

PRAKTIKUM

MK PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

KIF193009

PRAKTIKUM 7 GUI JTable dan CRUD Database



Muhammad Ilham Wahiduddin 1931730140

DIII Manajemen Informatika
Program Studi di Luar Kampus Utama
Politeknik Negeri Malang Kampus Kediri
2020

PRAKTIKUM 7

GUI JTable dan CRUD Database

A. TUJUAN

Setelah melakukan praktikum bab ini, Mahasiswa mampu menerapkan konsep GUI dalam bahasa pemrograman java dan menghubungkannya pada database MySQL

B. PERALATAN DAN KOMPONEN

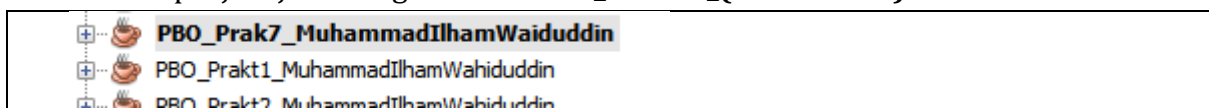
Pada kegiatan praktikum ini diperlukan alat sebagai berikut

1. Personal Computer/Laptop
2. Netbeans IDE
3. MySQL

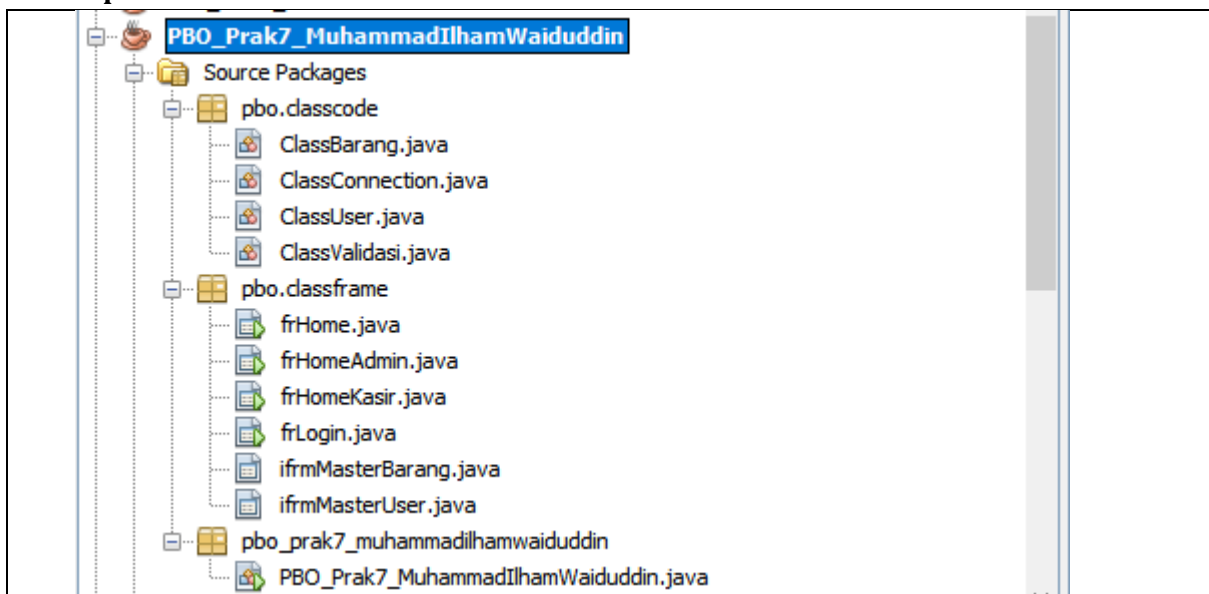
C. LANGKAH PRAKTIKUM

C1. Mempersiapkan frame untuk admin dan kasir

1. Buat project java dengan nama **PBO_Prakt7_(NamaAnda)**



2. Copy package pbo.classcode dan pbo.classframe pada praktikum 6 ke project praktikum 7.
7. Buat internal frame untuk mengelola data barang dan mengelola data user di **pbo.classframe**.



3. Buat frame beranda/home untuk admin! Buat pula konstruktor baru dengan parameter masukan nama!

FileMaster Data

SISTEM INFORMASI PENJUALAN

Selamat Datang !

Ilham W (c)

```
14 public class frHomeAdmin extends javax.swing.JFrame {
15
16     /**
17      * Creates new form frHomeAdmin
18      */
19
20     String nama;
21
22     public frHomeAdmin() {
23         initComponents();
24     }
25
26     public frHomeAdmin(String namalogin){
27         initComponents();
28         nama = namalogin;
29         jLabel17.setText("Selamat Datang "+nama+" (Admin) ");
30     }
31 }
```

4. Buat frame beranda/home untuk kasir! Buat pula konstruktor baru dengan parameter masukan nama!

FileMaster Data

SISTEM INFORMASI PENJUALAN

Selamat Datang !

Ilham W (c)

```
14 public class frHomeKasir extends javax.swing.JFrame {
15
16     /**
17      * Creates new form frHomeKasir
18      */
19
20     String nama;
21
22     public frHomeKasir() {
23         initComponents();
24     }
25
26     public frHomeKasir(String namallogin){
27         initComponents();
28         nama = namallogin;
29         jLabel17.setText("Selamat Datang "+nama+" (Kasir) ");
30     }
```

5. Buat internal frame master data user untuk menampilkan dan mengelola data user!

The screenshot shows a web application window titled "Data User". On the left, there is a form with the following fields: "Username" (text input), "Password" (text input), and "Level" (dropdown menu with "Admin" selected). Below these fields are two buttons: "Tambah" (Add) and "Edit". To the right of the form is a search bar labeled "Cari" and a "Hapus" (Delete) button. Below the search bar is a table with three columns: "Username", "Password", and "Level". The table has three empty rows. Below the table is a large empty rectangular area.

6. Buat internal frame master data barang untuk menampilkan dan mengelola data barang!

The screenshot shows a web application window titled "Data Barang". On the left, there is a form with the following fields: "Kode" (text input), "Nama" (text input), "Stok" (text input), and "Harga" (text input). Below these fields are two buttons: "Tambah" (Add) and "Edit". To the right of the form is a search bar labeled "Cari" and a "Hapus" (Delete) button. Below the search bar is a table with four columns: "Kode", "Nama", "Stok", and "Harga". The table has three empty rows. Below the table is a large empty rectangular area.

7. Panggil internal frame master data user pada home Admin!

```

144 private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt)
145
146     // TODO add your handling code here:
147     ifrmMasterUser ifr = new ifrmMasterUser();
148     ifr.setVisible(true);
149     jDesktopPanel.removeAll();
150     jDesktopPanel.add(ifr);
151 }

```

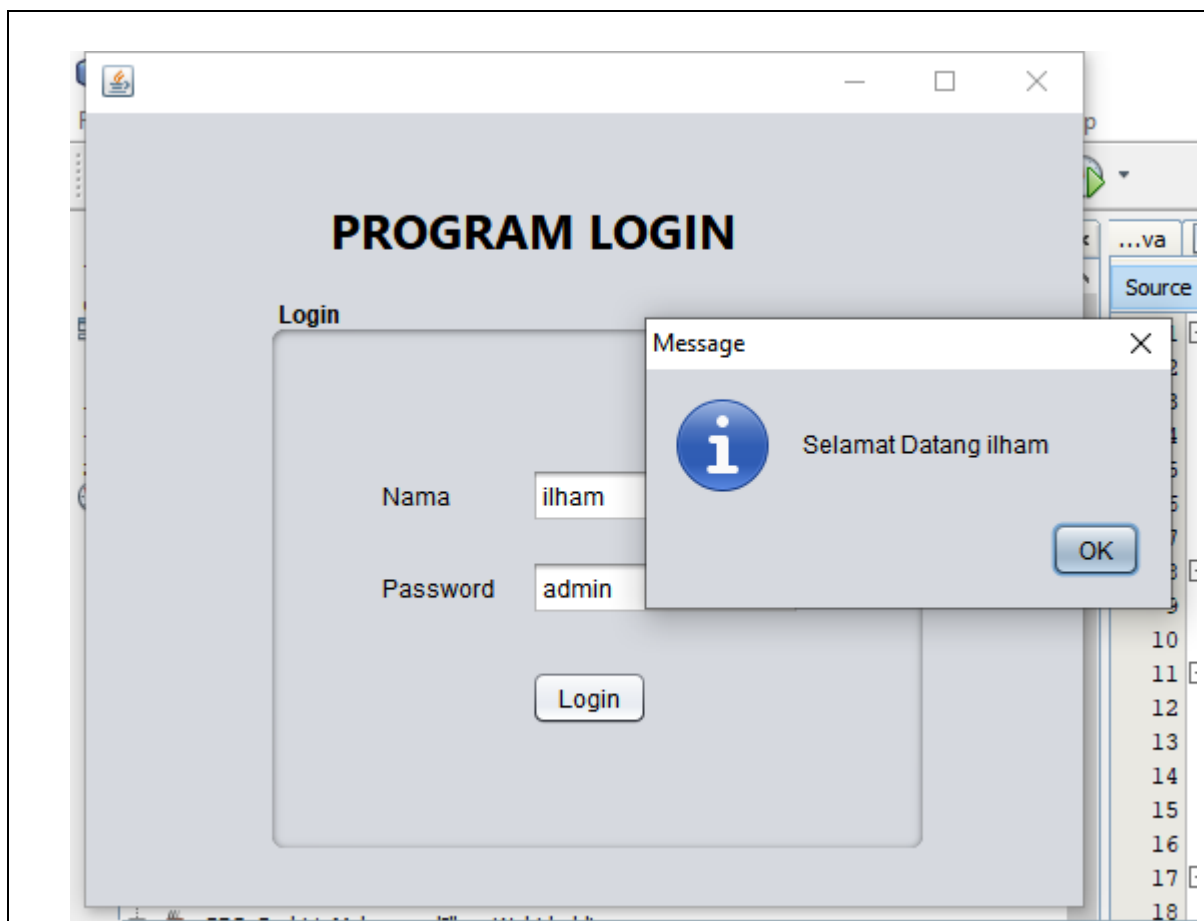
8. Panggil internal frame master data user pada home Kasir!

