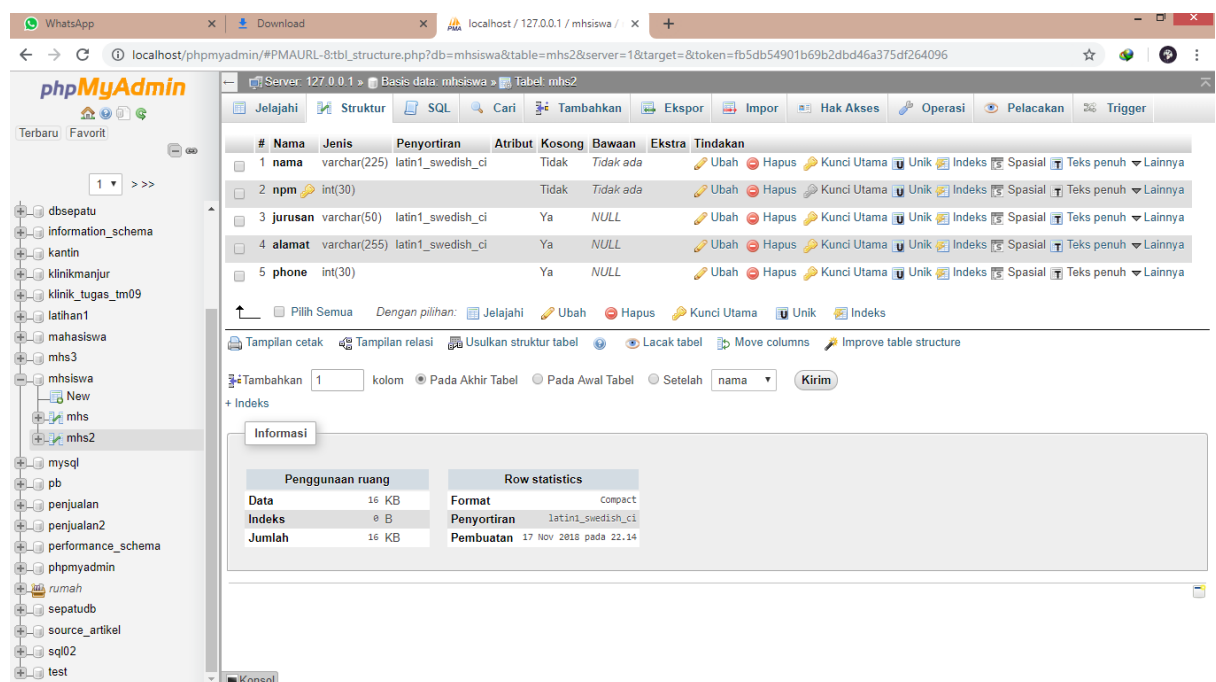
NB: Ctrl + Klik untuk langsung menuju halaman yang dituju.

Langkah demi langkah

#Persiapan Awal

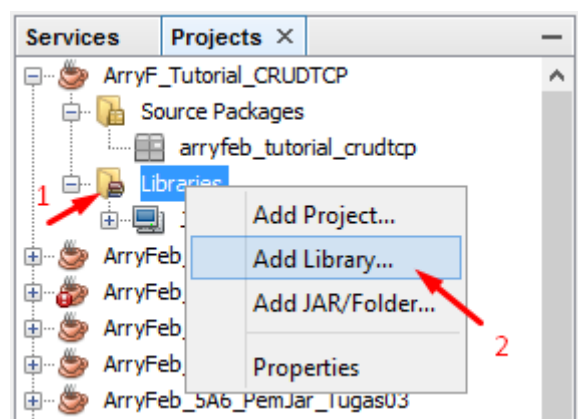
1. Langkah pertama, buat/siapkan terlebih dahulu sebuah database beserta tabel dan elemen-elemennya, nantinya database ini akan digunakan untuk menyimpan data data yang akan di-input. Silahkan kalian sesuaikan database kalian dengan program yang akan kalian buat.

