

PEMROGRAMAN KOMPUTER 2



NAMA : Solehudin Alfah Rezi

Kelas : 4D

NIM : 18090037

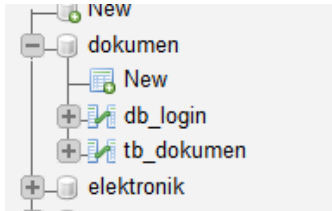
Dosen : M.Nishom, M.Kom.

POLITEKNIK HARAPAN BERSAMA
TEGAL

MEMBUAT MANUAL BOOK ARSIP DOKUMENTASI

1. Membuat Data Base

- Buka xampp kemudian aktifkan Apache dan My sql dan pilih admin untuk masuk ke php my admin
- Kemudian buat basis data seperti di bawah ini



- Bagian db_login :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(1)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	username	varchar(30)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	password	varchar(40)	latin1_swedish_ci	No	None			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Fulltext A

Remove from central columns

- Bagian tb_dokumen :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	kode	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	2	nama	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	kategori	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	4	lokasi	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	5	deskripsi	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	6	tanggal	date		No	None			Change Drop More

2. Menyiapkan Project

- Buka Netbeans, kemudian buat project baru dengan nama Dokumen
- Setelah membuat project baru, buatlah package baru dengan nama Apps

3. Membuat kelas

- Buat class konfigurasi dan buat JFrame Login dokumen
- Class konfigurasi untuk menyambungkan data base, kemudian di class konfigurasi diisi :
package Apps;

```
Start Page x Login.java x dokumen.java x konfig.java x
Source History
5  */
6  package Apps;
7
8  import java.sql.Connection;
9  import java.sql.DriverManager;
10 import java.sql.SQLException;
11
12 /**
13  *
14  * @author Asus
15  */
16 public class konfig {
17     private static Connection MySQLConfig;
18
19     public static Connection configDB () throws SQLException{
20         try {
21             String url = "jdbc:mysql://localhost:3306/dokumen";
22             String user = "root";
23             String pass = "";
24
25             DriverManager.registerDriver(new com.mysql.jdbc.Driver());
26             MySQLConfig = DriverManager.getConnection(url , user , pass);
27
28         } catch (Exception e) {
29             System.out.println("Koneksi Ke Database Gagal"+e.getMessage());
30         }
31         return MySQLConfig;
32     }
33 }
34
```

➤ Package Apps Login

```
Start Page x Login.java
Source Design History
5  */
6  package Apps;
7
8  import java.awt.Frame;
9  import java.security.MessageDigest;
10 import java.security.NoSuchAlgorithmException;
11 import java.sql.Connection;
12 import java.sql.ResultSet;
13 import java.sql.Statement;
14 import javax.swing.JOptionPane;
15
16 /**
17  *
18  * @author Asus
19  */
20 public class Login extends javax.swing.JFrame {
21
22     /**
23      * Creates new form Login
24      */
25     public Login() {
26         initComponents();
27     }
28
29     /**
30      * This method is called from within the constructor to initialize the form.
31      * WARNING: Do NOT modify this code. The content of this method is always
32      * regenerated by the Form Editor.
33      */
34     @SuppressWarnings("unchecked")
35     // Generated Code

