

JAVA x MySQL

Pada pertemuan sebelumnya kita telah mempraktekkan bagaimana program dapat menyimpan data pada database SQLite. Salah satu kekurangan SQLite adalah pada kapasitas penyimpanannya yang hanya dapat menampung hingga 2 GB.



Untuk mendapatkan kapasitas lebih besar, kita dapat menggunakan Server based SQL seperti MySQL, MariaDB, PostgreSQL, SQL Server, Oracle, dll. Pertemuan kali ini, kita akan membuat program dengan MySQL sebagai penyimpanan datanya.

Beberapa kebutuhan yang harus disiapkan sebelum memulai praktek pada modul ini antara lain:

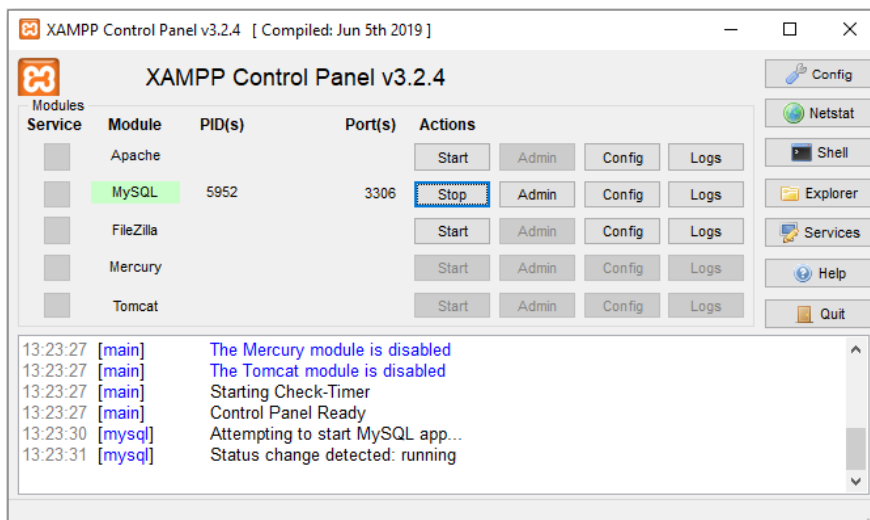
1. DBMS MySQL/MariaDB (Bisa didapatkan dari aplikasi XAMPP)
2. Netbeans IDE
3. MySQL JDBC Driver (Sudah default tersedia di Netbeans)
4. Aplikasi pengelola database MySQL (PHPMysqlAdmin, HeidiSQL, MySQL Workbench, dll)

Berikut ini materi yang akan kita ulas pada modul kali ini :

1. Membuat database dan tabel di MySQL
2. Menambahkan Library MySQL JDBC Driver di Netbeans
3. Menguji koneksi database MySQL
4. Menampilkan record pada console output
5. Menampilkan record pada form GUI
6. Membuat aplikasi CRUD dengan Java

Membuat database dan tabel di MySQL

1. Jalankan XAMPP dan nyalakan modul MySQL.



2. Akses MySQL dengan editor SQL favorit kamu. Mis. PHPMyAdmin, MySQLFront, MySQLWorkbench, HeidiSQL, dll.
3. Buatlah database baru dengan nama “toko” dan tabel “produk” pada database tersebut, lalu sesuaikan kolom-kolom seperti gambar dibawah ini.

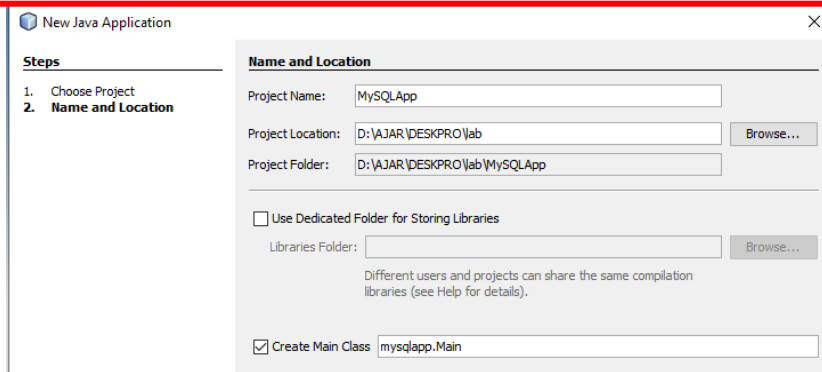
```
1 CREATE TABLE `produk` (  
2   `id_produk` INT(10) NOT NULL AUTO_INCREMENT,  
3   `nm_produk` VARCHAR(150) NOT NULL DEFAULT '',  
4   `harga_produk` INT(11) NOT NULL,  
5   PRIMARY KEY (`id_produk`)  
6 );
```

4. Isikan beberapa record ke tabel produk.

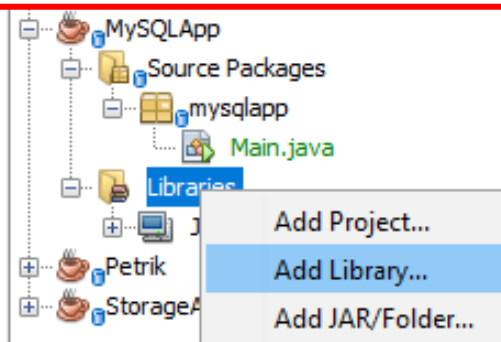
id_produk	nm_produk	harga_produk
1	Pisang Lumer	16.000
2	Tape Lumer	15.000
3	Durian Lumer	17.000