Di sini saya membuat database dengan nama **mhsiswa** beserta struktur tabel sebagai berikut:

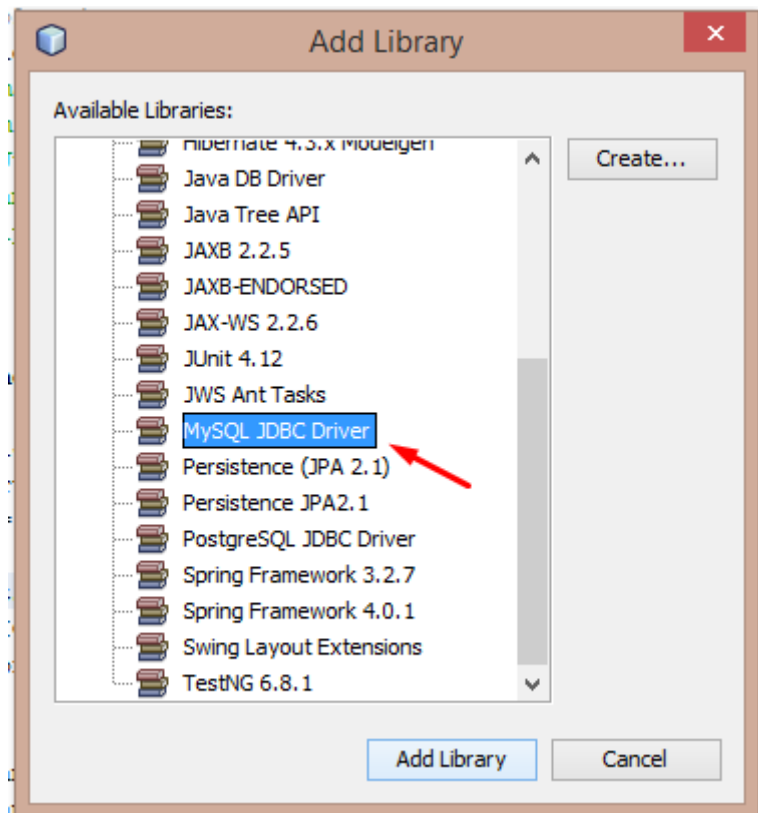


2. Selanjutnya buka aplikasi netbeans dan buatlah sebuah project baru.

Kemudian tambahkan library MySQL JDBC dengan cara klik kanan pada Libraries > Add Library

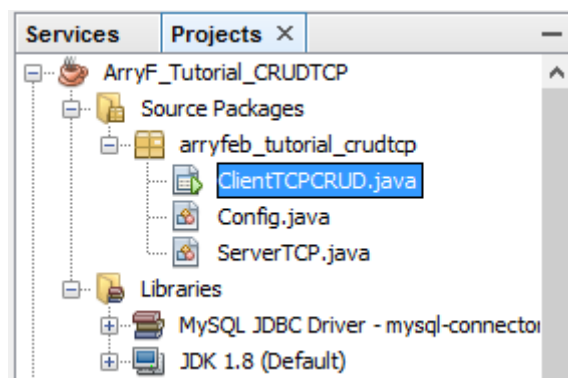


Pilih MySQL JDBC Driver, dan klik Add Library.



3. Langkah selanjutnya, siapkan terlebih dahulu 2 buah class dan 1 buah JFrame Form dimana ketiga class tersebut adalah :

- Class Config, berfungsi untuk membuat koneksi dengan database.
- Class Server, berfungsi untuk menerima dan menjalankan permintaan client.
- JFrame Form Client, berfungsi sebagai user interface serta menerima data inputan.



#Class Config : Membuat konfigurasi untuk mengkoneksikan program dengan database

4. Selanjutnya, bukalah Class Config dan ikuti kode program dibawah ini.

Kode Program Class Config :

```
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  package arrayfeb_tutorial_crudtcp;
7
8  import java.sql.Connection;
9  import java.sql.DriverManager;
10 import java.sql.SQLException;
11
12 /**
13 *
14 * @author ArrayFebryan
15 */
16 public class Config {
17     private static Connection mysqlconfig;
18     public static Connection configDB() throws SQLException{
19         try {
20             String url="jdbc:mysql://localhost:3306/mhsiswa"; //url database
21             String user="root"; //user database
22             String pass=""; //password database
23             DriverManager.registerDriver(new com.mysql.jdbc.Driver());
24             mysqlconfig=DriverManager.getConnection(url, user, pass);
25         } catch (Exception e) {
26             System.err.println("koneksi gagal " +e.getMessage()); //perintah menampilkan error pada koneksi
27         }
28         return mysqlconfig;
29     }
30 }
31
```