```

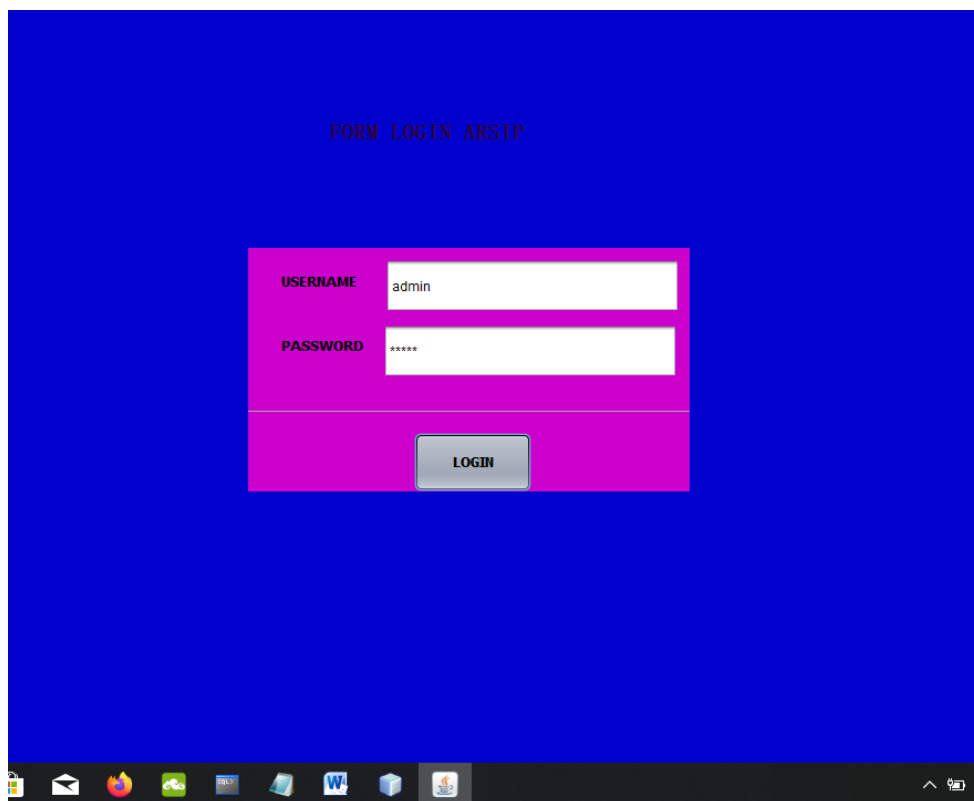
```
Start Page x Login.java x
Source Design History
33  */
34  @SuppressWarnings("unchecked")
35  Generated Code
175
176  private void txtusernameActionPerformed(java.awt.event.ActionEvent evt) {
177      // TODO add your handling code here:
178  }
179
180  private void txtpasswordActionPerformed(java.awt.event.ActionEvent evt) {
181      // TODO add your handling code here:
182  }
183
184  private void btnloginActionPerformed(java.awt.event.ActionEvent evt) {
185      doLogin();
186  }
187
188  /**
189   * @param args the command line arguments
190   */
191  public static void main(String args[]) {
192      /* Set the Nimbus look and feel */
193      Look and feel setting code (optional)
214
215      /* Create and display the form */
216      java.awt.EventQueue.invokeLater(new Runnable() {
217          public void run() {
218              Login login = new Login();
219              login.setExtendedState(Frame.MAXIMIZED_BOTH);
220              login.setVisible(true);
221          }
222      });
};
```

```
Start Page x Login.java x
Source Design History
222  });
223  }
224
225  // Variables declaration - do not modify
226  private javax.swing.JButton btnlogin;
227  private javax.swing.JLabel jLabel1;
228  private javax.swing.JLabel jLabel2;
229  private javax.swing.JLabel jLabel3;
230  private javax.swing.JPanel jPanel1;
231  private javax.swing.JPanel jPanel2;
232  private javax.swing.JPanel jPanel3;
233  private javax.swing.JSeparator jSeparator1;
234  private javax.swing.JPasswordField txtpassword;
235  private javax.swing.JTextField txtusername;
236  // End of variables declaration
237
238  public void doLogin() {
239      btnlogin.setEnabled(false);
240      String user = txtusername.getText();
241      char[] pass = txtpassword.getPassword();
242      if (user.isEmpty()) {
243          JOptionPane.showMessageDialog(this, "Username harus diisi");
244          txtusername.requestFocus();
245      } else if (pass.length == 0) {
246          JOptionPane.showMessageDialog(this, "Password harus diisi");
247          txtpassword.requestFocus();
248      } else {
249          String password = new String(pass);
250          String md5_pass = md5(password);
251          try {
252              Connection c = konfig.configDB();
```