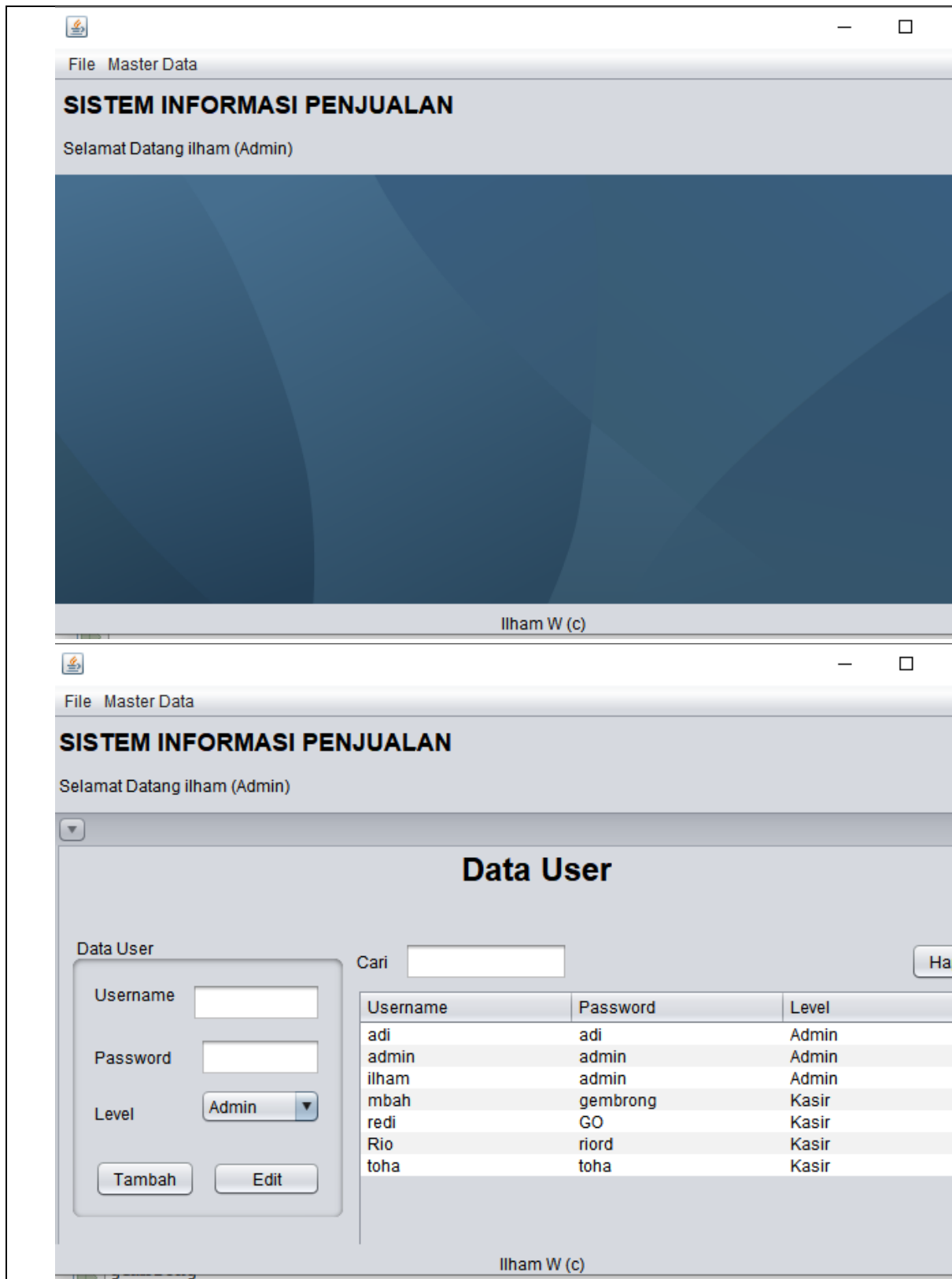
```

140 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
141
142     // TODO add your handling code here:
143     ifrmMasterBarang ifr = new ifrmMasterBarang();
144     ifr.setVisible(true);
145     jDesktopPane2.removeAll();
146     jDesktopPane2.add(ifr);
147 }

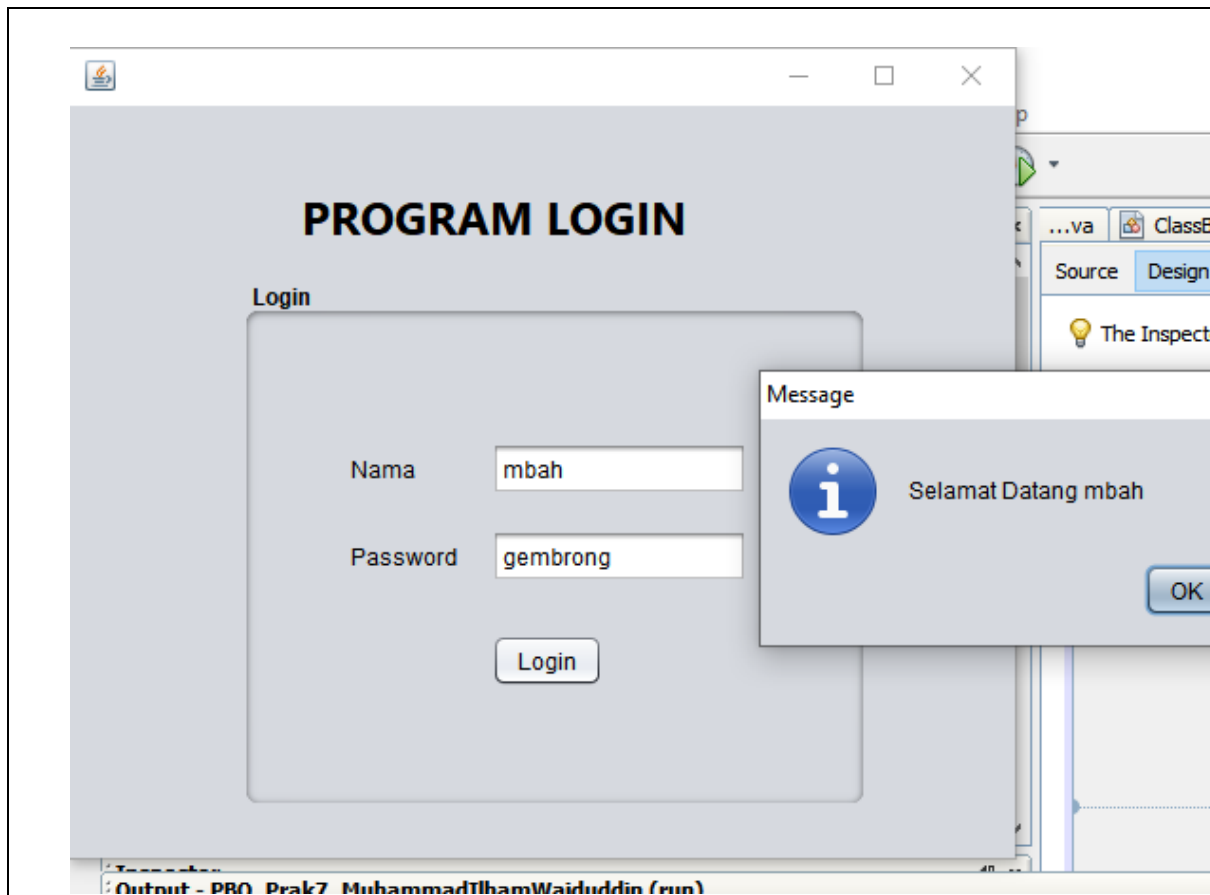
```

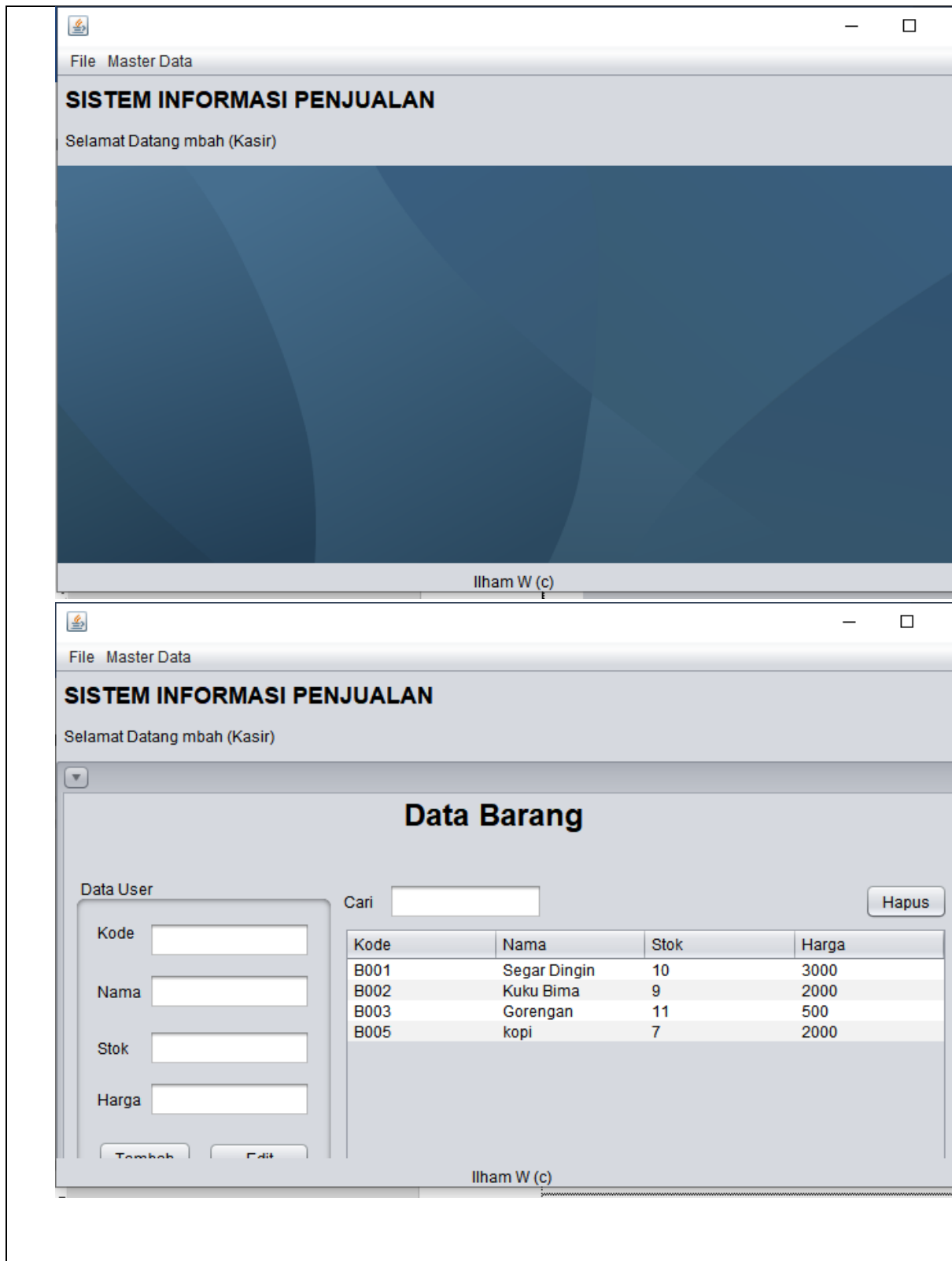
9. Jalankan aplikasi dan tunjukkan hasil running dimulai dari login admin > home admin > master data user!