Edit bagian ini sesuai dengan nama database yang kalian buat

#Class Server : Membuat Class Server TCP

5. Langkah berikutnya, bukalah Class Server dan ikuti kode program dibawah ini.

Kode Program Class Server :

```
1  package arrayfeb_tutorial_crudtcp;
2
3  import java.io.BufferedReader;
4  import java.io.IOException;
5  import java.io.InputStream;
6  import java.io.InputStreamReader;
7  import java.io.OutputStream;
8  import java.io.PrintWriter;
9  import java.net.ServerSocket;
10 import java.net.Socket;
11 import java.sql.Connection;
12 import java.sql.SQLException;
13
14 /**
15 *
16 * @author ArrayFebryan
17 */
18 public class ServerTCP {
19
20     public static void main(String args[]) throws IOException{
21         System.out.print("=====
22             + "=====
23             + "\t\tTCP Server - Array Febryan
24             + "=====
25
26         //Membuat jalur komunikasi
27         ServerSocket serverSocket = null;
28         int port = 8000;
29         try{
30             serverSocket = new ServerSocket(port);
31             System.out.println("[TCP server] Jalur komunikasi berhasil dibuat di port = "+port);
32         }catch(IOException e){
33             System.out.println("Error: Server tidak dapat membuat port = "+port);
34         }
35
36         while(true){
37             Socket clientSocket = null;
38
39             //Menunggu request client
40
```