Menambahkan library MySQL JDBC Driver

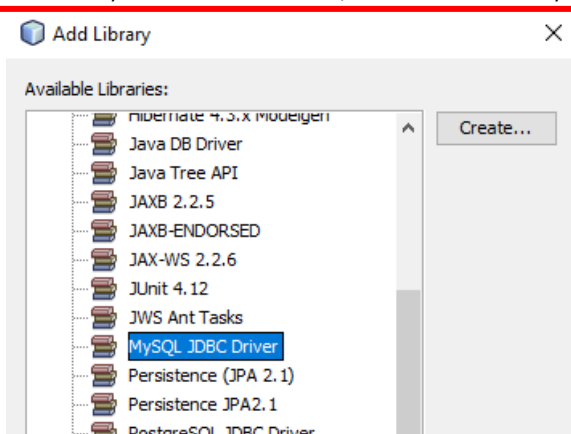
1. Buatlah project baru di IDE Netbeans



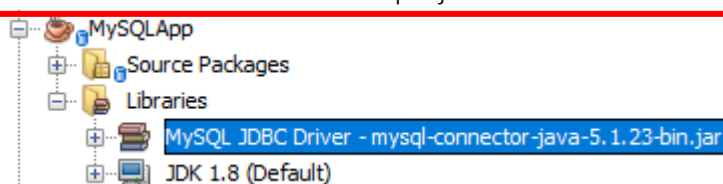
2. Library MySQL sudah secara default ada pada distribusi Netbeans jadi kita tidak perlu mengunduh file apapun lagi. Untuk menambahkan library MySQL, klik kanan pada direktori “libraries” project lalu pilih “Add Library...”



3. Pilih “MySQL JDBC Driver”, klik “Add Library”

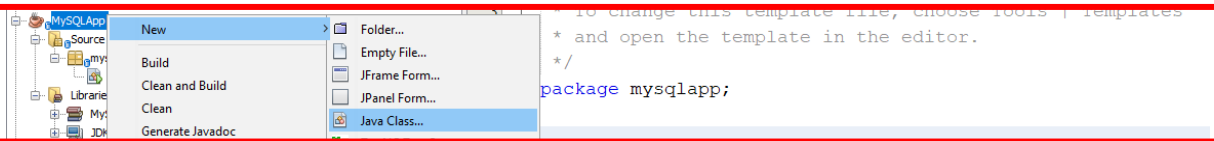


4. Bila langkah-langkah diatas telah dilakukan dengan benar, maka library MySQL akan muncul di direktori “libraries” project.

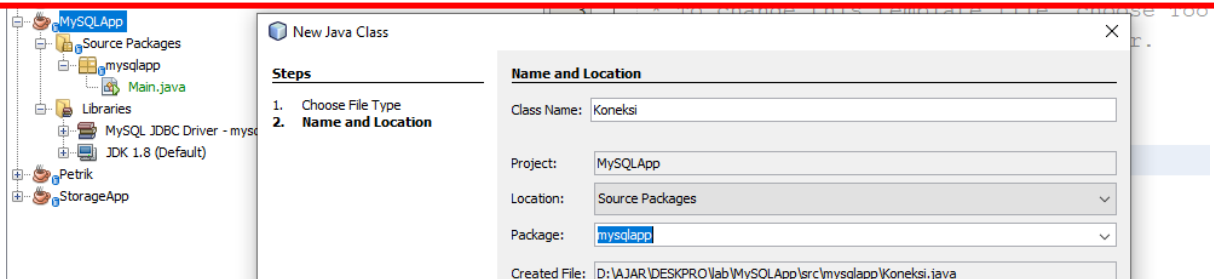


Menguji Koneksi Database MySQL

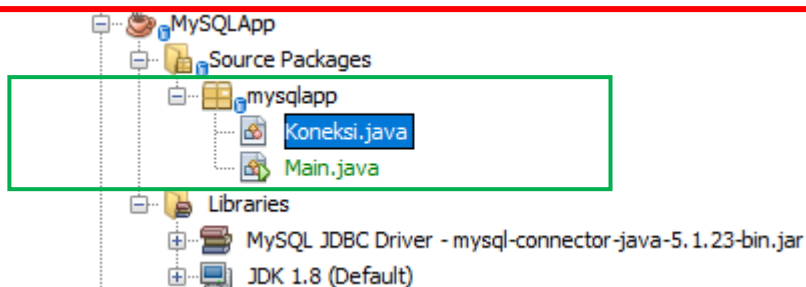
1. Buatlah class baru pada project.