10. Jalankan aplikasi dan tunjukkan hasil running dimulai dari login kasir > home kasir > master data barang!





C2. Menampilkan data dari database di JTable

1. Tambahkan atribut dan method pada ClassUser yang diperlukan untuk menampilkan data!

```
21 public class ClassUser {
22
23     private String username;
24     private String password;
25     private String level;
26     private int ndata = 0;
27     private Object[][] data;
28
29     private Connection conn;
30     private Statement st;
31
32     public ClassUser () {
33
34
35     }
36
```

```

108 public void getDataUser() {
109     try{
110         conn = ClassConnection.getKoneksi();
111         st = conn.createStatement();
112         String sql = "select count(*) from user";
113         ResultSet rs = st.executeQuery(sql);
114         while(rs.next()){
115             ndata = rs.getInt("COUNT(*)");
116         }
117         sql = "select * from user";
118         rs = st.executeQuery(sql);
119         data = new Object[ndata][3];
120         int idx=0;
121         while(rs.next()){
122             data[idx][0] = rs.getString("username");
123             data[idx][1] = rs.getString("password");
124             data[idx][2] = rs.getString("level");
125             idx++;
126         }
127         rs.close();
128         st.close();
129     }catch (SQLException ex){
130         Logger.getLogger(ClassUser.class.getName()).log(Level.SEVERE, null, ex);
131     }
132 }
133
134 public void getDataUser(String key){
135     try{
136         conn = ClassConnection.getKoneksi();
137         st = conn.createStatement();
138         String sql = "select count(*) from user where username like '%" + key + "%'";
139         ResultSet rs = st.executeQuery(sql);
140         while(rs.next()){
141             ndata = rs.getInt("COUNT(*)");
142         }
143         sql = "select * from user where username like '%" + key + "%'";
144         rs = st.executeQuery(sql);
145         data = new Object[ndata][3];
146         int idx=0;
147         while(rs.next()){
148             data[idx][0] = rs.getString("username");
149             data[idx][1] = rs.getString("password");
150             data[idx][2] = rs.getString("level");
151             idx++;
152         }
153         rs.close();
154         st.close();
155     }catch (SQLException ex){
156         Logger.getLogger(ClassUser.class.getName()).log(Level.SEVERE, null, ex);
157     }
158 }
159
160 public int getNumberDataUser() {
161     return ndata;
162 }
163
164 public Object[][] getAllDataUser() {
165     return data;

```

2. Buat method loadData() pada internal frame master user untuk memanggil data User!

```

28 public void loadData() {
29     ModelTabelUser();
30     ClassUser cu = new ClassUser();
31     cu.getDataUser();
32     int ndata = cu.getNumberDataUser();
33     Object[][] data = cu.getAllDataUser();
34     Object[] datal = new Object[3];
35     for (int i = 0; i < ndata; i++){
36         for (int j = 0; j < 3;j++){
37             datal[j] = data[i][j];
38             System.out.println(datal[j]);
39         }
40         tabelmodel.addRow(datal);
41     }
42 }

```

3. Buat method ModelTabelUser() pada internal frame master user untuk membuat kolom!

```

54     DefaultTableModel tabelmodel;
55
56 public ifrmMasterUser() {
57     initComponents();
58     loadData();
59 }
60
61 public void ModelTabelUser() {
62     tabelmodel = new DefaultTableModel();
63     jTable1.setModel(tabelmodel);
64     tabelmodel.addColumn("Username");
65     tabelmodel.addColumn("Password");
66     tabelmodel.addColumn("Level");
67     tabelmodel.getDataVector().removeAllElements();
68     tabelmodel.fireTableDataChanged();
69 }

```

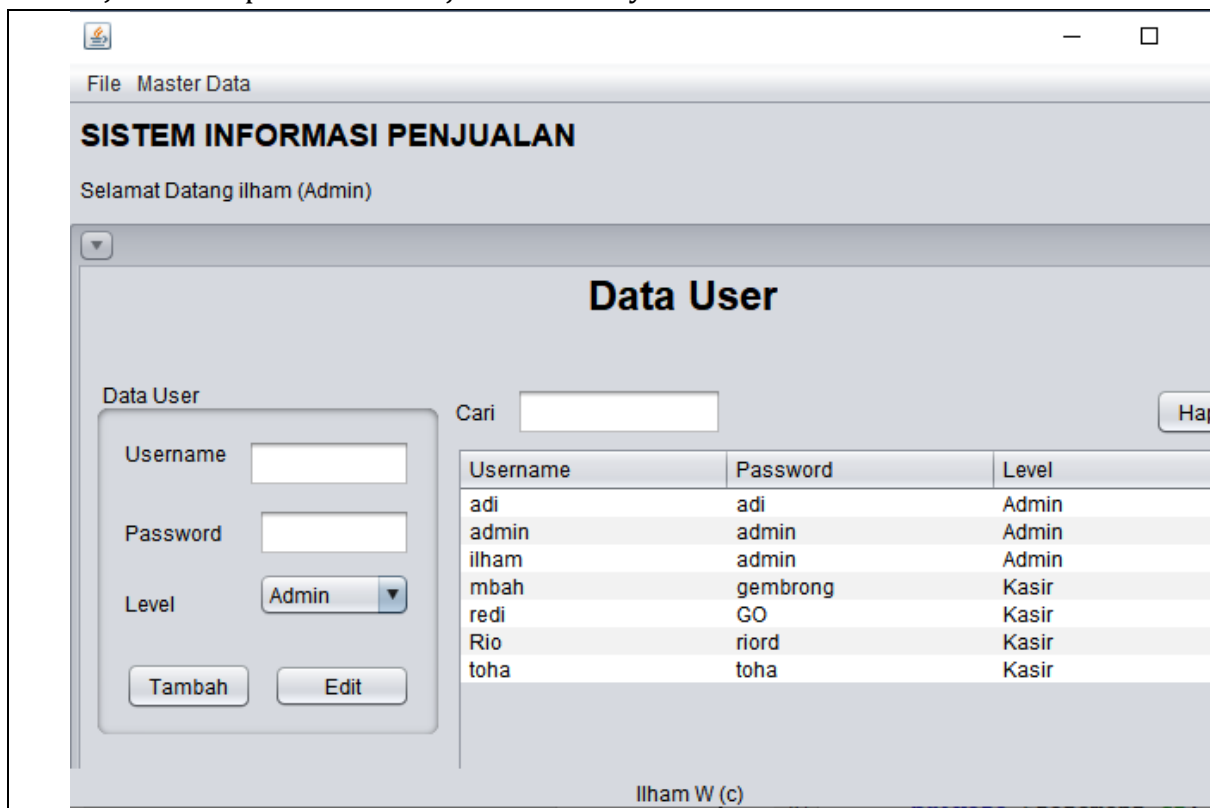
4. Panggil method loadData() pada konstruktor internal frame!

```

39 public void loadData(String key){
40     ModelTabelUser();
41     ClassUser cu = new ClassUser();
42     cu.getDataUser(key);
43     int ndata = cu.getNumberDataUser();
44     Object[][] data = cu.getAllDataUser();
45     Object[] datal = new Object[3];
46     for (int i = 0; i < ndata; i++){
47         for (int j = 0; j < 3;j++){
48             datal[j] = data[i][j];
49             System.out.println(datal[j]);
50         }
51         tabelmodel.addRow(datal);
52     }
53 }
54 DefaultTableModel tabelmodel;
55
56 public ifrmMasterUser() {
57     initComponents();
58     loadData();
59 }

```

5. Jalankan aplikasi dan tunjukkan hasilnya!



6. Tambahkan atribut dan method pada ClassBarang yang diperlukan untuk menampilkan data!

```
21 public class ClassBarang {
22
23     private String kode;
24     private String nama;
25     private int stok;
26     private int harga;
27     private int ndata = 0;
28     private Object[][] data;
29
30     private Connection conn;
31     private Statement st;
32
33     public ClassBarang() {
34
35     }
```

```

110
111 public void getDataBarang() {
112     try{
113         conn = ClassConnection.getKoneksi();
114         st = conn.createStatement();
115         String sql = "select count(*) from barang";
116         ResultSet rs = st.executeQuery(sql);
117         while(rs.next()) {
118             ndata = rs.getInt("COUNT(*)");
119         }
120         sql = "select * from barang order by kode asc";
121         rs = st.executeQuery(sql);
122         data = new Object[ndata][4];
123         int idx=0;
124         while(rs.next()) {
125             data[idx][0] = rs.getString("kode");
126             data[idx][1] = rs.getString("nama");
127             data[idx][2] = rs.getInt("stok");
128             data[idx][3] = rs.getInt("harga");
129             idx++;
130         }
131         rs.close();
132         st.close();
133     }catch (SQLException ex){
134         Logger.getLogger(ClassBarang.class.getName()).log(Level.SEVERE, null, ex);
135     }
136 }

138 public void getDataBarang(String key) {
139     try{
140         conn = ClassConnection.getKoneksi();
141         st = conn.createStatement();
142         String sql = "select count(*) from barang where nama like '%" + key + "%'";
143         ResultSet rs = st.executeQuery(sql);
144         while(rs.next()) {
145             ndata = rs.getInt("COUNT(*)");
146         }
147         sql = "select * from barang where nama like '%" + key + "%'";
148         rs = st.executeQuery(sql);
149         data = new Object[ndata][4];
150         int idx=0;
151         while(rs.next()) {
152             data[idx][0] = rs.getString("kode");
153             data[idx][1] = rs.getString("nama");
154             data[idx][2] = rs.getInt("stok");
155             data[idx][3] = rs.getInt("harga");
156             idx++;
157         }
158         rs.close();
159         st.close();
160     }catch (SQLException ex){
161         Logger.getLogger(ClassBarang.class.getName()).log(Level.SEVERE, null, ex);
162     }
163 }

164
165 public int getNumberDataBarang() {
166     return ndata;
167 }

168
169 public Object[][] getAllDataBarang() {
170     return data;

```

7. Buat method loadData() pada internal frame master barang untuk memanggil data Barang!

```

27 public void loadData() {
28     ModelTabelBarang();
29     ClassBarang cb = new ClassBarang();
30     cb.getDataBarang();
31     int ndata = cb.getNumberDataBarang();
32     Object[][] data = cb.getAllDataBarang();
33     Object[] datal = new Object[4];
34     for (int i = 0; i < ndata; i++){
35         for (int j = 0; j < 4;j++){
36             datal[j] = data[i][j];
37             System.out.println(datal[j]);
38         }
39         tabelmodel.addRow(datal);
40     }
41 }