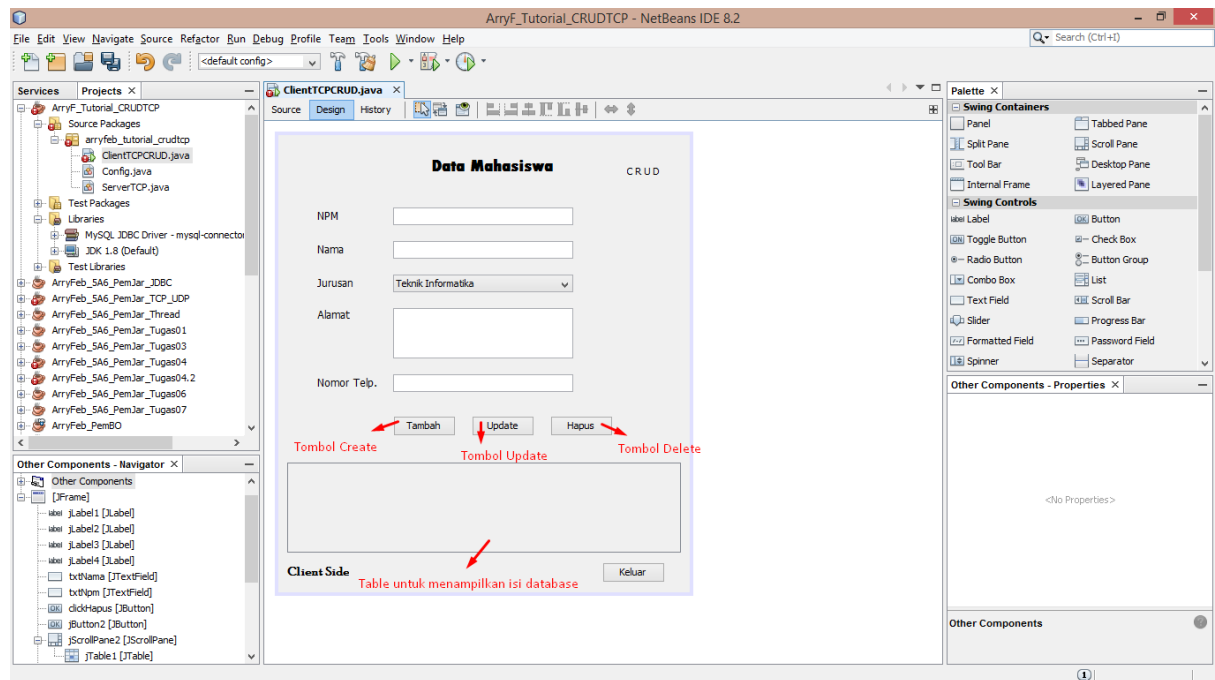
```

41     try {
42         System.out.println("\n[TCP Server] Menunggu permintaan dari client ...");
43         clientSocket = serverSocket.accept();
44     } catch (IOException e) {
45         System.out.println("Error: Tidak dapat menerima permintaan client.. Keluar Program");
46         return;
47     }
48
49     //Memproses permintaan client
50     try {
51         processClientRequest(clientSocket);
52     } catch (IOException e) {
53         System.out.println("Error: Tidak dapat menerima permintaan client..");
54     }
55 }
56
57 //Method untuk memproses data yang dikirim oleh client
58 public static void processClientRequest(Socket clientSocket) throws IOException {
59     System.out.println("[TCP Server] Memproses permintaan yang masuk..");
60     try {
61         OutputStream output = clientSocket.getOutputStream();
62         InputStream input = clientSocket.getInputStream();
63
64         PrintStream printStream = new PrintStream(output);
65         InputStreamReader inputStream = new InputStreamReader(input);
66
67         BufferedReader bufferedReader = new BufferedReader(inputStream);
68
69         String message = null;
70         message = bufferedReader.readLine();
71         System.out.println("Permintaan telah diterima : "+message);
72
73         ////Message incoming
74         try{
75             String sql = message;
76             java.sql.Connection conn=(Connection)Config.configDB();
77             java.sql.PreparedStatement pst = conn.prepareStatement(sql);
78             pst.execute();
79
80             System.out.println("\tPermintaan berhasil diproses..");
81
82             String messageSend = "Permintaan berhasil diproses";
83             printStream.println(messageSend);
84
85         }catch (SQLException e){
86             System.out.println("\tError: Permintaan tidak dapat diproses..");
87
88             String messageSend = "Permintaan tidak dapat diproses";
89             printStream.println(messageSend);
90         }
91
92         printStream.close(); //menutup jalur komunikasi
93
94     }catch (Exception e){
95         System.out.print("[TCP Server] The server cannot send the message");
96     }
97 }
98
99 }
100
101
102

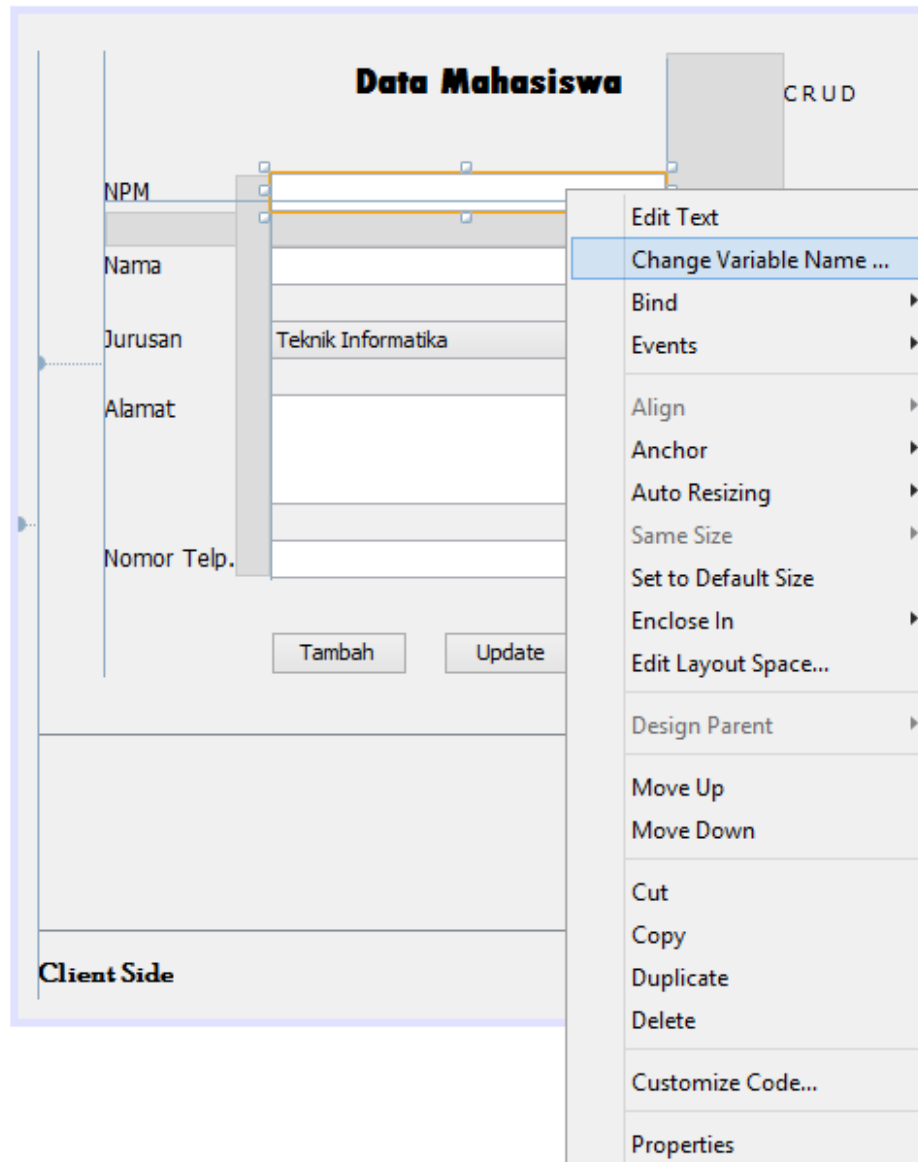
```