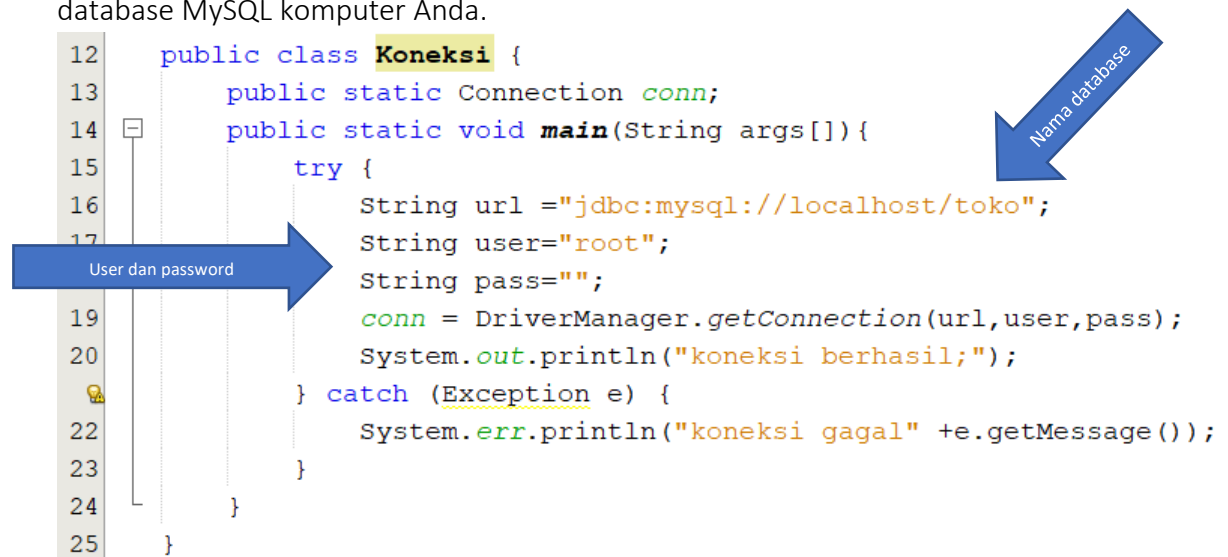
2. Beri nama Koneksi pada class baru tersebut, jangan lupa untuk mengatur package agar sama dengan class Main.



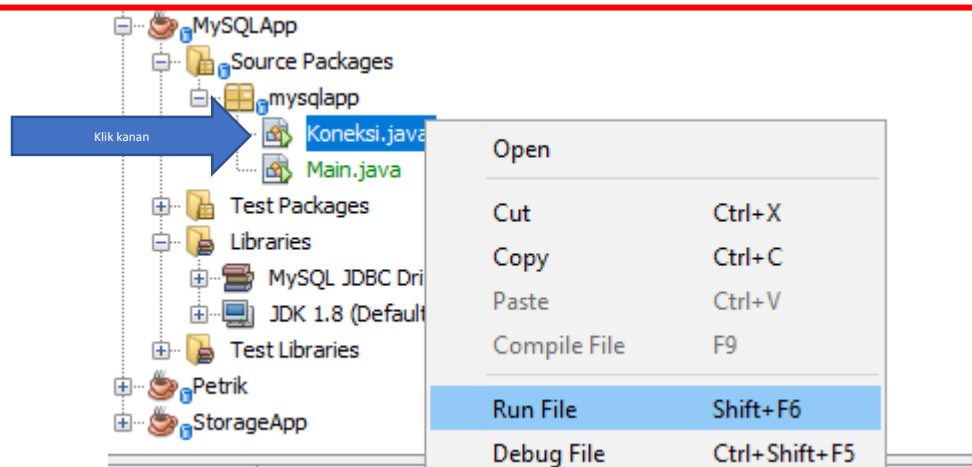
3. Jika langkah diatas diikuti dengan benar, maka class Koneksi akan berada dalam satu package dengan class Main.



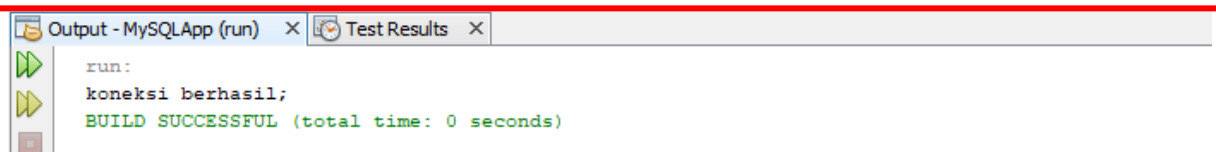
4. Tuliskan kode berikut pada class Koneksi, sesuaikan dengan user dan password database MySQL komputer Anda.