```

8. Buat method ModelTabelBarang() pada internal frame master user untuk membuat kolom!

```

59 public void ModelTabelBarang() {
60     tabelmodel = new DefaultTableModel();
61     jTable1.setModel(tabelmodel);
62     tabelmodel.addColumn("Kode");
63     tabelmodel.addColumn("Nama");
64     tabelmodel.addColumn("Stok");
65     tabelmodel.addColumn("Harga");
66     tabelmodel.getDataVector().removeAllElements();
67     tabelmodel.fireTableDataChanged();
68 }

```

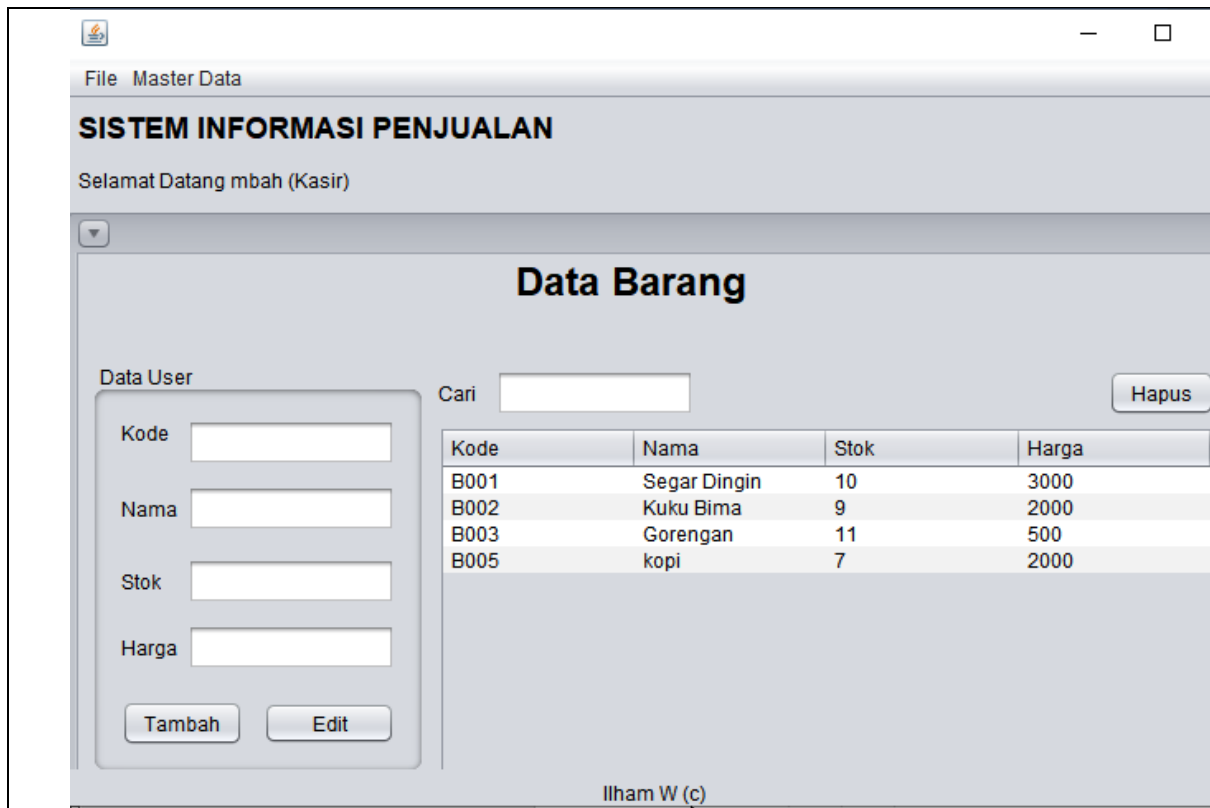
9. Panggil method loadData() pada konstruktor internal frame!

```

20     DefaultTableModel tabelmodel;
21
22 public ifrmMasterBarang() {
23     initComponents();
24     loadData();
25 }
26
59 public void ModelTabelBarang() {
60     tabelmodel = new DefaultTableModel();
61     jTable1.setModel(tabelmodel);
62     tabelmodel.addColumn("Kode");
63     tabelmodel.addColumn("Nama");
64     tabelmodel.addColumn("Stok");
65     tabelmodel.addColumn("Harga");
66     tabelmodel.getDataVector().removeAllElements();
67     tabelmodel.fireTableDataChanged();
68 }

```

10. Jalankan aplikasi dan tunjukkan hasilnya!



C3. Mencari data dari database dan menampilkannya di jTable

1. Buat method loadData(...) dengan parameter masukan key pada internal frame master user untuk mencari data User berdasarkan key!

```

39 public void loadData(String key){
40     ModelTabelUser();
41     ClassUser cu = new ClassUser();
42     cu.getDataUser(key);
43     int ndata = cu.getNumberDataUser();
44     Object[][] data = cu.getAllDataUser();
45     Object[] datal = new Object[3];
46     for (int i = 0; i < ndata; i++){
47         for (int j = 0; j < 3;j++){
48             datal[j] = data[i][j];
49             System.out.println(datal[j]);
50         }
51         tabelmodel.addRow(datal);
52     }
53 }

```

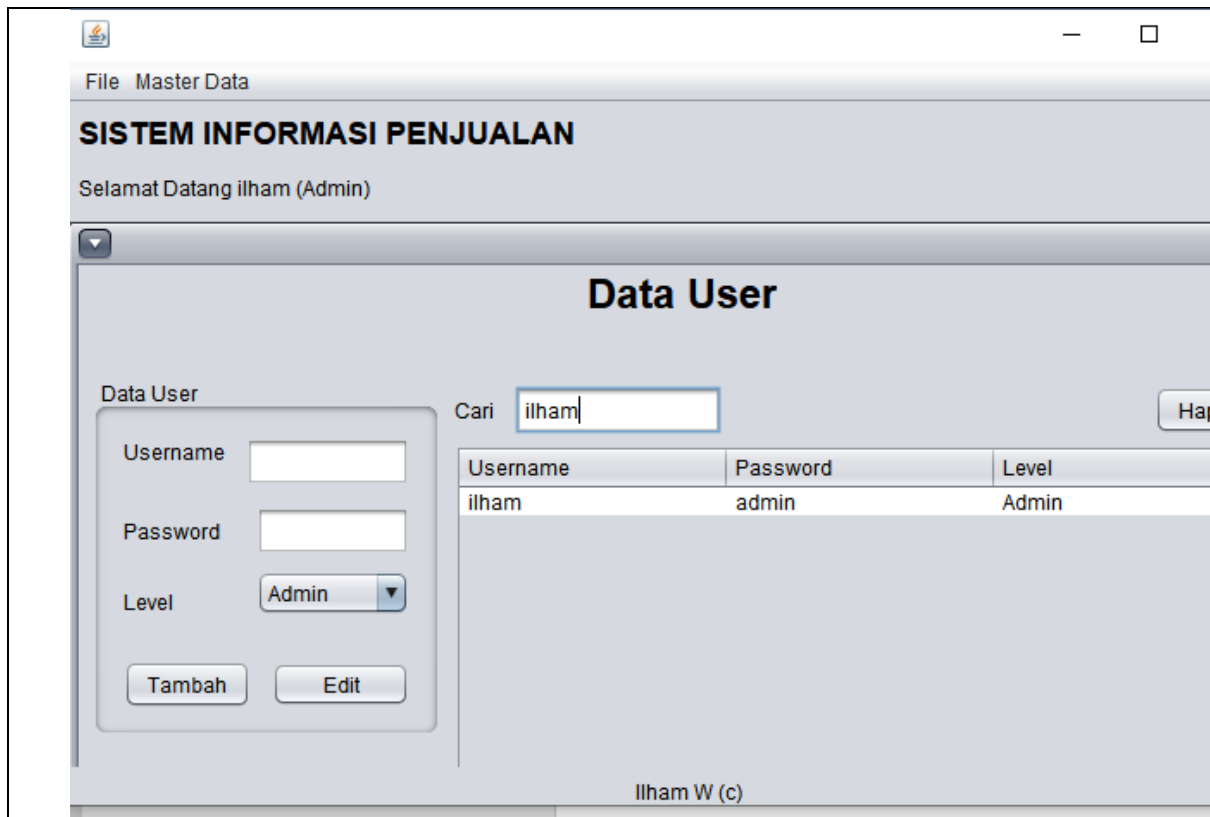
2. Panggil method loadData(...) pada aksi keyTyped pada JTextField pencarian!

```

277 private void txt_cariKeyTyped(java.awt.event.KeyEvent evt) {
278     // TODO add your handling code here:
279     loadData(txt_cari.getText());
280 }

```

3. Jalankan aplikasi dan tunjukkan hasilnya!



4. Buat method loadData(...) dengan parameter masukan key pada internal frame master barang untuk mencari data Barang berdasarkan key!

```

42
43 public void loadData(String key){
44     ModelTabelBarang();
45     ClassBarang cb = new ClassBarang();
46     cb.getDataBarang(key);
47     int ndata = cb.getNumberDataBarang();
48     Object[][] data = cb.getAllDataBarang();
49     Object[] datal = new Object[4];
50     for (int i = 0; i < ndata; i++){
51         for (int j = 0; j < 4;j++){
52             datal[j] = data[i][j];
53             System.out.println(datal[j]);
54         }
55         tabelmodel.addRow(datal);
56     }
57 }