#Jframe Client : Membuat Tampilan Jframe Form Client

6. Selanjutnya buatlah design pada Jframe Form Client sesuai yang kalian inginkan.



Tidak lupa untuk merubah nama variabel setiap kolom sesuai fungsinya untuk mempermudah proses proses selanjutnya.



Di sini saya merubah nama variabel setiap fungsi kolom menjadi :

- Jfield1 = txtNPM
- Jfield2 = txtNama
- ComboBox1 = comboJurusan
- Jfield3 = txtAlamat
- Jfield4 = txtTelp
- Jbutton1 = clickTambah
- Jbutton2 = clickUpdate
- Jbutton3 = clickHapus

Silahkan kalian sesuaikan dengan program yang akan kalian buat

#Kode Program JFrame Form Client

- Langkah selanjutnya adalah menulis kode program untuk JFrame Form Client beserta kode program untuk tiap-tiap tombol.

Silahkan kalian ikuti dan sesuaikan kode program dibawah ini dengan kode program yang akan kalian buat. Khususnya pada bagian bagian yang telah saya tandai.

```

1 package arrayfeb_tutorial_crudtcp;
2
3 import java.io.*;
4 import java.net.Socket;
5 import java.sql.Connection;
6 import java.sql.SQLException;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9 import javax.swing.JOptionPane;
10 import javax.swing.table.DefaultTableModel;
11
12 public class ClientTCP CRUD extends javax.swing.JFrame {
13
14     public ClientTCP CRUD() {
15         initComponents();
16         loadTable();
17     }
18
19     //Fungsi Bersihkan kolom input
20     public void clear() {
21         txtNama.setText(null);
22         txtNpm.setText(null);
23         comboJurusan.setAction(null);
24         txtAlamat.setText(null);
25         txtTelp.setText(null);
26     }
27
28     //koneksi TCP
29     public void koneksiTCP(String msg) throws IOException {
30
31         Socket clientSocket = null;
32         String serverHostName = "localhost";
33         int port = 8000;
34         try {
35             clientSocket = new Socket(serverHostName, port);
36         } catch (IOException e) {
37             JOptionPane.showMessageDialog(null, "Tidak bisa terhubung dengan server");
38         }
39
40         PrintStream printStream = new PrintStream(clientSocket.getOutputStream());
41         InputStreamReader inputStream = new InputStreamReader(clientSocket.getInputStream());
42
43         String message = msg + "\n";
44
45         printStream.print(message);
46
47         BufferedReader bufferedReader = new BufferedReader(inputStream);
48
49         String pesanError = bufferedReader.readLine();
50
51         System.out.print("\nFeedback Server : \n\t");
52         System.out.println(pesanError);
53
54         JOptionPane.showMessageDialog(null, pesanError);
55     }
56