5. Jalankan class Koneksi dengan cara klik kanan pada file “Koneksi.java” lalu pilih “Run File”



6. Berikut ini tampilan di console output apabila class Koneksi berhasil dijalankan.




Menampilkan record database pada console output

1. Tambahkan method "connectDB" pada class Koneksi, berikut ini kode yang harus ditulis.

```
28 public static Connection connectDB() {
29     String url = "jdbc:mysql://localhost/toko";
30     String user = "root";
31     String pass = "";
32     try {
33         conn = DriverManager.getConnection(url, user, pass);
34     } catch (SQLException e) {
35         System.err.println("koneksi gagal" + e.getMessage());
36     }
37     return conn;
38 }
```

2. Ingat, kode diatas harus dituliskan di dalam class Koneksi

```
13 public class Koneksi {
14     public static Connection conn;
15
16     public static void main(String args[]) { ...11 lines }
17
18     public static Connection connectDB() { ...11 lines }
19 }
```

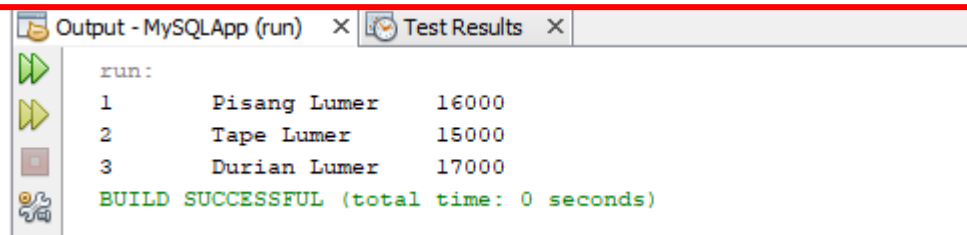


Catatan : Tujuan dari pembuatan method `connectDB()` adalah agar kita tidak perlu berulang-kali menuliskan konfigurasi database di setiap class atau form.

3. Berikut ini contoh pemanggilan method `connectDB()` pada class Main. Tuliskan kode berikut pada class Main untuk menampilkan record tabel produk di console output.

```
16 public class Main {
17
18     public static void main(String[] args) {
19         String sql = "SELECT * FROM produk";
20
21         try (Connection conn = Koneksi.connectDB()) {
22             Statement stmt = conn.createStatement();
23             ResultSet rs = stmt.executeQuery(sql);
24
25             while (rs.next()) {
26                 System.out.println(rs.getInt("id_produk") + "\t" +
27                                     rs.getString("nm_produk") + "\t" +
28                                     rs.getString("harga_produk"));
29             }
30
31         } catch (SQLException e) {
32             System.out.println(e.getMessage());
33         }
34     }
```

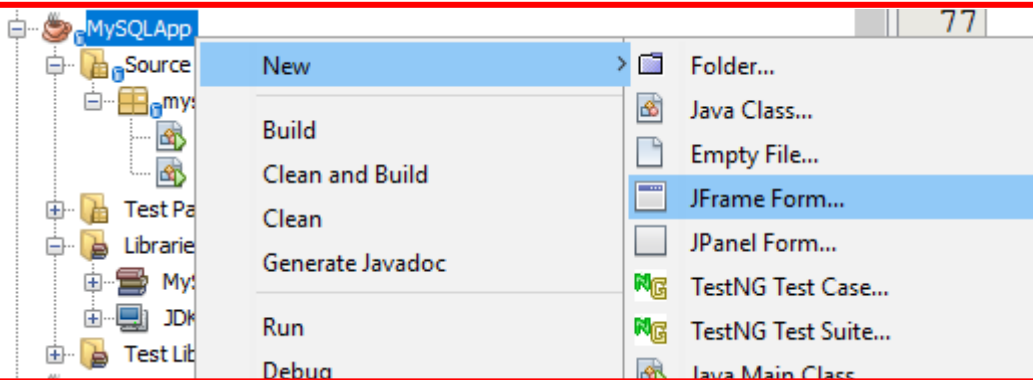
4. Berikut ini tampilan di console output apabila class Main berhasil dijalankan.



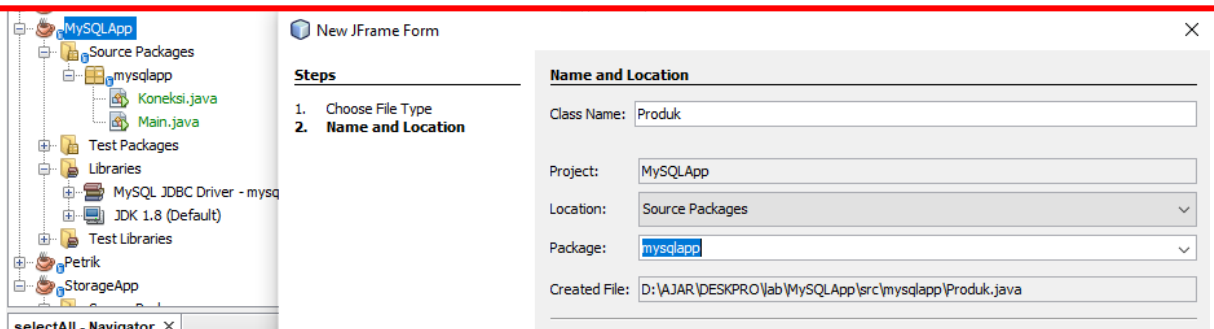
```
run:
1    Pisang Lumer    16000
2    Tape Lumer     15000
3    Durian Lumer    17000
BUILD SUCCESSFUL (total time: 0 seconds)
```