```

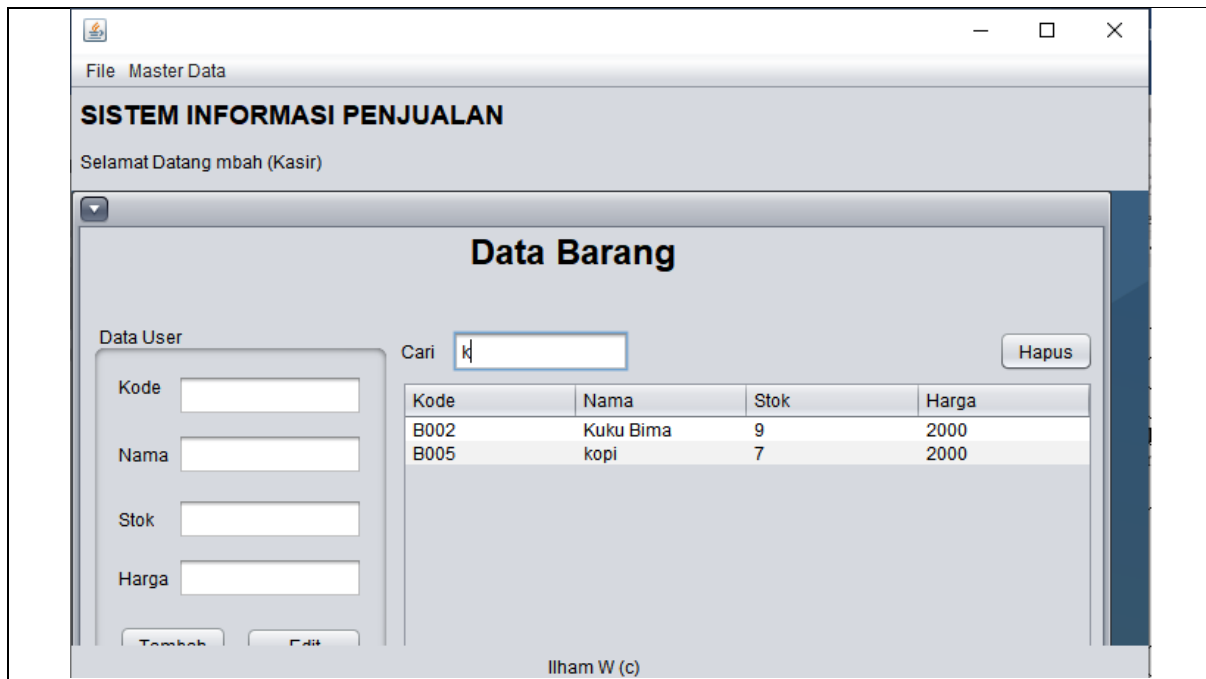
5. Panggil method loadData(...) pada aksi keyTyped pada JTextField pencarian!

```

297
298 private void txt_cariKeyTyped(java.awt.event.KeyEvent evt) {
299     // TODO add your handling code here:
300     loadData(txt_cari.getText());
301 }
302

```

6. Jalankan aplikasi dan tunjukkan hasilnya!



C4. Menambahkan data ke database dan menampilkannya di jTable

1. Buat method `clearData()` di internal frame master user untuk menghapus text di `textField`!

```

70 public void clearData() {
71     txt_username.setText("");
72     txt_password.setText("");
73 }
74

```

2. Tangkap masukan dari `textField` username, password, dan level dari `iframe` dan simpan ke dalam variabel!

```

282 private void btn_tambahActionPerformed(java.awt.event.ActionEvent evt) {
283     // TODO add your handling code here:
284     String uname = txt_username.getText();
285     String psw = txt_password.getText();
286     String level = combolevel.getSelectedItem().toString();
287
288     ClassUser cu = new ClassUser();
289     cu.InsertUser(uname, psw, level);
290
291     clearData();
292     loadData();
293 }

```

3. Buat objek `ClassUser` dan gunakan variabel masukan di no 2 sebagai parameter masukan untuk method `InsertUser()`!

```

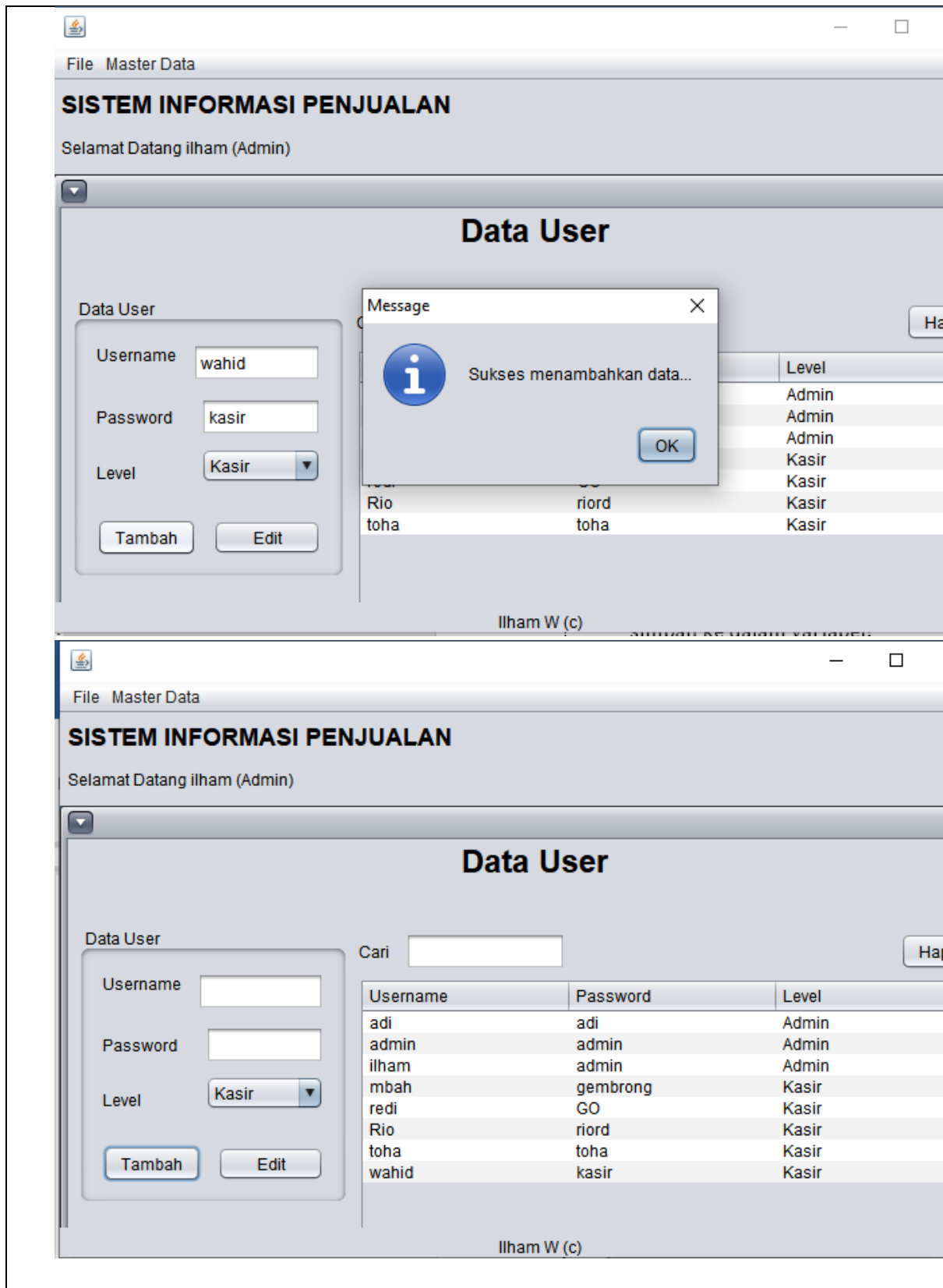
316 ClassUser cu = new ClassUser();
317 cu.UpdateUser(uname, psw, level);
318

```

4. Panggil method `clearData()` untuk menghapus text di `textField` dan `loadData()` untuk menampilkan kembali data dari database setelah data ditambahkan satu record!

```
318  
319         clearData();  
320         loadData();  
321     }
```

5. Jalankan aplikasi dan tunjukkan hasilnya!



6. Buat method `clearData()` di internal frame master barang untuk menghapus text di `(jTextField)`!

```

69
70 public void clearData() {
71     txt_kode.setText("");
72     txt_nama.setText("");
73     txt_stok.setText("");
74     txt_harga.setText("");
75 }
76

```

7. Tangkap masukan dari jTextField kode, nama, harga, dan stok dari iframe dan simpan ke dalam variabel!

```

303 private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
304     // TODO add your handling code here:
305     String kode = txt_kode.getText();
306     String nama = txt_nama.getText();
307     int stok = Integer.valueOf(txt_stok.getText());
308     int harga = Integer.valueOf(txt_harga.getText());
309 }

```

8. Buat objek ClassBarang dan gunakan variabel masukan di no 7 sebagai parameter masukan untuk method InsertBarang()!

```

310 ClassBarang cb = new ClassBarang();
311 cb.UpdateBarang(kode, nama, stok, harga);
312

```

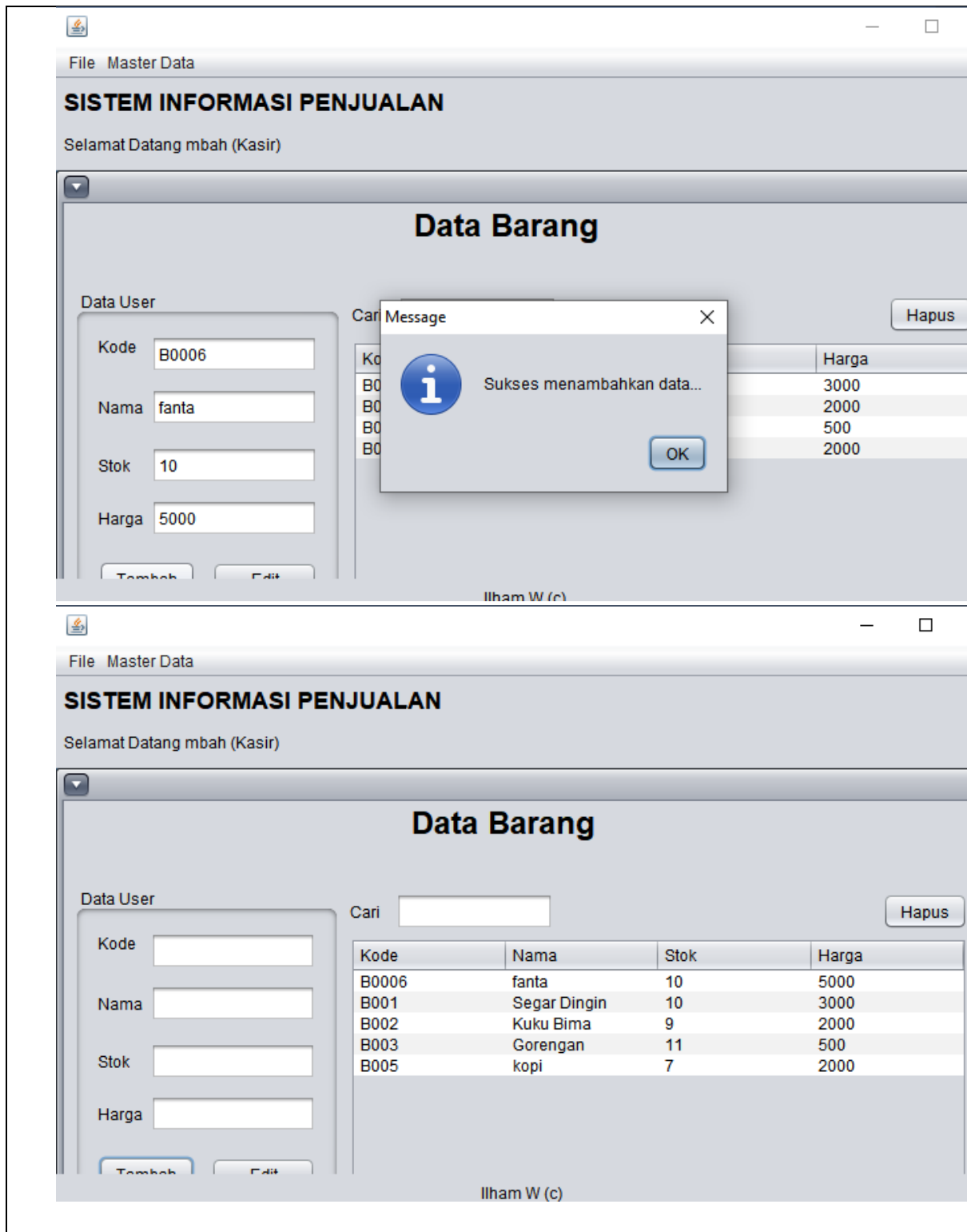
9. Panggil method clearData() untuk menghapus text di jTextField dan loadData() untuk menampilkan kembali data dari database setelah data ditambahkan satu record!

```

313 clearData();
314 loadData();
315 }