```

Method untuk membersihkan kolom input

Method untuk membuat koneksi dan mengirim message kepada server


```

57  @SuppressWarnings("unchecked")
58  Generated Code
273
274  private void txtNamaActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
277
278  private void txtNpmActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
281
282  //Kode Program Untuk Tombol Hapus
283  private void clickHapusActionPerformed(java.awt.event.ActionEvent evt) {
284      String message = "DELETE FROM mhs WHERE npm='"+txtNpm.getText()+"'";
285
286      try {
287          koneksiTCP(message);
288      } catch (IOException e) {
289
290      }
291
292      clear();
293      loadTable();
294  }
295
296  //Kode Program Untuk Keluar
297  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
298      int selectedOption = JOptionPane.showConfirmDialog(null,
299          "Apakah anda ingin menutup aplikasi?", "Tutup Aplikasi", JOptionPane.YES_NO_OPTION);
300
301      if (selectedOption == JOptionPane.YES_OPTION) {
302          System.exit(0);
303      }
304  }
305
306  private void comboJurusanActionPerformed(java.awt.event.ActionEvent evt) {
307      // TODO add your handling code here:
308  }
309
310  private void comboJurusanActionPerformed(java.awt.event.ActionEvent evt) {
311      // TODO add your handling code here:
312  }
313
314  private void txtTelpActionPerformed(java.awt.event.ActionEvent evt) {
315      // TODO add your handling code here:
316  }
317
318  //Kode Program Untuk Tombol Tambah
319  private void clickTambahActionPerformed(java.awt.event.ActionEvent evt) {
320      //Query Untuk Insert Data
321      String message = "INSERT INTO mhs VALUES ('"+txtNama.getText()+"','"+txtNpm.getText()+"','"+
322          +comboJurusan.getSelectedItem()+"','"+txtAlamat.getText()+"','"+txtTelp.getText()+"')\n";
323
324      try {
325          koneksiTCP(message);
326      } catch (IOException ex) {
327
328      }
329
330      clear();
331      loadTable();
332  }
333
334  //Kode Program Untuk Tombol Update
335  private void clickUpdateActionPerformed(java.awt.event.ActionEvent evt) {
336      //Query Untuk Update Data
337      String message = "UPDATE mhs SET nama='"+txtNama.getText()+"',npm='"+txtNpm.getText()+"',jurusan='"+comboJurusan.getSelectedItem()
338          +"',alamat='"+txtAlamat.getText()+"',phone='"+txtTelp.getText()+"' WHERE npm = '"+txtNpm.getText()+"';
339
340      try {
341          koneksiTCP(message);
342      } catch (IOException ex) {
343
344      }
345
346      clear();
347      loadTable();
348  }
349
350  private void gridtampilMouseClicked(java.awt.event.MouseEvent evt) {
351      int pilih = gridtampil.getSelectedRow();
352      String nama = gridtampil.getValueAt(pilih, 1).toString();
353      String npm = gridtampil.getValueAt(pilih, 2).toString();
354      String jurusan = gridtampil.getValueAt(pilih, 3).toString();
355      String alamat = gridtampil.getValueAt(pilih, 4).toString();
356      String telp = gridtampil.getValueAt(pilih, 5).toString();
357
358      txtNama.setText(nama);
359      txtNpm.setText(npm);
360      comboJurusan.setSelectedItem(jurusan);
361      txtAlamat.setText(alamat);
362      txtTelp.setText(telp);
363  }
364
365  public static void main(String args[]) {
366
367      System.out.print("
368          + "=====\\n"
369          + "\\t\\tTCP Client - Arry Febyran\\n"
370          + "=====\\n");
371
372      /* Set the Nimbus look and feel */
373      Look and feel setting code (optional)
374
375
376      /* Create and display the form */
377      java.awt.EventQueue.invokeLater(new Runnable() {
378          public void run() {
379              new ClientTCPCRUd().setVisible(true);
380          }
381      });
382  }