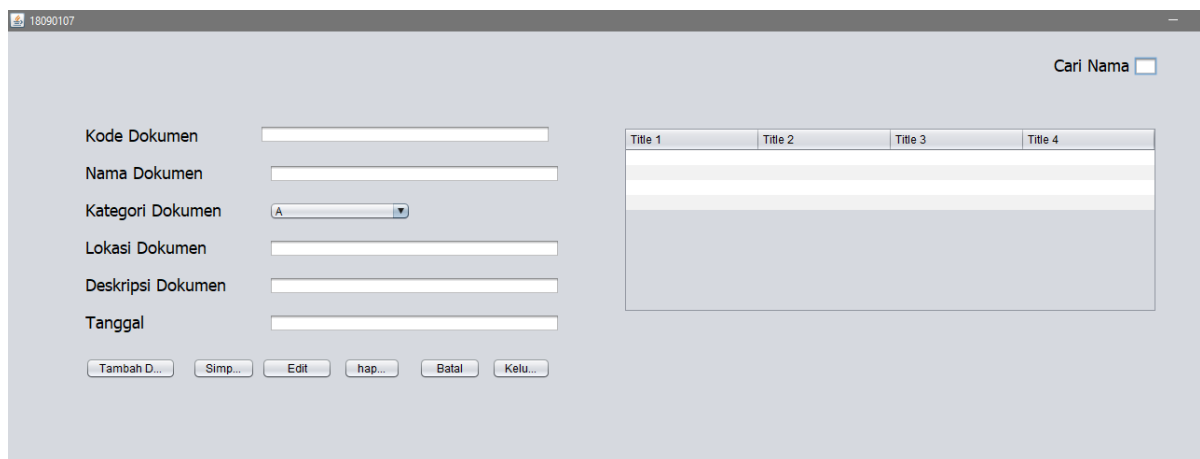
```
Start Page x Login.java x
Source Design History
243 JOptionPane.showMessageDialog(this, "Username harus diisi");
244 txtusername.requestFocus();
245 } else if (pass.length == 0) {
246     JOptionPane.showMessageDialog(this, "Password harus diisi");
247     txtpassword.requestFocus();
248 } else {
249     String password = new String(pass);
250     String md5_pass = md5(password);
251     try {
252         Connection c = konfig.configDB();
253         Statement st = c.createStatement();
254         String q = " SELECT * FROM db_login "
255             + "WHERE username = '" + user + "' "
256             + "AND password = '" + md5_pass + "'";
257         ResultSet rs = st.executeQuery(q);
258         int n = 0;
259         while (rs.next()) {
260             String u = rs.getString("username");
261             String p = rs.getString("password");
262             n++;
263         }
264         if (n == 0) {
265             JOptionPane.showMessageDialog(this, "Username/password salah!");
266             txtusername.requestFocus();
267             btnlogin.setEnabled(true);
268         } else {
269             dokumen kar = new dokumen();
270             this.setVisible(false);
271             kar.setExtendedState(Frame.MAXIMIZED_BOTH);
272             kar.setVisible(true);
273         }
274     }
275 }
```

```
Start Page x Login.java x
Source Design History
267 btnlogin.setEnabled(true);
268 } else {
269     dokumen kar = new dokumen();
270     this.setVisible(false);
271     kar.setExtendedState(Frame.MAXIMIZED_BOTH);
272     kar.setVisible(true);
273 }
274 } catch (Exception e) {
275     System.out.println(e.getMessage());
276 }
277 }
278 }
279
280 public static String md5(String data) {
281     String hashed = "";
282     try {
283         MessageDigest md5 = MessageDigest.getInstance("MD5");
284         md5.update(data.getBytes());
285         byte[] byteData = md5.digest();
286         StringBuilder sb = new StringBuilder();
287         for (int i = 0; i < byteData.length; i++) {
288             sb.append(Integer.toString((byteData[i] & 0xff)
289                 + 0x100, 16).substring(1));
290         }
291         hashed = sb.toString();
292     } catch (NoSuchAlgorithmException e) {
293     }
294     return hashed;
295 }
296 }
297 }
```