```

10. Jalankan aplikasi dan tunjukkan hasilnya!



C5. Mengedit data di database dan menampilkannya di jTable

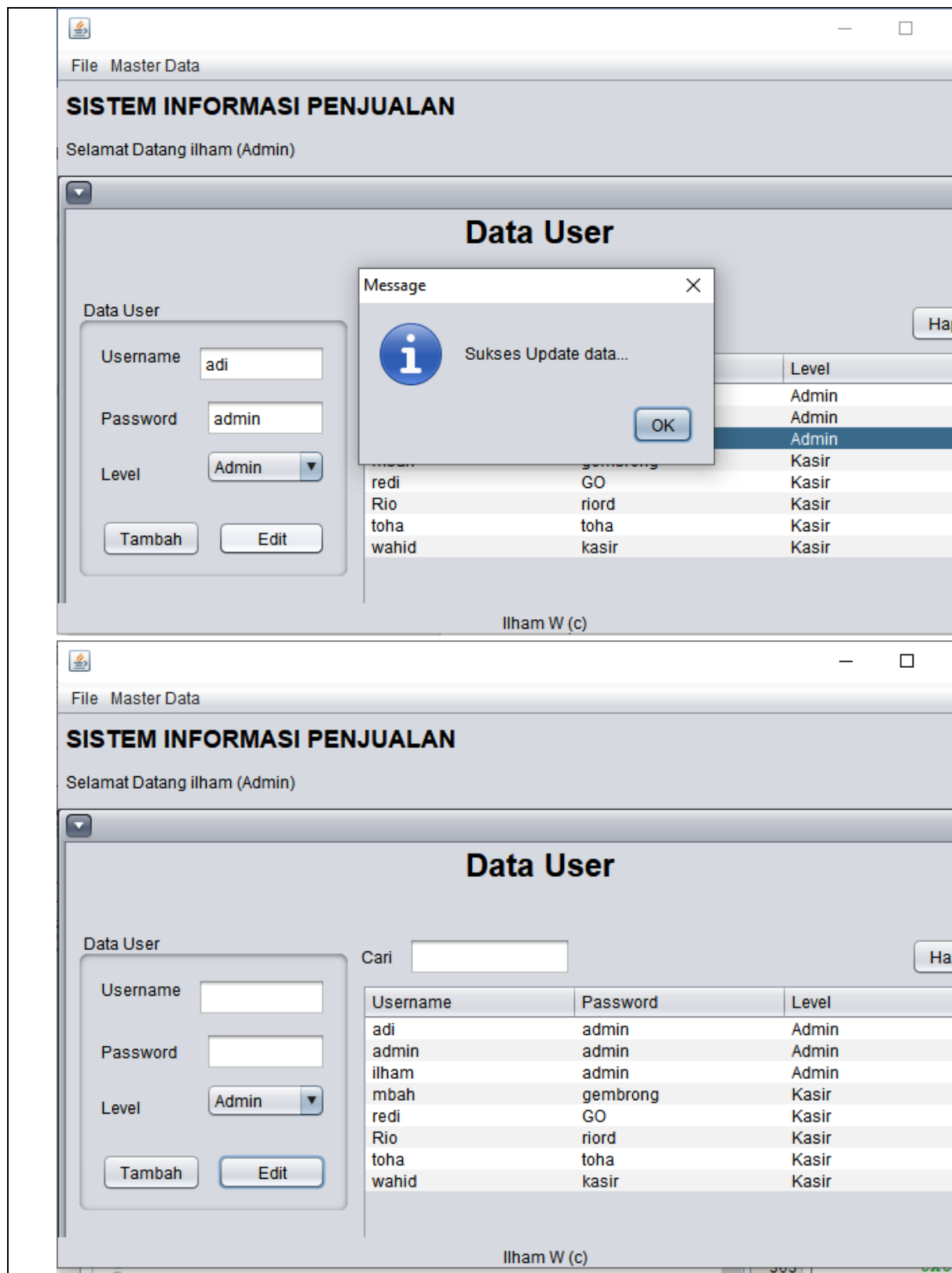
1. Buat aksi MouseClicked() pada iframe master user untuk memberi aksi pada jTable jika salah satu barisnya terpilih, maka value dari data tersebut akan tampil di jTextField!

```

295 private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
296     // TODO add your handling code here:
297     int i = jTable1.getSelectedRow(); //baris kesatu
298     if(i == -1){
299         return;
300     }
301
302     String username = (String) tabelmodel.getValueAt(i, 0); //kolom ke 0
303     txt_username.setText(username);
304     String password = (String) tabelmodel.getValueAt(i, 1); //kolom ke 1
305     txt_password.setText(password);
306     String level = (String) tabelmodel.getValueAt(i, 2); //kolom ke 2
307     combolevel.setSelectedItem(level);
308 }

```

2. Jalankan aplikasi dan tunjukkan hasilnya!



3. Pada button edit, beri aksi tangkap masukan dari jTextField username, password, dan level dari iframe dan simpan ke dalam variabel!

```

310 private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
311     // TODO add your handling code here:
312     String uname = txt_username.getText();
313     String psw = txt_password.getText();
314     String level = combolevel.getSelectedItem().toString();
315
316     ClassUser cu = new ClassUser();
317     cu.UpdateUser(uname, psw, level);
318
319     clearData();
320     loadData();
321 }
322

```

4. Buat objek ClassUser dan gunakan variabel masukan di no 3 sebagai parameter masukan untuk method UpdateUser()!

```

310 private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
311     // TODO add your handling code here:
312     String uname = txt_username.getText();
313     String psw = txt_password.getText();
314     String level = combolevel.getSelectedItem().toString();
315
316     ClassUser cu = new ClassUser();
317     cu.UpdateUser(uname, psw, level);
318
319     clearData();
320     loadData();
321 }
322

```

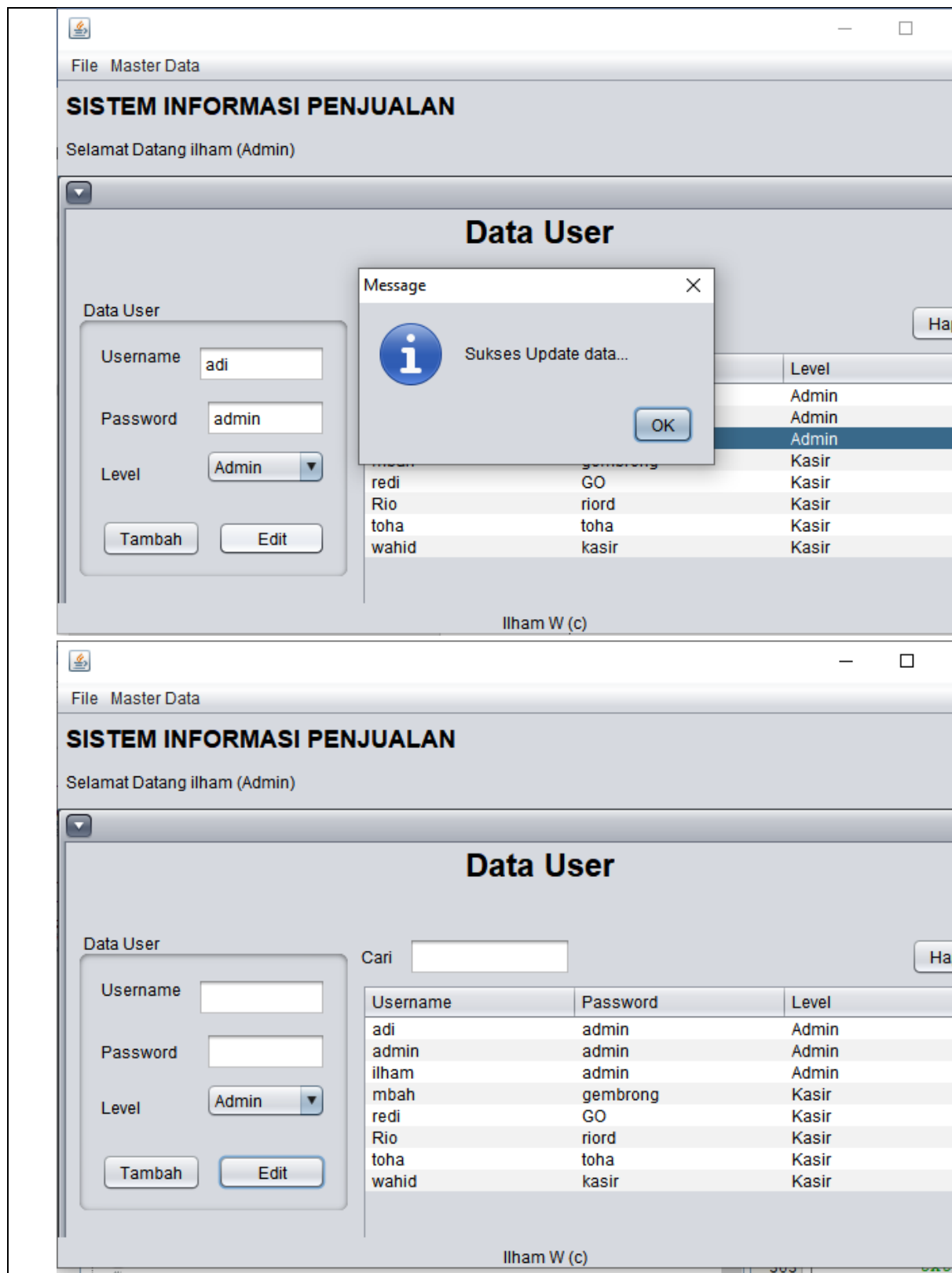
5. Panggil method clearData() untuk menghapus text di jTextField dan loadData() untuk menampilkan kembali data dari database setelah data ditambahkan satu record!

```

310 private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
311     // TODO add your handling code here:
312     String uname = txt_username.getText();
313     String psw = txt_password.getText();
314     String level = combolevel.getSelectedItem().toString();
315
316     ClassUser cu = new ClassUser();
317     cu.UpdateUser(uname, psw, level);
318
319     clearData();
320     loadData();
321 }
322