```

Kode program untuk tombol hapus

Kode program untuk tombol keluar/exit

Kode program untuk tombol tambah

Kode program untuk tombol update

Kode program untuk menampilkan data ketika kolom tabel di klik

```

403
404 private void loadTable(){
405     // membuat tampilan model tabel
406     DefaultTableModel model = new DefaultTableModel();
407     model.addColumn("No");
408     model.addColumn("Nama");
409     model.addColumn("NIM");
410     model.addColumn("Jurusan");
411     model.addColumn("Alamat");
412     model.addColumn("Phone");
413
414     //menampilkan data database kedalam tabel
415     try {
416         int no=1;
417         String sql = "select * from mhs";
418         java.sql.Connection conn=(Connection)Config.configDB();
419         java.sql.Statement stm=conn.createStatement();
420         java.sql.ResultSet res=stm.executeQuery(sql);
421         while(res.next()){
422             model.addRow(new Object[] {no++,res.getString(1),res.getString(2),res.getString(3),res.getString(4),res.getString(5)});
423         }
424         gridtampil.setModel(model);
425     } catch (Exception e) {
426     }
427 }
428
429 // Variables declaration - do not modify
430 private javax.swing.JButton clickHapus;
431 private javax.swing.JButton clickTambah;
432 private javax.swing.JButton clickUpdate;
433 private javax.swing.JComboBox<String> comboJurusan;
434 private javax.swing.JTable gridtampil;
435 private javax.swing.JButton jButton2;
436 private javax.swing.JLabel jLabel1;
437 private javax.swing.JLabel jLabel2;
438 private javax.swing.JLabel jLabel3;
439 private javax.swing.JLabel jLabel4;
440 private javax.swing.JLabel jLabel5;
441 private javax.swing.JLabel jLabel6;
442 private javax.swing.JLabel jLabel7;
443 private javax.swing.JLabel jLabel8;
444 private javax.swing.JScrollPane jScrollPane1;
445 private javax.swing.JScrollPane jScrollPane2;
446 private javax.swing.JTextArea jTextArea1;
447 private javax.swing.JTextField txtAlamat;
448 private javax.swing.JTextField txtNama;
449 private javax.swing.JTextField txtNpm;
450 private javax.swing.JTextField txtTelp;
451 // End of variables declaration
452
453

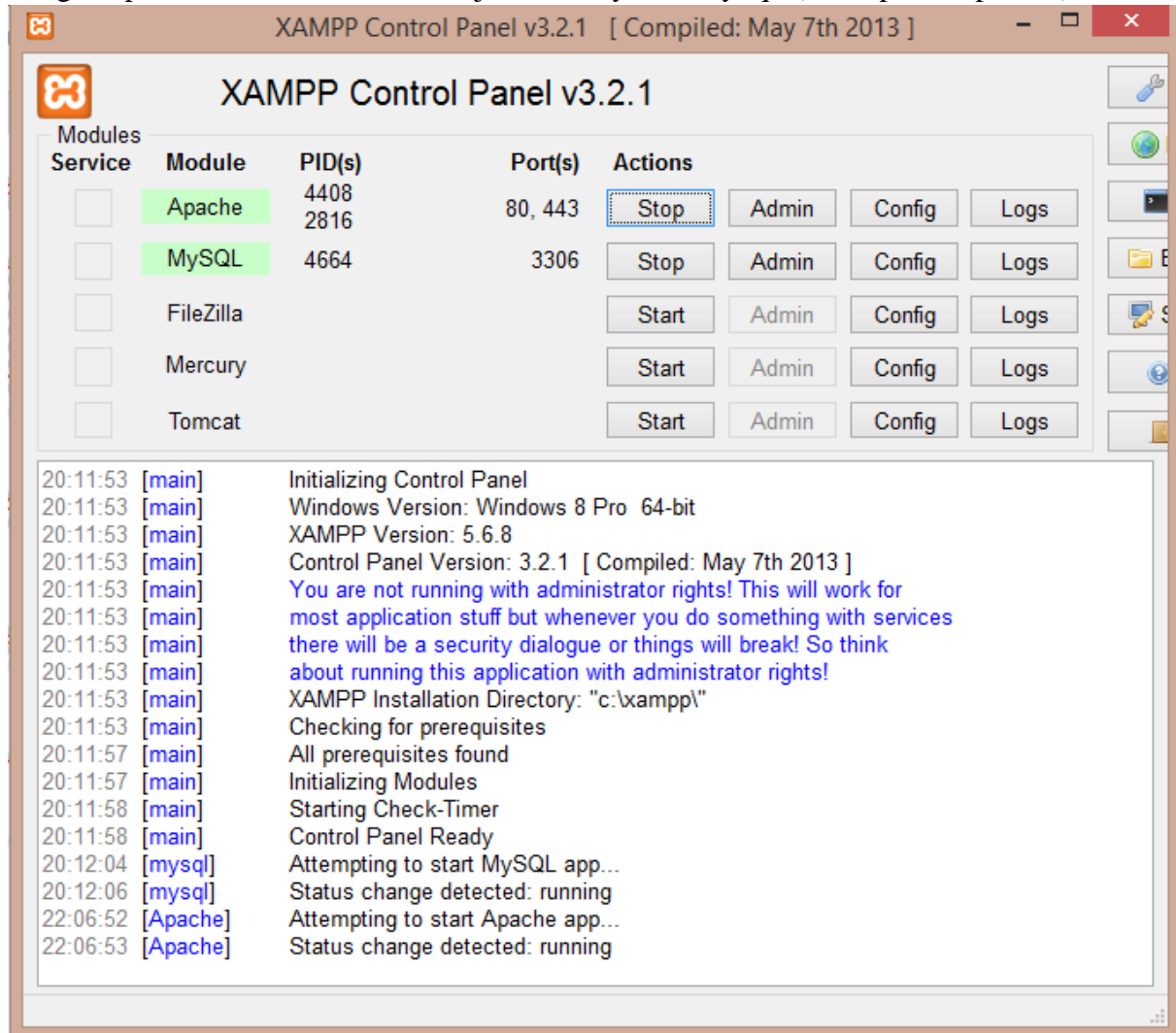
```