Menampilkan Record pada Form Dalam Bentuk Tabel

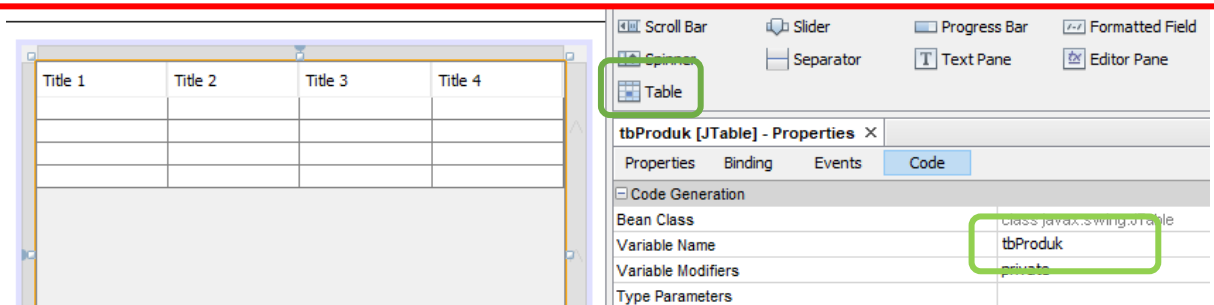
1. Buatlah Form baru pada project dengan memilih "Jframe Form"



2. Isikan "Produk" sebagai "Class Name", jangan lupa untuk mengatur agar class berada dalam satu package dengan class Koneksi.



3. Tambahkan komponen "Table" ke dalam form, dan ubah "variable name" menjadi "tbProduk".



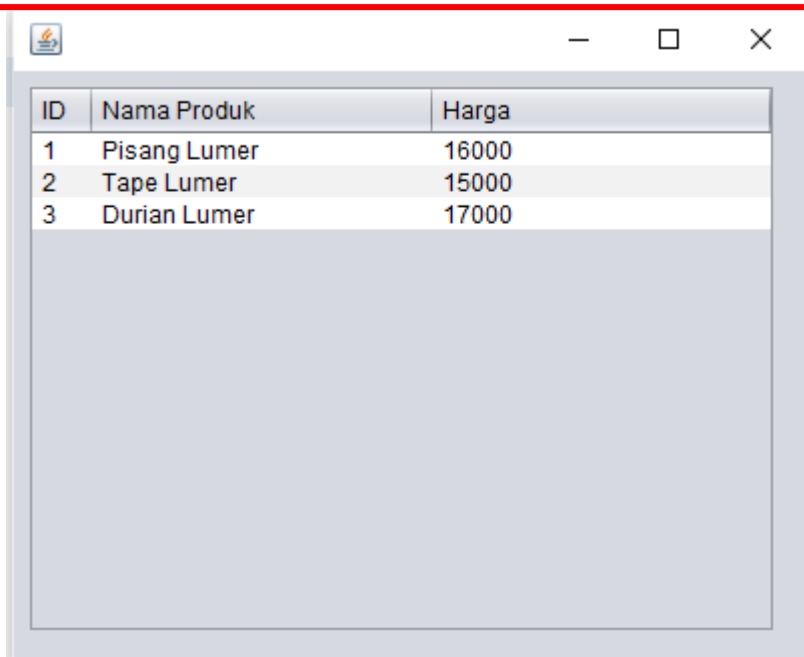
4. Buatlah method “loadTabelProduk” pada class Produk, berikut ini kode yang harus ditulis.

```
28 Connection conn = Koneksi.connectDB();
29 public void loadTabelProduk() {
30     String sql = "SELECT * FROM produk";
31     Object[] kolom = { "ID", "Nama Produk", "Harga" };
32     DefaultTableModel dataModel = new DefaultTableModel(null, kolom);
33     tbProduk.setModel(dataModel);
34     tbProduk.getColumnModel().getColumn(0).setMaxWidth(30);
35
36     try {
37         Statement stmt = conn.createStatement();
38         ResultSet rs = stmt.executeQuery(sql);
39
40         while(rs.next()) {
41             int id = rs.getInt("id_produk");
42             String nama = rs.getString("nm_produk");
43             String harga = rs.getString("harga_produk");
44
45             Object[] data={id, nama, harga};
46             dataModel.addRow(data);
47         }
48
49     } catch (SQLException e) {
50         System.out.println(e.getMessage());
51     }
52 }
```

5. Panggil method “loadTabelProduk” di constructor class Produk.

```
23 public Produk() {
24     initComponents();
25     loadTabelProduk();
26 }
```