- Hasil Project Login



- Membuat Jframe Dokumen



- Package App Dokumen

```
Start Page x Login.java x dokumen.java x
Source Design History
6 package Apps;
7
8 import java.awt.HeadlessException;
9 import java.sql.Connection;
10 import java.sql.SQLException;
11 import javax.swing.JOptionPane;
12 import javax.swing.table.DefaultTableModel;
13
14
15 /**
16  *
17  * @author Asus
18  */
19 public class dokumen extends javax.swing.JFrame {
20
21     private void kosongkan_from() {
22         txtkodedokumen.setEditable(true);
23         txtkodedokumen.setText(null);
24         txtnamadokumen.setText(null);
25         cmbkatagoridokumen.setSelectedItem(this);
26         txtlokasi.setText(null);
27         txtdeskripsi.setText(null);
28         txttanggal.setText(null);
29     }
30
31     private void tampilkan_data() {
32         DefaultTableModel model = new DefaultTableModel();
33         model.addColumn("No.");
34         model.addColumn("Kode Dokumen");
35         model.addColumn("Nama Dokumen");
36         model.addColumn("Kategori Dokumen");
37         model.addColumn("Lokasi Dokumen");
38     }
39 }
```

```
Start Page x Login.java x dokumen.java x
Source Design History
31 private void tampilkan_data() {
32     DefaultTableModel model = new DefaultTableModel();
33     model.addColumn("No.");
34     model.addColumn("Kode Dokumen");
35     model.addColumn("Nama Dokumen");
36     model.addColumn("Kategori Dokumen");
37     model.addColumn("Lokasi Dokumen");
38     model.addColumn("Deskripsi Dokumen");
39     model.addColumn("Tanggal");
40
41     String cari = txtcari.getText();
42     try {
43         int no = 1;
44         String sql = "SELECT *FROM tb_dokumen WHERE nama LIKE '%" + cari + "%' ";
45         java.sql.Connection conn = (Connection)konfig.configDB();
46         java.sql.Statement stm = conn.createStatement();
47         java.sql.ResultSet res = stm.executeQuery(sql);
48
49         while (res.next()) {
50             model.addRow(new Object[] {no++, res.getString(1),
51                 res.getString(2), res.getString(3), res.getString(4), res.getString(5), res.getString(6)});
52             tabledokumen.setModel(model);
53         } catch (Exception e) {
54             System.out.println("Error : " + e.getMessage());
55         }
56     }
57
58     public dokumen() {
59         initComponents();
60         tampilkan_data();
61         kosongkan_from();
62     }
63 }
```

```
Start Page x Login.java x dokumen.java x
Source Design History
59 initComponents();
60 tampilkan_data();
61 kosongkan_from();
62 }
63
64 /**
65  * This method is called from within the constructor to initialize the form.
66  * WARNING: Do NOT modify this code. The content of this method is always
67  * regenerated by the Form Editor.
68  */
69 @SuppressWarnings("unchecked")
70 Generated Code
254
255 private void btnBatalActionPerformed(java.awt.event.ActionEvent evt) {
256     kosongkan_from();
257 }
258
259 private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {
260     System.exit(0);
261 }
262
263 private void btnTambahDataActionPerformed(java.awt.event.ActionEvent evt) {
264     try {
265         String sql = "INSERT INTO tb_dokumen VALUES ('"+txtkodedokumen.getText()+"','"+txtnamadokumen.getTe
266         java.sql.Connection conn = (Connection)konfig.configDB();
267         java.sql.PreparedStatement pstm = conn.prepareStatement(sql);
268         pstm.execute();
269         JOptionPane.showMessageDialog(null, "Proses simpan Data berhasil");
270         tampilkan_data();
271         kosongkan_from();
272     } catch (HeadlessException | SQLException e) {
```

```
Start Page x Login.java x dokumen.java x
Source Design History
268 pstm.execute();
269 JOptionPane.showMessageDialog(null, "Proses simpan Data berhasil");
270 tampilkan_data();
271 kosongkan_from();
272 } catch (HeadlessException | SQLException e) {
273     JOptionPane.showMessageDialog(this, e.getMessage());
274 }
275
276 txtkodedokumen.setFocusable(true);
277 }
278
279 private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
280     try {
281         String sql = "INSERT INTO tb_dokumen VALUES ('"+txtkodedokumen.getText()+"','"+txtnamadokumen.getTe
282         java.sql.Connection conn = (Connection)konfig.configDB();
283         java.sql.PreparedStatement pstm = conn.prepareStatement(sql);
284         pstm.execute();
285         JOptionPane.showMessageDialog(null, "Proses simpan Data berhasil");
286         tampilkan_data();
287         kosongkan_from();
288     } catch (HeadlessException | SQLException e) {
289         JOptionPane.showMessageDialog(this, e.getMessage());
290     }
291 }
292
293
294 private void tabledokumenMouseClicked(java.awt.event.MouseEvent evt) {
295     int baris = tabledokumen.rowAtPoint(evt.getPoint());
296     String nik = tabledokumen.getValueAt(baris, 1).toString();
297     txtkodedokumen.setText(nik);
298 }
```



```
Start Page x Login.java x dokumen.java x
Source Design History
298
299     String nama = tabledokumen.getValueAt(baris, 2).toString();
300     txtnamadokumen.setText(nama);
301
302     String jab = tabledokumen.getValueAt(baris, 3).toString();
303     cmbkatagoridokumen.setSelectedItem(jab);
304
305     String alamat = tabledokumen.getValueAt(baris, 4).toString();
306     txtlokasi.setText(alamat);
307
308     String email = tabledokumen.getValueAt(baris, 5).toString();
309     txtdeskripsi.setText(email);
310
311     String telp = tabledokumen.getValueAt(baris, 6).toString();
312     txttanggal.setText(telp);
313 }
314
315 private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
316     try {
317         String sql = "UPDATE tb_dokumen SET kode = '"+txtkodedokumen.getText()+"',nama = '"+txtnamadokumen.
318         java.sql.Connection conn = (Connection)konfig.configDB();
319         java.sql.PreparedStatement pstm = conn.prepareStatement(sql);
320         pstm.execute();
321         JOptionPane.showMessageDialog(null, "Edit Data Berhasil");
322     } catch (HeadlessException | SQLException e) {
323         JOptionPane.showMessageDialog(this, e.getMessage());
324     }
325     tampilkan_data();
326     kosongkan_from();
327 }
328
```