Kode program untuk menampilkan table pada database

Cara Menjalankan Program

Sebelumnya pastikan bahwa dalam kode program kita tidak terdapat eror. Setelah seluruh proses diatas berhasil kita lewati saatnya menjalankan program.

1. Langkah pertama, buka XAMPP dan jalankan layanan MySql, (dan Apache optional).



2. Pastikan bahwa layanan MySql telah berjalan dengan baik, kemudian jalankan program server terlebih dahulu.



3. Selanjutnya jalankan program client.

Data Mahasiswa CRUD

NPM

Nama

Jurusan

Alamat

Nomor Telp.

No	Nama	NIM	Jurusan	Alamat	Phone
1	Arry	22312	Teknik Inf...	23213	23123
2	Arry	121212	Teknik Inf...	Kepo	9128

Client Side

4. Silahkan lakukan test pada setiap fungsi pada program. Masukkan sebuah data untuk mengecek apakah program kita telah berjalan dengan baik atau tidak.

The screenshot shows a web application window titled "Data Mahasiswa" with a "CRUD" label. The form contains the following fields:

- NPM: 10225265
- Nama: So Hyun
- Jurusan: Teknik Perminyakan (dropdown menu)
- Alamat: Kepu
- Nomor Telp.: 99922187

Below the form are three buttons: "Tambah", "Update", and "Hapus".

No	Nama	NIM	Jurusan	Alamat	Phone
1	Arry	22312	Teknik Inf...	23213	23123
2	Arry	121212	Teknik Inf...	Kepo	9128

At the bottom left is the text "Client Side" and at the bottom right is a "Keluar" button.

Hasil Akhir

Client

Data Mahasiswa CRUD

NPM

Nama

Jurusan

Alamat

Nomor Telp.

No	Nama	NIM	Jurusan	Alamat	Phone
1	Arry	22312	Teknik Inf...	23213	23123
2	Arry	121212	Teknik Inf...	Kepo	9128
3	So Hyun	10225265	Teknik Pe...	Kepu	99922187

Client Side

Server

```
run:
=====
TCP Server - Arry Febryan
=====
[TCP server] Jalur komunikasi berhasil dibuat di port = 8000

[TCP Server] Menunggu permintaan dari client ...
[TCP Server] Memproses permintaan yang masuk...
Permintaan telah diterima : INSERT INTO mhs VALUES ('So Hyun','10225265','Teknik Perminyakan','Kepu','99922187')
Permintaan berhasil diproses..
```

Penutup

Alhamdulillah. Terima kasih telah mengikuti tutorial dari saya, mohon maaf jika tutorial ini kurang sempurna. Harapan saya sebagai pembuat sekaligus penulis tutorial ini, semoga tutorial ini bermanfaat untuk kita semua dan semoga ada seseorang yang dapat menyempurnakan tutorial ini menjadi lebih baik lagi.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Credits by:

- Arry Febryan (201610225200)
- Anis Koubaa
- Bpk. Rakhmat Purnomo (Dosen pembimbing mata kuliah Pemrograman Jaringan)