6. Berikut ini tampilan aplikasi apabila berhasil dijalankan.

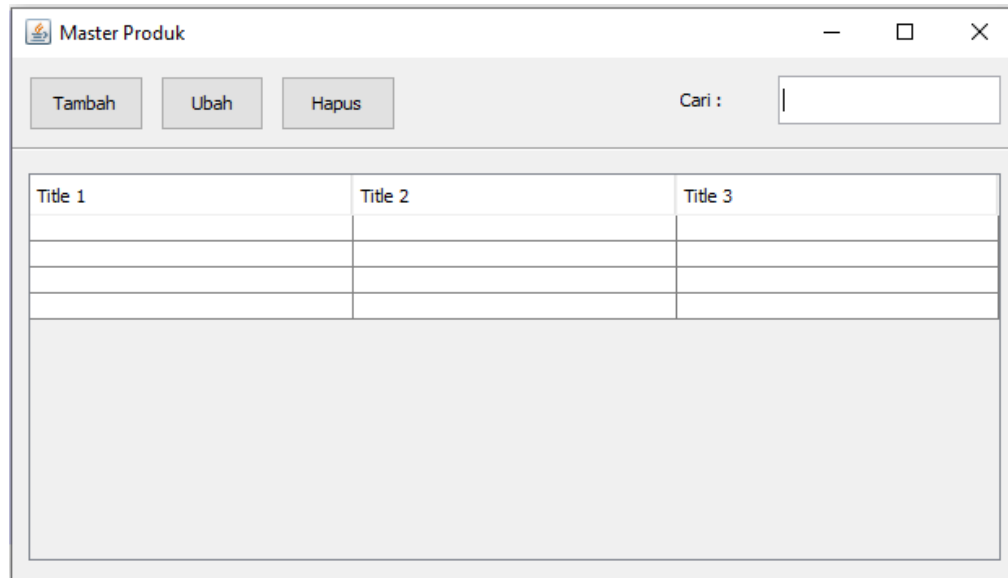


The screenshot shows a Java Swing window with a title bar containing standard OS controls (minimize, maximize, close). Inside the window is a table with three columns: "ID", "Nama Produk", and "Harga". The table contains three rows of data. Below the table is a large, empty rectangular area, likely a placeholder for a text area or another component.

ID	Nama Produk	Harga
1	Pisang Lumer	16000
2	Tape Lumer	15000
3	Durian Lumer	17000

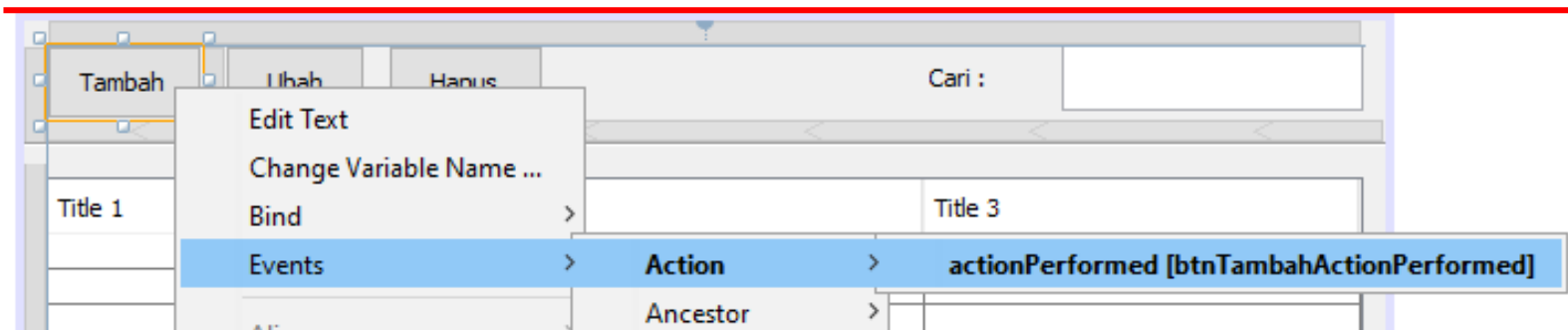
Membuat Aplikasi CRUD dengan Java

1. Ubahlah antarmuka pada latihan sebelumnya sehingga menjadi seperti dibawah ini.



Komponen	variable name
JButton	btnTambah
JButton	btnUbah
JButton	btnHapus
TextField	txtCari
Jtable	tbProduk

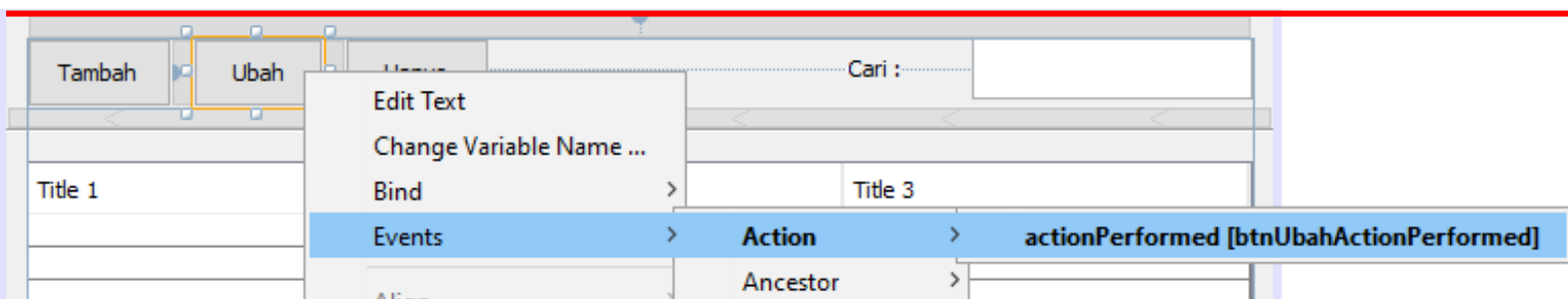
2. Tambahkan event “actionPerformed” untuk “btnTambah”.