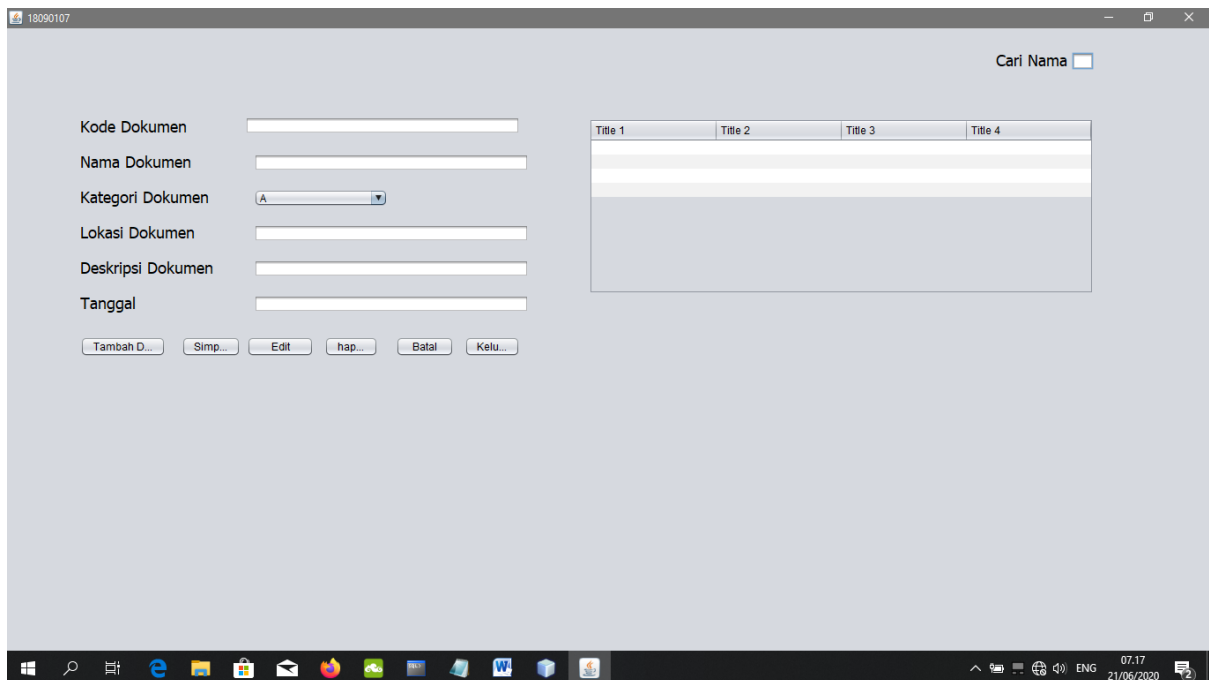
```
Start Page x Login.java x dokumen.java x
Source Design History
342
343 private void txtcariActionPerformed(java.awt.event.ActionEvent evt) {
344     // TODO add your handling code here:
345 }
346
347 private void txtcariKeyPressed(java.awt.event.KeyEvent evt) {
348     tampilkan_data();
349 }
350
351 private void txtnamadokumenActionPerformed(java.awt.event.ActionEvent evt) {
352     // TODO add your handling code here:
353 }
354
355 /**
356  * @param args the command line arguments
357  */
358 public static void main(String args[]) {
359     /* Set the Nimbus look and feel */
360     Look and feel setting code (optional)
361     //</editor-fold>
362
363     /* Create and display the form */
364     java.awt.EventQueue.invokeLater(new Runnable() {
365         public void run() {
366             new dokumen().setVisible(true);
367         }
368     });
369 }
370
371 // Variables declaration - do not modify
372 private javax.swing.JButton btnBatal;
```

```

388     });
389 }
390
391 // Variables declaration - do not modify
392 private javax.swing.JButton btnBatal;
393 private javax.swing.JButton btnEdit;
394 private javax.swing.JButton btnHapus;
395 private javax.swing.JButton btnKeluar;
396 private javax.swing.JButton btnSimpan;
397 private javax.swing.JButton btnTambahData;
398 private javax.swing.JComboBox<String> cmbkatagoridokumen;
399 private javax.swing.JLabel jLabel1;
400 private javax.swing.JLabel jLabel2;
401 private javax.swing.JLabel jLabel3;
402 private javax.swing.JLabel jLabel4;
403 private javax.swing.JLabel jLabel5;
404 private javax.swing.JLabel jLabel6;
405 private javax.swing.JLabel jLabel7;
406 private javax.swing.JLabel jLabel8;
407 private javax.swing.JLabel jLabel9;
408 private javax.swing.JScrollPane jScrollPane1;
409 private javax.swing.JTable tabledokumen;
410 private javax.swing.JTextField txtcari;
411 private javax.swing.JTextField txtdeskripsi;
412 private javax.swing.JTextField txtkodedokumen;
413 private javax.swing.JTextField txtlokasi;
414 private javax.swing.JTextField txtnamadokumen;
415 private javax.swing.JTextField txttanggal;
416 // End of variables declaration
417 }
418