```

6. Jalankan aplikasi dan tunjukkan hasilnya!



7. Buat aksi `MouseClicked()` pada `iframe` master barang untuk memberi aksi pada `jTable` jika salah satu barisnya terpilih, maka value dari data tersebut akan tampil di `jTextField`!

```

316
317 private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
318     // TODO add your handling code here:
319     int i = jTable1.getSelectedRow(); //baris kesatu
320     if(i == -1){
321         return;
322     }
323
324     String kode = (String) tabelmodel.getValueAt(i, 0); //kolom ke 0
325     txt_kode.setText(kode);
326     String nama = (String) tabelmodel.getValueAt(i, 1); //kolom ke 1
327     txt_nama.setText(nama);
328     int stok = (int) tabelmodel.getValueAt(i, 2);
329     String stk = String.valueOf(stok);
330     txt_stok.setText(stk);
331     int harga = (int) tabelmodel.getValueAt(i, 3);
332     String hrg = String.valueOf(harga);
333     txt_harga.setText(hrg);
334 }

```

8. Jalankan aplikasi dan tunjukkan hasilnya!

SISTEM INFORMASI PENJUALAN

Selamat Datang mbah (Kasir)

Data Barang

Data User

Kode: B005

Nama: pula

Stok: 5

Harga: 6000

Tambah Edit

Cari:

Hapus

Kode	Nama	Stok	Harga
B0006	fanta	10	5000
B001	Segar Dingin	10	3000
B002	Kuku Bima	9	2000
B003	Gorengan	11	500
B005	kopi	7	2000

Ilham W (c)

9. Pada button edit, beri aksi tangkap masukan dari jTextField kode, nama, harga, dan stok dari iframe dan simpan ke dalam variabel!

```

336 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
337     // TODO add your handling code here:
338     String kode = txt_kode.getText();
339
340     ClassBarang cb = new ClassBarang();
341     cb.DeleteBarang(kode);
342
343     clearData();
344     loadData();
345 }

```

10. Buat objek ClassBarang dan gunakan variabel masukan di no 9 sebagai parameter masukan untuk method UpdateBarang()!

```

336 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
337     // TODO add your handling code here:
338     String kode = txt_kode.getText();
339
340     ClassBarang cb = new ClassBarang();
341     cb.DeleteBarang(kode);
342
343     clearData();
344     loadData();
345 }

```

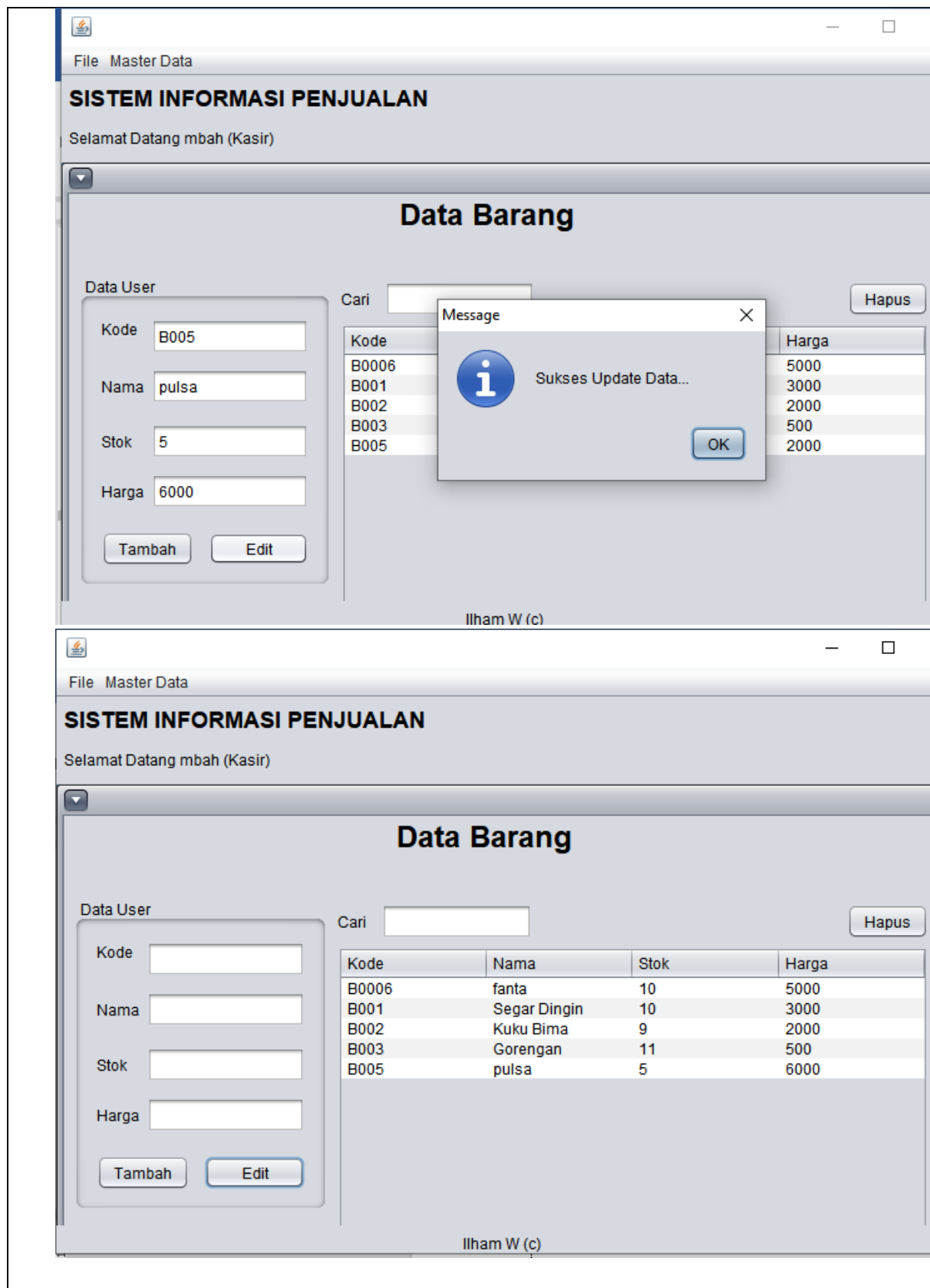
11. Panggil method clearData() untuk menghapus text di jTextField dan loadData() untuk menampilkan kembali data dari database setelah data ditambahkan satu record!

```

336 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
337     // TODO add your handling code here:
338     String kode = txt_kode.getText();
339
340     ClassBarang cb = new ClassBarang();
341     cb.DeleteBarang(kode);
342
343     clearData();
344     loadData();
345 }

```

12. Jalankan aplikasi dan tunjukkan hasilnya!



C6. Menghapus data di database dan menampilkannya di jTable

1. Pada button hapus di iframe master data user, beri aksi tangkap masukan dari jTextField username, password, dan level dari iframe dan simpan ke dalam variabel!

```
323 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {  
324     // TODO add your handling code here:  
325     String uname = txt_username.getText();  
326  
327     ClassUser cu = new ClassUser();  
328     cu.DeleteUser(uname);  
329  
330     clearData();  
331     loadData();  
332 }  
333
```

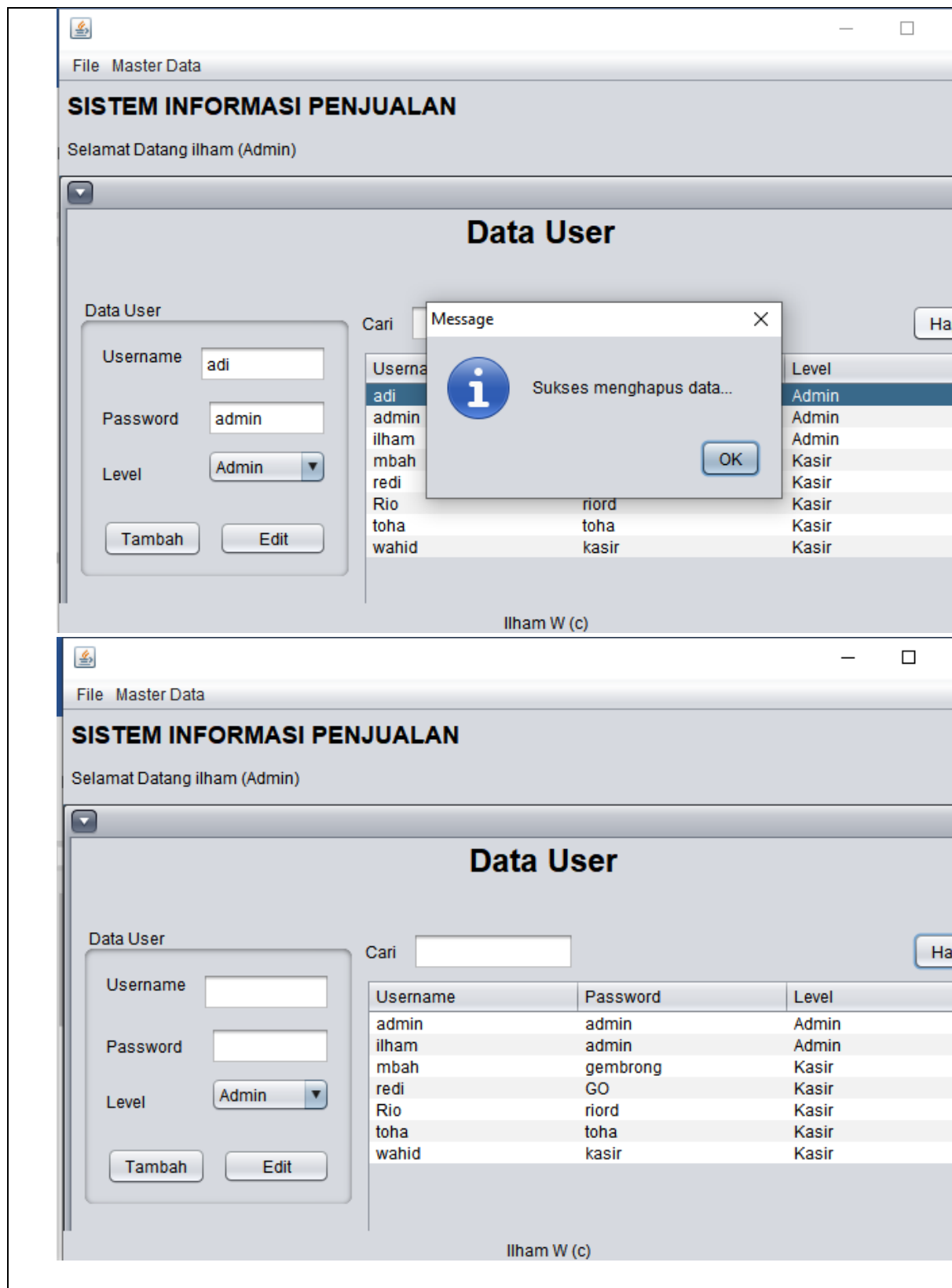
2. Buat objek ClassUser dan gunakan variabel masukan di no 1 sebagai parameter masukan untuk method DeleteUser()!

```
323 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {  
324     // TODO add your handling code here:  
325     String uname = txt_username.getText();  
326  
327     ClassUser cu = new ClassUser();  
328     cu.DeleteUser(uname);  
329  
330     clearData();  
331     loadData();  
332 }  
333
```