3. Berikut ini kode event “actionPerformed” untuk “btnTambah”.

```
200 private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {  
201  
202     JTextField txtNama = new JTextField(15);  
203     JTextField txtHarga = new JTextField(15);  
204  
205     JPanel myPanel = new JPanel();  
206     myPanel.setLayout(new GridLayout(2,2,0,10));  
207     myPanel.add(new JLabel("Nama Produk"));  
208     myPanel.add(txtNama);  
209     myPanel.add(new JLabel("Harga"));  
210     myPanel.add(txtHarga);  
211  
212     int result = JOptionPane.showConfirmDialog(this, myPanel,  
213         "Input produk baru",  
214         JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);  
215  
216     if (result == JOptionPane.OK_OPTION) {  
217         try {  
218             Statement stmt = conn.createStatement();  
219             stmt.executeUpdate("INSERT INTO produk(nm_produk,harga_produk)" +  
220                 " VALUES('" + txtNama.getText() + "', '" +  
221                 txtHarga.getText() + "')");  
222             loadTabelProduk();  
223         } catch (SQLException e) {  
224             System.out.println(e.getMessage());  
225         }  
226     }  
227 }
```

4. Tambahkan event “actionPerformed” untuk “btnUbah”



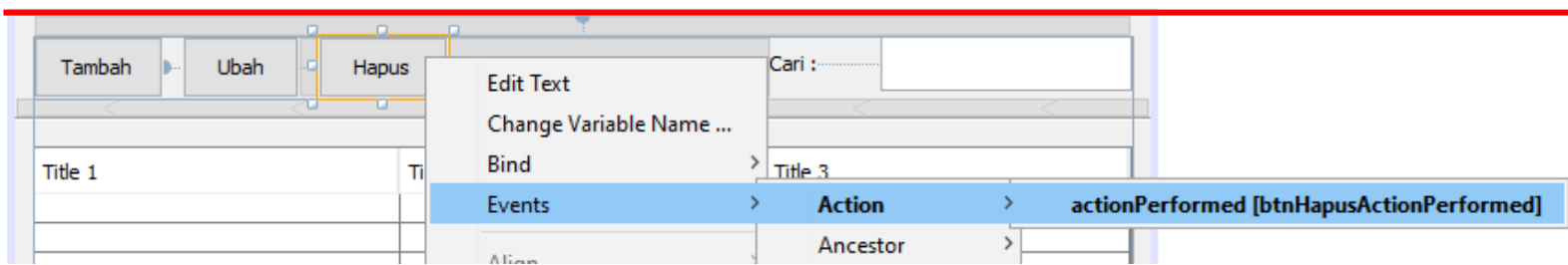
5. Berikut ini kode event “actionPerformed” untuk “btnUbah”.

```

229     int selectedID;
230     private void btnUbahActionPerformed(java.awt.event.ActionEvent evt) {
231         if (tbProduk.isRowSelected(tbProduk.getSelectedRow()) == true) {
232             selectedID = (int)tbProduk.getValueAt(tbProduk.getSelectedRow(), 0);
233             JTextField txtNama = new JTextField(15);
234             JTextField txtHarga = new JTextField(15);
235             txtNama.setText(tbProduk.getValueAt(tbProduk.getSelectedRow(), 1).toString());
236             txtHarga.setText(tbProduk.getValueAt(tbProduk.getSelectedRow(), 2).toString());
237             JPanel myPanel = new JPanel();
238             myPanel.setLayout(new GridLayout(2, 2, 0, 10));
239             myPanel.add(new JLabel("Nama Produk"));
240             myPanel.add(txtNama);
241             myPanel.add(new JLabel("Harga"));
242             myPanel.add(txtHarga);
243
244             int result = JOptionPane.showConfirmDialog(this, myPanel,
245                 "Ubah produk",
246                 JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);
247
248             if (result == JOptionPane.OK_OPTION) {
249                 try {
250                     Statement stmt = conn.createStatement();
251                     stmt.executeUpdate("UPDATE produk SET nm_produk='"
252                         + txtNama.getText() + "', harga_produk='"
253                         + txtHarga.getText() + "' WHERE id_produk='"
254                         + selectedID + "'");
255                     loadTabelProduk();
256                 } catch (SQLException e) {
257                     System.out.println(e.getMessage());
258                 }
259             }
260             else {
261                 JOptionPane.showMessageDialog(this, "Tidak ada yang dipilih");
262             }
263         }
264     }

```

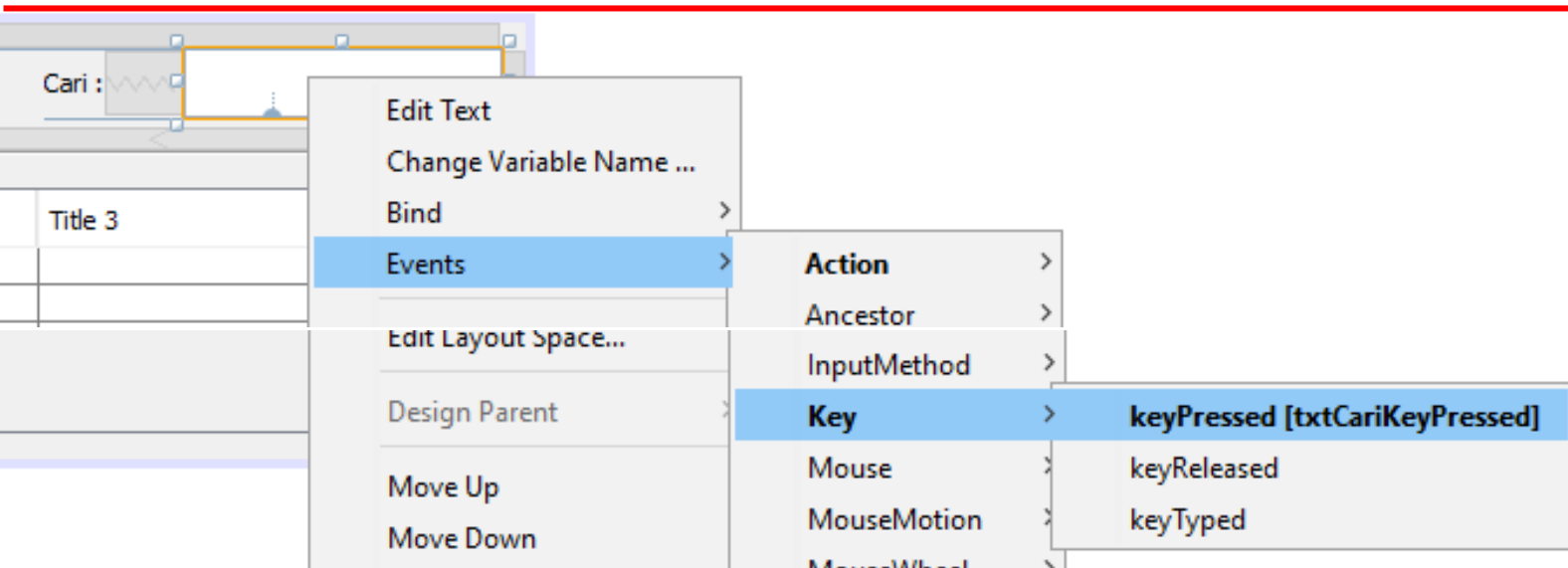
6. Tambahkan event “actionPerformed” untuk “btnHapus”



7. Berikut ini kode event “actionPerformed” untuk “btnHapus”.

```
269 private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  
270     if(tbProduk.isRowSelected(tbProduk.getSelectedRow())==true){  
271         selectedID = (int)tbProduk.getValueAt(tbProduk.getSelectedRow(), 0);  
272  
273         int result = JOptionPane.showConfirmDialog(  
274             this,  
275             "Kamu akan menghapus data "  
276                 + tbProduk.getValueAt(tbProduk.getSelectedRow(), 1).toString(),  
277             "Peringatan!",  
278             JOptionPane.YES_NO_OPTION,  
279             JOptionPane.QUESTION_MESSAGE  
280         );  
281  
282         if (result == JOptionPane.YES_OPTION) {  
283             try {  
284                 Statement stmt = conn.createStatement();  
285                 stmt.executeUpdate("DELETE FROM produk WHERE id_produk='"  
286                     + selectedID + "'");  
287                 loadTabelProduk();  
288             } catch (SQLException e) {  
289                 System.out.println(e.getMessage());  
290             }  
291         }  
292     } else {  
293         JOptionPane.showMessageDialog(this, "Tidak ada yang dipilih");  
294     }  
295 }
```

8. Tambahkan event “keyPressed” untuk “txtCari”



9. Sebelum menuliskan kode event “keyPressed” untuk “txtCari”, kita overload dahulu method “loadTabelProduk”, berikut ini kode nya.

```
64 public void loadTabelProduk(String teks){
65     String sql = "SELECT * FROM produk WHERE nm_produk like '%" + teks + "%'";
66     Object[] kolom = { "ID", "Nama Produk", "Harga" };
67     DefaultTableModel dataModel = new DefaultTableModel(null, kolom);
68     tbProduk.setModel(dataModel);
69     tbProduk.getColumnModel().getColumn(0).setMaxWidth(30);
70
71     try {
72         Statement stmt = conn.createStatement();
73         ResultSet rs = stmt.executeQuery(sql);
74
75         while(rs.next()){
76             int id = rs.getInt("id_produk");
77             String nama = rs.getString("nm_produk");
78             String harga = rs.getString("harga_produk");
79
80             Object[] data={id, nama, harga};
81             dataModel.addRow(data);
82         }
83
84     } catch (SQLException e) {
85         System.out.println(e.getMessage());
86     }
87 }
```

Catatan : perbedaan antara kedua method “loadTabelProduk” ada pada dua baris pertamanya, selebihnya sama dan identik.

10. Berikut ini kode event “keyPressed” untuk “txtCari”.

```
297 private void txtCariKeyPressed(java.awt.event.KeyEvent evt) {
298     if(evt.getKeyCode() == KeyEvent.VK_ENTER) {
299         String teks = txtCari.getText();
300         loadTabelProduk(teks);
301     }
302 }
```

11. Berikut ini tampilan apabila program berhasil dijalankan.