```

- Hasil Codingan



4. Hasil Uji Coba

❖ Jframe Login

FORM LOGIN ARSIP

USERNAME admin

PASSWORD *****

LOGIN

❖ Hasil Login

Cari Nama

Kode Dokumen

Nama Dokumen

Kategori Dokumen

Lokasi Dokumen

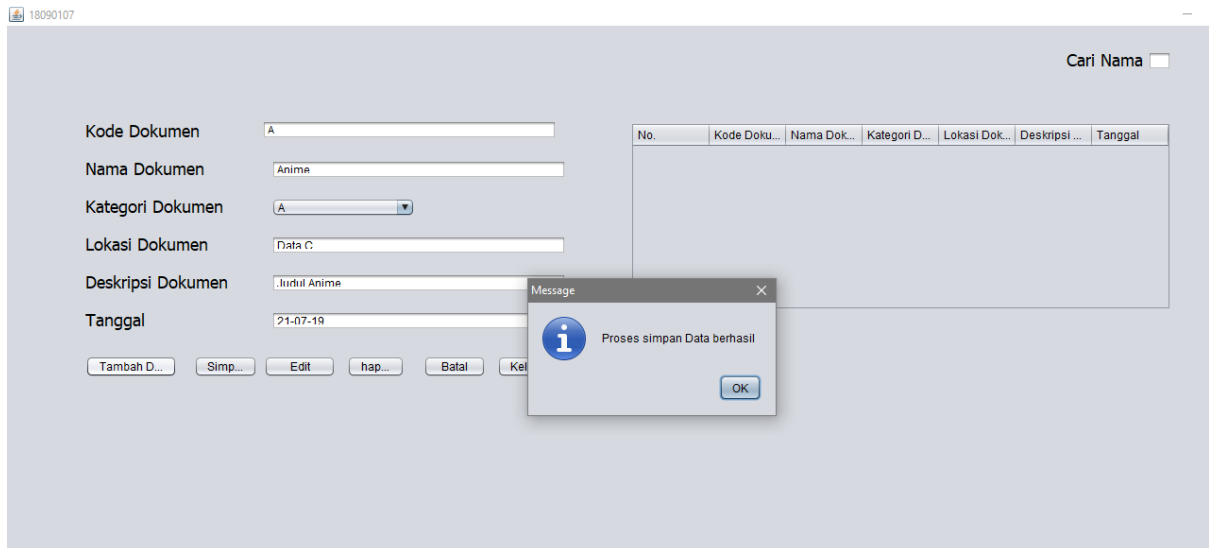
Deskripsi Dokumen

Tanggal

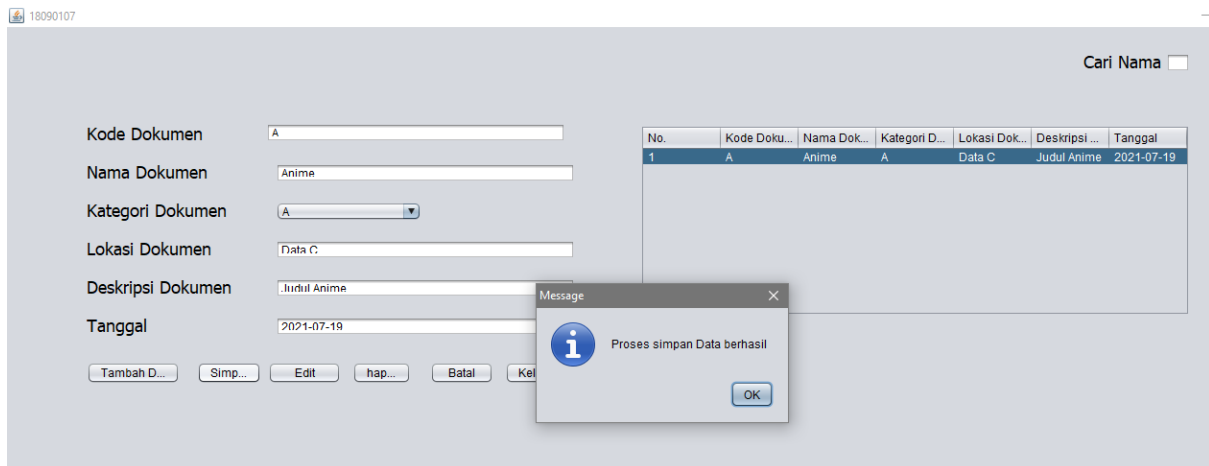
Tambah D... Simp... Edit hap... Batal Kelu...

Title 1	Title 2	Title 3	Title 4

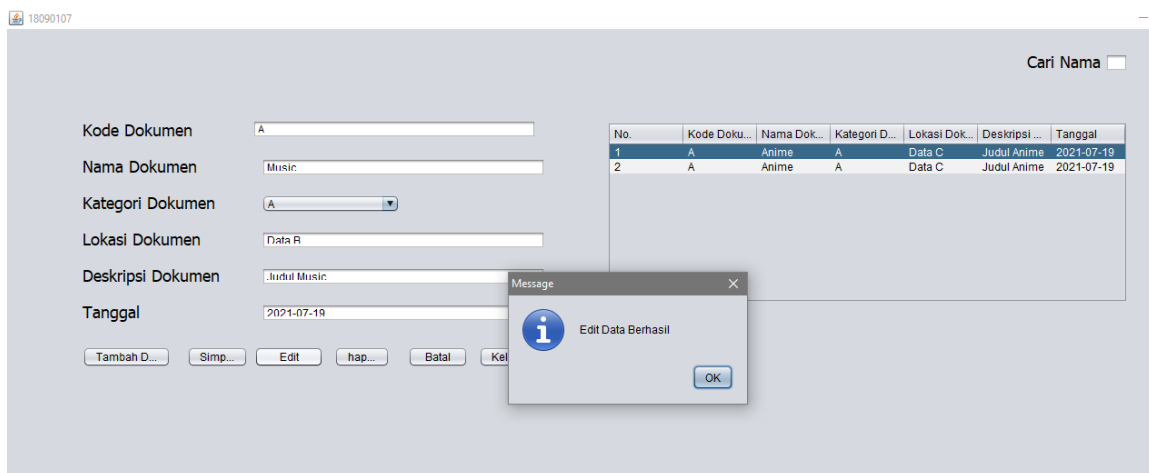
❖ Tambah Data



❖ Simpan Data



❖ Edit Data



❖ Hapus Data

18090107

Cari Nama

Kode Dokumen

A

Nama Dokumen

Music

Kategori Dokumen

A

Lokasi Dokumen

Data B

Deskripsi Dokumen

Judul Music

Tanggal

2021-07-19

Tambah D...

Simp...

Edit

hap...

Batal

Kel...

No.	Kode Doku...	Nama Dok...	Kategori D...	Lokasi Dok...	Deskripsi ...	Tanggal
1	A	Music	A	Data B	Judul Music	2021-07-19
2	A	Music	A	Data B	Judul Music	2021-07-19

Message

i

Hapus Data Berhasil

OK