3. Panggil method clearData() untuk menghapus text di jTextField dan loadData() untuk menampilkan kembali data dari database setelah data ditambahkan satu record!

```
323 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {  
324     // TODO add your handling code here:  
325     String uname = txt_username.getText();  
326  
327     ClassUser cu = new ClassUser();  
328     cu.DeleteUser(uname);  
329  
330     clearData();  
331     loadData();  
332 }  
333
```

4. Jalankan aplikasi dan tunjukkan hasilnya!



5. Pada button hapus di iframe master data barang, beri aksi tangkap masukan dari jTextField kode, nama, harga, dan stok dari iframe dan simpan ke dalam variabel!


```

335
336 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
337
338     // TODO add your handling code here:
339     String kode = txt_kode.getText();
340
341     ClassBarang cb = new ClassBarang();
342     cb.DeleteBarang(kode);
343
344     clearData();
345     loadData();
346 }

```

6. Buat objek ClassUser dan gunakan variabel masukan di no 5 sebagai parameter masukan untuk method DeleteBarang()!

```

335
336 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
337
338     // TODO add your handling code here:
339     String kode = txt_kode.getText();
340
341     ClassBarang cb = new ClassBarang();
342     cb.DeleteBarang(kode);
343
344     clearData();
345     loadData();
346 }

```

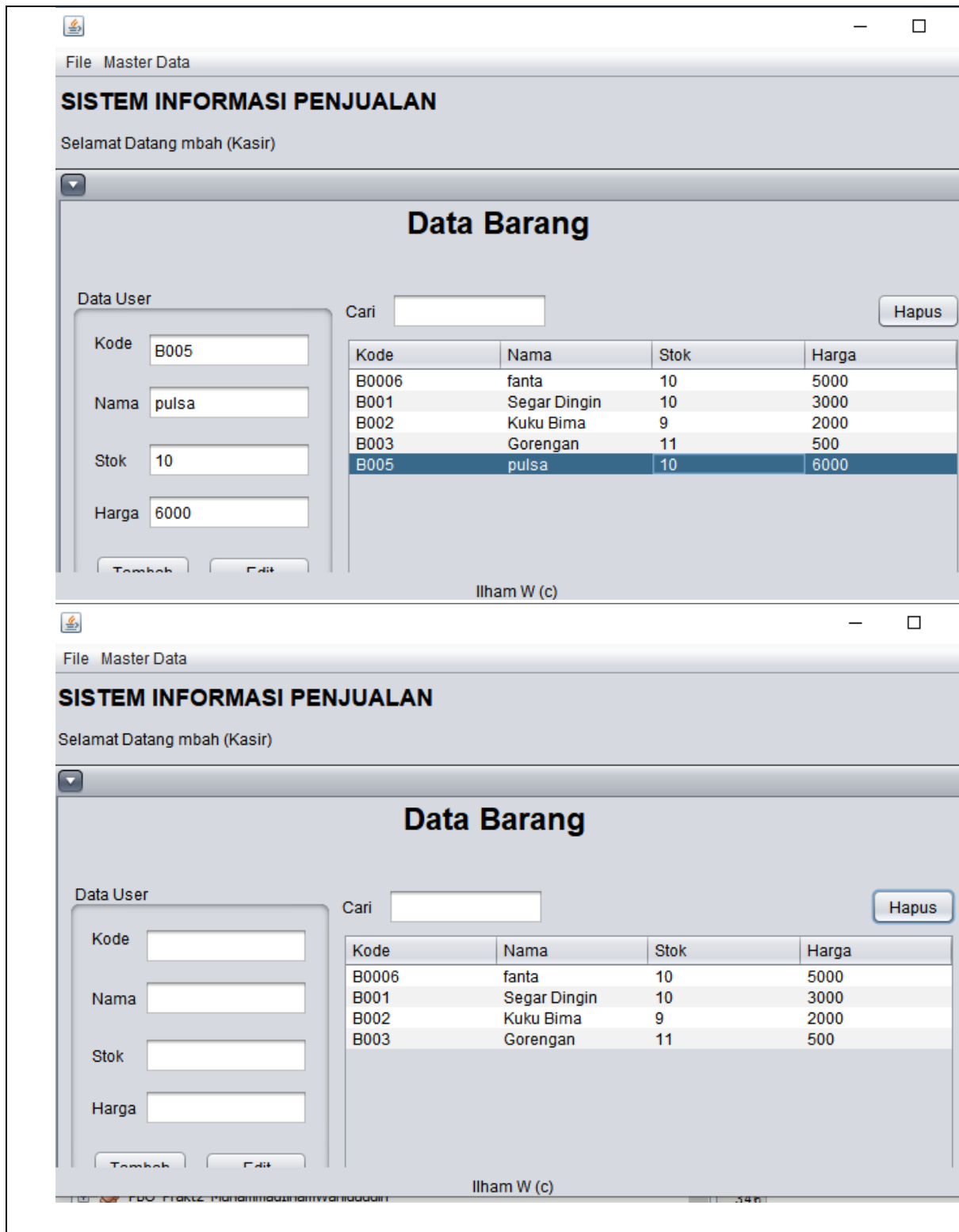
7. Panggil method clearData() untuk menghapus text di jTextField dan loadData() untuk menampilkan kembali data dari database setelah data ditambahkan satu record!

```

335
336 private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
337
338     // TODO add your handling code here:
339     String kode = txt_kode.getText();
340
341     ClassBarang cb = new ClassBarang();
342     cb.DeleteBarang(kode);
343
344     clearData();
345     loadData();
346 }

```

8. Jalankan aplikasi dan tunjukkan hasilnya!



D. TUGAS

Buat rancangan use case dan diagram activity dari sebuah aplikasi diatas.

Hint:

- Terdapat dua aktor yaitu admin dan kasir

- Admin dan kasir dapat login dan logout
- Admin dapat melihat data user
- Admin dapat menambah data user
- Admin dapat mengedit data user
- Admin dapat menghapus data user
- Kasir dapat melihat data barang
- Kasir dapat menambah data barang
- Kasir dapat mengedit data barang
- Kasir dapat menghapus data barang

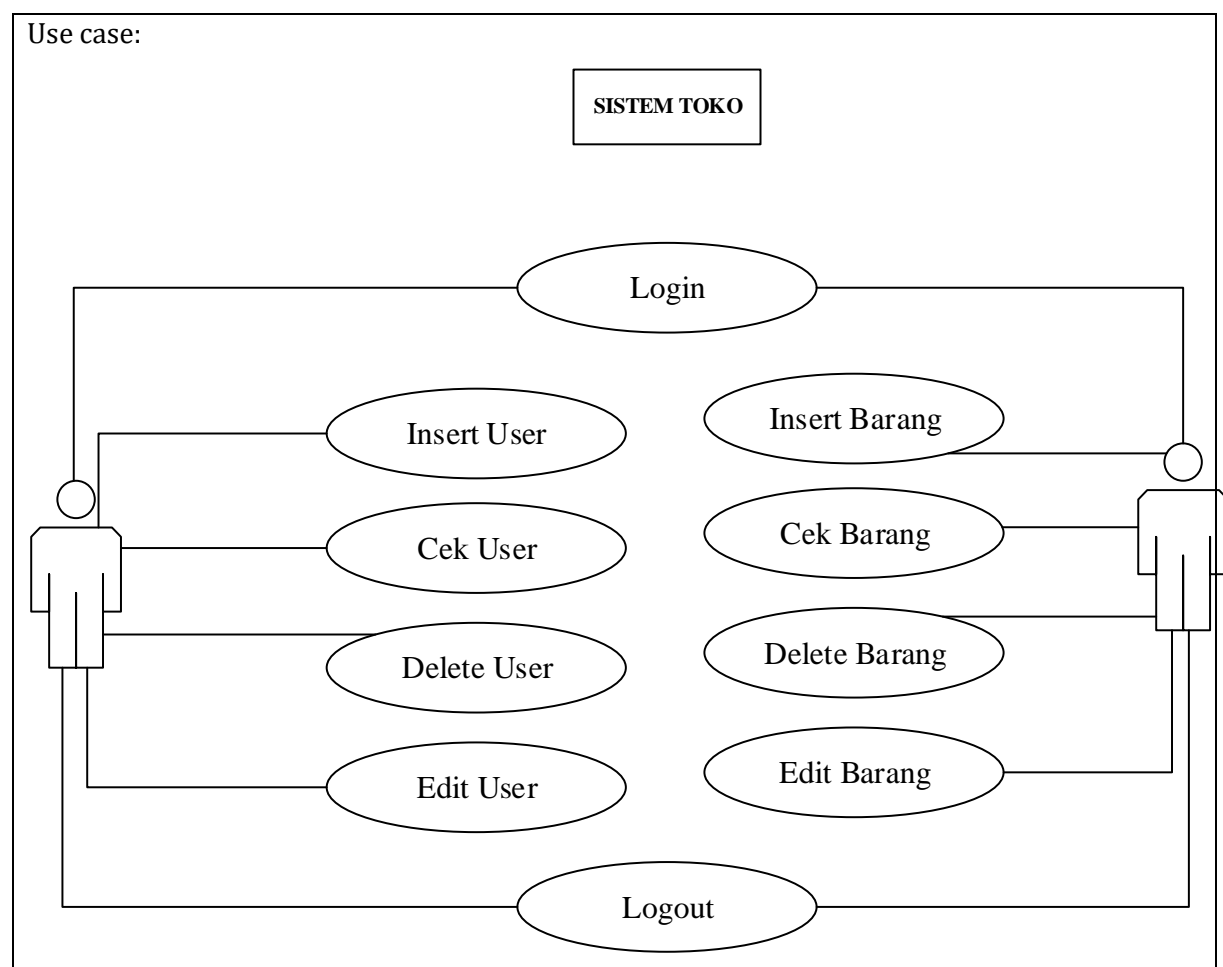


Diagram activity:

