

# **LAPORAN TUGAS AKHIR**

## **SISTEM APLIKASI PENYEWAAN MOBIL BERBASIS JAVA GUI**

Disusun Untuk Memenuhi Tugas Akhir Mata Kuliah Pemrograman  
Berorientasi Objek

Dosen Pengampu: Ratna Mufidah, S. Kom., M. Kom.



Nama Kelompok : Hawantar

Muhammad Hadi Royan Firdaus      2110631170079

Tarida Grace Wahyuni Margaretha S.   2110631170109

Wanda Putra Ramadhan                      2110631170115

**PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS SINGAPERBANGSA KARAWANG**

## **DAFTAR ISI**

<b>BAB I PENDAHULUAN .....</b>	<b>3</b>
1.1    Judul Sistem .....	3
1.2    Tentang Kelompok .....	3
1.3    Pembagian Tugas .....	3
1.4    Latar Belakang .....	3
1.5    Rumusan Masalah .....	4
1.6    Tujuan Penelitian.....	4
1.7    Manfaat Penelitian.....	4
<b>BAB II PEMBAHASAN .....</b>	<b>5</b>
2.1    Tentang Sistem.....	5
2.2    Class Diagram .....	5
2.3    Database Diagram .....	6
2.4    Design GUI .....	6
2.5    Kode Program .....	8
2.6    Screenshot Program Yang Dijalankan .....	32
<b>BAB III PENUTUP.....</b>	<b>40</b>
3.1    Kesimpulan.....	40
3.2    Saran.....	40

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Judul Sistem**

Sistem Aplikasi Rental Mobil

#### **1.2 Tentang Kelompok**

Nama Kelompok : HAWANTAR

1. Muhammad Hadi Royan Firdaus (2110631170079)
2. Tarida Grace Wahyuni Margaretha Sidabutar (2110631170109)
3. Wanda Putra Ramadhan (2110631170115)

#### **1.3 Pembagian Tugas**

Nama	NPM	Tugas
Muhammad Hadi Royan Firdaus	2110631170079	<ul style="list-style-type: none"><li>● Membuat Kode Program</li><li>● Membuat Basis Data</li><li>● Membuat Laporan</li></ul>
Tarida Grace Wahyuni Margaretha Sidabutar	2110631170109	<ul style="list-style-type: none"><li>● Membuat Design GUI</li><li>● Membuat Laporan</li><li>● Membuat Laporan</li></ul>
Wanda Putra Ramadhan	2110631170115	<ul style="list-style-type: none"><li>● Membuat Kode Program</li><li>● Membuat Design GUI</li><li>● Membuat Laporan</li></ul>

#### **1.4 Latar Belakang**

Rental mobil merupakan bisnis berbentuk jasa yang sudah cukup banyak dilakukan saat ini. Bisnis ini banyak dilakukan karena tidak ada persyaratan yang rumit dalam melakukannya. Pelaku hanya perlu memiliki mobil dengan merek atau jenis apapun, sehingga pelaku sudah dapat membuka usaha penyewaan mobil miliknya sendiri. Banyaknya pelaku usaha sewa mobil juga disebabkan karena banyak orang yang membutuhkan jasa persewaan mobil untuk berbagai keperluannya, sehingga peluang usaha ini tentu saja menarik banyak orang untuk melakukannya.

Semakin banyak orang yang membuka usaha sewa mobil, semakin tinggi pula tingkat persaingannya. Oleh karena itu, para pelaku usaha ini banyak yang melakukan terobosan-terobosan untuk meningkatkan kinerja pelayanan terhadap konsumennya. Mengingat perkembangan teknologi saat ini yang pesat dan beragam, tentu saja banyak terobosan yang dapat digunakan oleh para pelaku usaha tersebut, salah satunya yaitu dengan cara membangun sistem aplikasi dengan bahasa pemrograman sesuai

dengan kebutuhan yang diperlukan pelaku usaha, sehingga dapat membantu proses pelayanan usaha ini.

Pendataan dan pencatatan dalam usaha ini merupakan sesuatu yang penting. Pencatatan yang dilakukan secara manual tentu saja kurang efektif dibandingkan dengan melakukan pendataan melalui komputer. Selain menghemat pengeluaran untuk membeli kertas, melalui komputer, membuat data laporan juga menjadi lebih efektif dan aman. Oleh karena itu, diperlukan sistem aplikasi yang dapat mencatat dan mendata informasi-informasi penyewaan yang terjadi, sehingga akan lebih mudah dalam membuat laporan.

### **1.5 Rumusan Masalah**

Berdasarkan latar belakang diatas, maka didapatkan rumusan masalah yaitu “Apa sistem aplikasi yang dapat membantu pencatatan dan pendataan untuk usaha sewa mobil”.

### **1.6 Tujuan Penelitian**

Berdasarkan latar belakang dan rumusan masalah diatas, maka didapatkan tujuan dari penelitian ini yaitu “Dapat mengetahui sistem aplikasi yang dapat membantu para pelaku usaha sewa mobil dalam melakukan pendataan dan pencatatan”.

### **1.7 Manfaat Penelitian**

Adapun manfaat yang didapatkan dari penelitian ini, terlebih khususnya yang didapatkan dari penggunaan aplikasi ini, yaitu:

- a. Memudahkan pendataan informasi pelanggan yang menyewa.
- b. Memudahkan pemilihan mobil yang dapat disewa.
- c. Membantu proses pembayaran.
- d. Memudahkan pengguna dalam membuat laporan.

## BAB II PEMBAHASAN

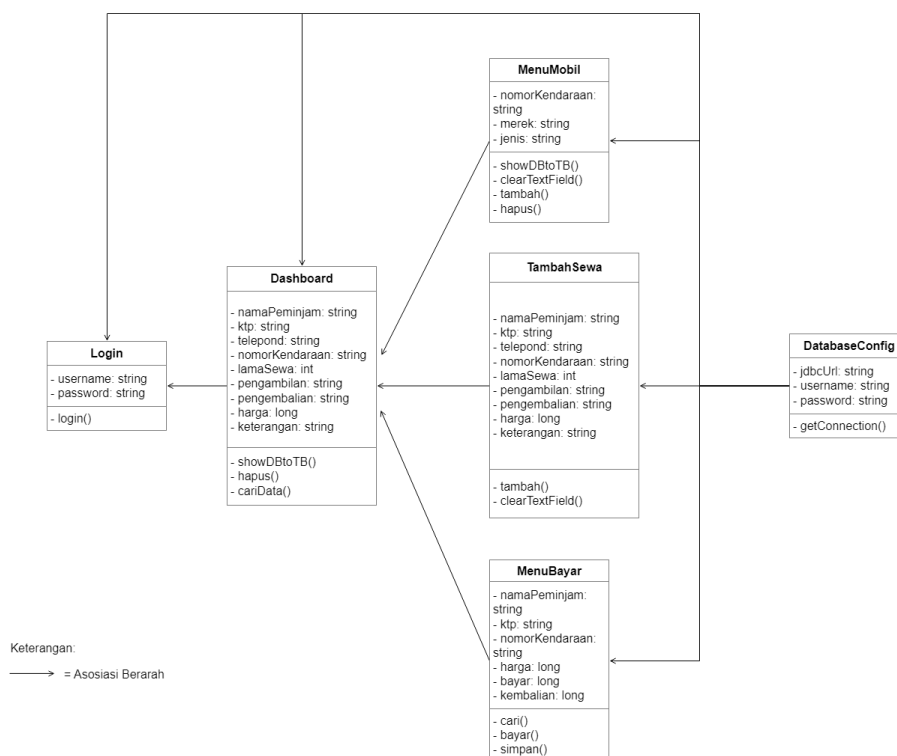
### 2.1 Tentang Sistem

Sistem Aplikasi yang kami rancang adalah aplikasi penyewaan mobil. Aplikasi ini merupakan aplikasi yang digunakan oleh pihak yang menyewakan mobil supaya lebih mudah dalam melakukan pendataan dan pencatatan terhadap penyewaan mobil yang sedang atau telah dilakukan.

Aplikasi ini dapat menyimpan informasi mulai dari identitas penyewa, mobil yang disewa, durasi sewa, harga total sewa, dan informasi status pembayarannya, apakah sudah dibayar atau belum.

Aplikasi ini juga dapat menyimpan data mobil-mobil yang dapat disewakan. Sehingga, pengguna (admin) lebih mudah memberi tahu daftar mobil yang tersedia untuk disewakan.

### 2.2 Class Diagram



## 2.3 Database Diagram



## 2.4 Design GUI

### a. Design Halaman Login

### b. Design Halaman Utama/Dashboard

Nama	No. KTP	No. Telepon	No. Kendaraan	Lama Sewa	Pengambilan	Pengembalian	Harga	Keterangan
Tarida Grace	2110631170109	0123456789	B111AAA	1	2022-12-01	2022-12-02	300000	BELUM DIBAY...
Hadi Royan	2110631170079	0123456788	B222BBB	3	2022-12-01	2022-12-03	900000	BELUM DIBAY...
Wanda Putra	2110631170115	0123456787	B333CCC	3	2022-12-01	2022-12-03	900000	BELUM DIBAY...

### c. Design Halaman Mobil

Kembali

MOBIL

Nomor Kendaraan	Merek	Jenis
B111AAA	Toyota	Agya
B222BBB	Toyota	Avanza
B333CCC	Suzuki	Ertiga

Hapus

Tambah List Mobil

Nomor Kendaraan

Merek

Jenis

Tambah

d. Design Halaman Tambah Sewa

Kembali

TAMBAH SEWA

Nama

Nomor KTP

Nomor Telepon

Nomor Kendaraan

Lama Sewa

Tanggal Pengambilan

Tanggal Pengembalian

Harga

Tambah

e. Design Halaman Bayar

Kembali **BAYAR**

Nomor KTP

Cari

Nama

-

Nomor Kendaraan

-

Harga

-

Bayar

Bayar

Kembalian

-

Simpan

## 2.5 Kode Program

### 2.5.1 Class DatabaseConfig

	DatabaseConfig
1	package packageConfig;
2	import java.sql.*;
3	
4	public class DatabaseConfig {
5	private Connection connection;
6	
7	public Connection getConnection() {
8	
9	String jdbcUrl = "jdbc:mysql://localhost:3306/aplikasi_sewa_mobil";
10	String username = "root";
11	String password = "";
12	
13	try {
14	connection = DriverManager.getConnection(jdbcUrl, username,
15	password);
16	} catch (SQLException e) {
17	e.printStackTrace();
18	}
19	return connection;
20	}
21	}

### 2.5.2 Class Login



	Login
1	package package1;
2	import java.sql.*;
3	import packageConfig.DatabaseConfig;
4	import javax.swing.JOptionPane;
5	
6	public class Login extends javax.swing.JFrame {
7	
8	public Login() {
9	initComponents();
10	setLocationRelativeTo(null);
11	}
12	
13	@SuppressWarnings("unchecked")
14	Generated Code
116	
117	//method untuk cek apakah pada database kosong atau tidak
118	private boolean isUserExist(String username) {
119	String sql = ""
120	SELECT username FROM user WHERE username= ?
121	"";
122	
123	try (Connection connection = new DatabaseConfig().getConnection();
124	PreparedStatement statement = connection.prepareStatement(sql)
125	){
126	statement.setString(1, username);
127	try (ResultSet resultSet = statement.executeQuery()) {
128	if (resultSet.next()) {
129	return true;
130	} else {
131	return false;
132	}
133	}
134	}catch (SQLException ex) {
135	throw new RuntimeException(ex);
136	}
137	}
138	
139	//method login. Digunakan untuk melakukan check database sekaligus
140	masuk ke halaman utama
141	private void login(){
142	String username = usernameTF.getText();
143	String password = new String(passwordTF.getPassword());
144	if (isUserExist(username)) {
145	String query = ""
146	SELECT username,password FROM user WHERE username=
147	? AND password=?
144	"";
145	try (Connection connection = new

```

146 DatabaseConfig().getConnection();
147     PreparedStatement statement =
148 connection.prepareStatement(query)
149     ) {
150         statement.setString(1, username);
151         statement.setString(2, password);
152         //User user = new User();
153         try (ResultSet resultSet = statement.executeQuery()) {
154
155             if (resultSet.next()) {
156                 System.out.println("username "+username+", password
157 "+password+ " ditemukan");
158                 System.out.println(resultSet.getString("username"));
159                 System.out.println(resultSet.getString("password"));
160
161                 HalUtama dashboard = new HalUtama();
162                 setVisible(false);
163                 dispose();
164                 dashboard.setVisible(true);
165             } else {
166                 JOptionPane.showMessageDialog(null, "Password Salah!");
167             }
168         }
169
170     } catch (SQLException ex) {
171         JOptionPane.showMessageDialog(null, ex.getMessage());
172     }
173
174     } else {
175         JOptionPane.showMessageDialog(null, "Username Tidak
176 Ditemukan!");
177     }
178 }
179
180 private void loginBTActionPerformed(java.awt.event.ActionEvent evt) {
181     // TODO add your handling code here:
182     login();
183 }
184
185 private void exitBTActionPerformed(java.awt.event.ActionEvent evt) {
186     // TODO add your handling code here:
187     System.exit(0);
188 }
189
190 /**
191  * @param args the command line arguments
192  */
193 public static void main(String args[]) {
194     /* Set the Nimbus look and feel */

```

```

195     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
196 code (optional) ">
197     /* If Nimbus (introduced in Java SE 6) is not available, stay with the
198 default look and feel.
199     * For details see
200 http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
201     */
202     try {
203         for (javax.swing.UIManager.LookAndFeelInfo info :
204 javax.swing.UIManager.getInstalledLookAndFeels()) {
205             if ("Nimbus".equals(info.getName())) {
206
207 javax.swing.UIManager.setLookAndFeel(info.getClassName());
208                 break;
209             }
210         }
211     } catch (ClassNotFoundException ex) {
212
213 java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.log
214 ging.Level.SEVERE, null, ex);
215     } catch (InstantiationException ex) {
216
217 java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.log
218 ging.Level.SEVERE, null, ex);
219     } catch (IllegalAccessException ex) {
220
221 java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.log
222 ging.Level.SEVERE, null, ex);
223     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
224
225 java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.log
226 ging.Level.SEVERE, null, ex);
227     }
228     //</editor-fold>
229
230     /* Create and display the form */
231     java.awt.EventQueue.invokeLater(new Runnable() {
232         public void run() {
233             new Login().setVisible(true);
234         }
235     });
236 }
237
238 // Variables declaration - do not modify
239 private javax.swing.JButton exitBT;
240 private javax.swing.JLabel jLabel1;
241 private javax.swing.JLabel jLabel2;
242 private javax.swing.JLabel jLabel3;
243 private javax.swing.JPanel jPanel1;

```

244	private javax.swing.JButton loginBT;
245	private javax.swing.JPasswordField passwordTF;
246	private javax.swing.JTextField usernameTF;
247	// End of variables declaration
248	}

### 2.5.3 Class HalUtama

	HalUtama
1	package package1;
2	
3	import packageConfig.DatabaseConfig;
4	import java.sql.*;
5	import javax.swing.JOptionPane;
6	import javax.swing.table.DefaultTableModel;
7	
8	public class HalUtama extends javax.swing.JFrame {
9	
10	DefaultTableModel dtmSewa;
11	Connection connection;
12	Statement statement;
13	ResultSet resultSet;
14	
15	String namaPeminjam;
16	String noKTP;
17	String noKendaraan;
18	
19	
20	public HalUtama() {
21	initComponents();
22	setLocationRelativeTo(null);
23	showDBtoTB();
24	}
25	
26	@SuppressWarnings("unchecked")
27	Generated Code
204	
205	//method untuk menampilkan isi database ke tabel
206	private void showDBtoTB(){
207	//mengkoneksikan ke database tabel mobil
208	try {
209	connection = new DatabaseConfig().getConnection();
210	statement = connection.createStatement();
211	resultSet = statement.executeQuery("SELECT * FROM sewa");
212	} catch (SQLException ex) {
213	
214	}

215	
216	//membuat judul kolom tabel
217	dtmSewa =new DefaultTableModel();
218	dtmSewa.addColumn("Nama");
219	dtmSewa.addColumn("No. KTP");
220	dtmSewa.addColumn("No. Telepon");
221	dtmSewa.addColumn("No. Kendaraan");
222	dtmSewa.addColumn("Lama Sewa");
223	dtmSewa.addColumn("Pengambilan");
224	dtmSewa.addColumn("Pengembalian");
225	dtmSewa.addColumn("Harga");
226	dtmSewa.addColumn("Keterangan");
227	listSewaTBL.setModel(dtmSewa);
228	
229	//mengambil data dari database
230	try {
231	while(resultSet.next()){
232	dtmSewa.addRow(new Object[]{
233	resultSet.getString("nama_peminjam"),
234	resultSet.getString("no_ktp"),
235	resultSet.getString("no_telepon"),
236	resultSet.getString("no_kendaraan"),
237	resultSet.getString("lama_sewa"),
238	resultSet.getString("pengambilan"),
239	resultSet.getString("pengembalian"),
240	resultSet.getString("harga"),
241	resultSet.getString("keterangan")
242	});
243	}
244	} catch(SQLException ex) {
245	ex.printStackTrace();
246	}
247	
248	}
249	
250	private void hapusRecord(){
251	int column = 1;
252	int row = listSewaTBL.getSelectedRow();
253	String nomorKTP = listSewaTBL.getModel().getValueAt(row,
254	column).toString();
255	System.out.println(nomorKTP);
256	String sql = ""
257	DELETE FROM sewa WHERE no_ktp=?
258	"";
259	try (
260	Connection connection = new
261	DatabaseConfig().getConnection();
262	PreparedStatement statement = connection.prepareStatement(sql)
263	) {

```

262         statement.setString(1, nomorKTP);
263         statement.executeUpdate();
264     } catch (SQLException e) {
265     }
266 }
267 }
268
269 private boolean isDataExist(String noKTP) {
270     String sql = ""
271         SELECT no_ktp FROM sewa WHERE no_ktp= ?
272         "";
273
274     try (Connection connection = new DatabaseConfig().getConnection();
275         PreparedStatement statement = connection.prepareStatement(sql)
276     ){
277         statement.setString(1, noKTP);
278         try (ResultSet resultSet = statement.executeQuery()) {
279             return resultSet.next();
280         }
281     } catch (SQLException ex) {
282         throw new RuntimeException(ex);
283     }
284 }
285
286 public void cariData() {
287
288     String noKTP = inputCariKTPITF.getText();
289     if (isDataExist(noKTP)) {
290         String sql = ""
291             SELECT * FROM sewa WHERE no_ktp=?
292             "";
293
294         try (Connection connection = new
295 DatabaseConfig().getConnection();
296             PreparedStatement statement =
297 connection.prepareStatement(sql)
298         ) {
299             statement.setString(1, noKTP);
300             try (ResultSet resultSet = statement.executeQuery()) {
301                 //membuat judul kolom tabel
302                 dtmSewa =new DefaultTableModel();
303                 dtmSewa.addColumn("Nama");
304                 dtmSewa.addColumn("No. KTP");
305                 dtmSewa.addColumn("No. Telepon");
306                 dtmSewa.addColumn("No. Kendaraan");
307                 dtmSewa.addColumn("Lama Sewa");
308                 dtmSewa.addColumn("Pengambilan");
309                 dtmSewa.addColumn("Pengembalian");
310                 dtmSewa.addColumn("Harga");

```

308	dtmSewa.addColumn("Keterangan");
309	listSewaTBL.setModel(dtmSewa);
310	if (resultSet.next()) {
311	//mengambil data dari database
312	try {
313	dtmSewa.addRow(new Object[]{
314	resultSet.getString("nama_peminjam"),
315	resultSet.getString("no_ktp"),
316	resultSet.getString("no_telepon"),
317	resultSet.getString("no_kendaraan"),
318	resultSet.getString("lama_sewa"),
319	resultSet.getString("pengambilan"),
320	resultSet.getString("pengembalian"),
321	resultSet.getString("harga"),
322	resultSet.getString("keterangan")
323	});
324	} catch(SQLException ex) {
325	JOptionPane.showMessageDialog(null, "Error: "+
326	ex.getMessage(), "ERROR", ERROR);
327	ex.printStackTrace();
328	} else {
329	JOptionPane.showMessageDialog(null, "Data Tidak
330	Ditemukan");
331	}
332	
333	} catch (SQLException ex) {
334	JOptionPane.showMessageDialog(null, ex.getMessage());
335	throw new RuntimeException(ex);
336	}
337	} else {
338	JOptionPane.showMessageDialog(null, "Data tidak ditemukan");
339	}
340	}
341	
342	private void hapusBTActionPerformed(java.awt.event.ActionEvent evt) {
343	// TODO add your handling code here:
344	hapusRecord();
345	showDBtoTB();
346	}
347	
348	private void
349	tambahSewaBTSewaBTActionPerformed(java.awt.event.ActionEvent evt)
350	{
351	// TODO add your handling code here:
352	HalTambahSewa halTambahSewa = new HalTambahSewa();
	dispose();
	setVisible(false);

353	halTambahSewa.setVisible(true);
354	}
355	
356	private void mobilBTActionPerformed(java.awt.event.ActionEvent evt) {
357	// TODO add your handling code here:
358	HalMobil halMobil = new HalMobil();
359	dispose();
360	setVisible(false);
361	halMobil.setVisible(true);
362	}
363	
364	private void bayarBTActionPerformed(java.awt.event.ActionEvent evt) {
365	// TODO add your handling code here:
366	HalBayar halBayar = new HalBayar();
367	dispose();
368	setVisible(false);
369	halBayar.setVisible(true);
370	}
371	
372	private void cariBTNActionPerformed(java.awt.event.ActionEvent evt) {
373	// TODO add your handling code here:
374	cariData();
375	}
376	
377	private void
	tampilkanSeluruhDataBTNActionPerformed(java.awt.event.ActionEvent
	evt) {
378	// TODO add your handling code here:
379	showDBtoTB();
380	inputCariKTPITF.setText("");
381	}
382	
383	public static void main(String args[]) {
384	/* Set the Nimbus look and feel */
385	//<editor-fold defaultstate="collapsed" desc=" Look and feel setting
	code (optional) ">
386	/* If Nimbus (introduced in Java SE 6) is not available, stay with the
	default look and feel.
387	* For details see
	<a href="http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html">http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html</a>
388	*/
389	try {
390	for (javax.swing.UIManager.LookAndFeelInfo info :
	javax.swing.UIManager.getInstalledLookAndFeels()) {
391	if ("Nimbus".equals(info.getName())) {
392	
393	javax.swing.UIManager.setLookAndFeel(info.getClassName());
394	break;
395	}



396	}
397	} catch (ClassNotFoundException ex) {
398	java.util.logging.Logger.getLogger(HalUtama.class.getName()).log(java.ut
399	il.logging.Level.SEVERE, null, ex);
400	} catch (InstantiationException ex) {
401	java.util.logging.Logger.getLogger(HalUtama.class.getName()).log(java.ut
402	il.logging.Level.SEVERE, null, ex);
403	} catch (IllegalAccessException ex) {
404	java.util.logging.Logger.getLogger(HalUtama.class.getName()).log(java.ut
405	il.logging.Level.SEVERE, null, ex);
406	}
407	//</editor-fold>
408	/* Create and display the form */
409	java.awt.EventQueue.invokeLater(new Runnable() {
410	public void run() {
411	new HalUtama().setVisible(true);
412	}
413	});
414	}
415	
416	// Variables declaration - do not modify
417	private javax.swing.JButton bayarBT;
418	private javax.swing.JButton cariBTN;
419	private javax.swing.JButton hapusBT;
420	private javax.swing.JTextField inputCariKTPITF;
421	private javax.swing.JLabel jLabel1;
422	private javax.swing.JLabel jLabel3;
423	private javax.swing.JPanel jPanel1;
424	private javax.swing.JPanel jPanel2;
425	private javax.swing.JScrollPane jScrollPane1;
426	private javax.swing.JTable listSewaTBL;
427	private javax.swing.JButton mobilBT;
428	private javax.swing.JButton tambahSewaBT;
429	private javax.swing.JButton tampilkanSeluruhDataBTN;
430	// End of variables declaration
	}

#### 2.5.4 Class HalMobil

	HalMobil
1	package package1;
2	
3	import static java.awt.image.ImageObserver.ERROR;

4	import packageConfig.DatabaseConfig;
5	import java.sql.*;
6	import javax.swing.JOptionPane;
7	import javax.swing.table.DefaultTableModel;
8	
9	public class HalMobil extends javax.swing.JFrame {
10	
11	DefaultTableModel modelMobil;
12	Connection con;
13	Statement stm;
14	ResultSet res;
15	
16	public HalMobil() {
17	initComponents();
18	setLocationRelativeTo(null);
19	showDBtoTB();
20	}
21	
22	@SuppressWarnings("unchecked")
23	Generated Code
263	
264	private void showDBtoTB(){
265	//mengkoneksikan ke database tabel mobil
266	try {
267	con = new DatabaseConfig().getConnection();
268	stm= con.createStatement();
269	res = stm.executeQuery("SELECT * FROM mobil");
270	} catch (SQLException ex) {
271	
272	}
273	
274	//membuat judul kolom tabel
275	modelMobil =new DefaultTableModel();
276	modelMobil.addColumn("Nomor Kendaraan");
277	modelMobil.addColumn("Merek");
278	modelMobil.addColumn("Jenis");
279	listMobilTBL.setModel(modelMobil);
280	
281	//mengambil data dari database
282	try {
283	while(res.next()){
284	modelMobil.addRow(new Object[]{
285	res.getString("nomorKendaraan"),
286	res.getString("merek"),
287	res.getString("jenis")
288	});
289	}
290	} catch(SQLException ex) {
291	

```

292     }
293
294 }
295
296 private void hapusRecord(){
297     int column = 0;
298     int row = listMobilTBL.getSelectedRow();
299     String nomorKendaraan = listMobilTBL.getModel().getValueAt(row,
column).toString();
300
301     String sql = ""
302         DELETE FROM mobil WHERE nomorKendaraan=?
303         "";
304     try (
305         Connection connection = new
DatabaseConfig().getConnection();
306         PreparedStatement statement = connection.prepareStatement(sql)
307     ) {
308         statement.setString(1, nomorKendaraan);
309         statement.executeUpdate();
310     } catch (SQLException e) {
311
312     }
313 }
314
315 private void tambahRecord(){
316     String nomorKendaraan = nomorKendaraanTF.getText();
317     String merek = merekTF.getText();
318     String jenis = jenisTF.getText();
319
320     //memasukkan data dari textfield ke database
321     String sql = ""
322         INSERT INTO mobil(nomorKendaraan, merek, jenis)
VALUES(?, ?, ?)
323         "";
324     try (
325         Connection connection = new
DatabaseConfig().getConnection();
326         PreparedStatement statement = connection.prepareStatement(sql)
327     ) {
328         statement.setString(1, nomorKendaraan);
329         statement.setString(2, merek);
330         statement.setString(3, jenis);
331         statement.executeUpdate();
332     } catch (SQLException e) {
333         throw new RuntimeException(e.getMessage()); // Error
334     }
335 }
336

```

```

337 private void clearTextField(){
338     nomorKendaraanTF.setText("");
339     merekTF.setText("");
340     jenisTF.setText("");
341 }
342 private boolean isDataMobilExist(String noKTP) {
343     String sql = ""
344     SELECT nomorKendaraan FROM mobil WHERE
nomorKendaraan= ?
345     "";
346
347     try (Connection connection = new DatabaseConfig().getConnection();
348         PreparedStatement statement = connection.prepareStatement(sql)
349     ){
350         statement.setString(1, noKTP);
351         try (ResultSet resultSet = statement.executeQuery()) {
352             return resultSet.next();
353         }
354     }catch (SQLException ex) {
355         throw new RuntimeException(ex);
356     }
357 }
358
359 public void cariDataMobil() {
360
361     String noKendaraan = cariNopolTF.getText();
362     if (isDataMobilExist(noKendaraan)) {
363         String sql = ""
364         SELECT * FROM mobil WHERE nomorKendaraan=?
365         "";
366
367         try (Connection connection = new
DatabaseConfig().getConnection();
368             PreparedStatement statement =
connection.prepareStatement(sql)
369         ) {
370             statement.setString(1, noKendaraan);
371             try (ResultSet resultSet = statement.executeQuery()) {
372                 //membuat judul kolom tabel
373                 modelMobil =new DefaultTableModel();
374                 modelMobil.addColumn("Nomor Kendaraan");
375                 modelMobil.addColumn("Merek");
376                 modelMobil.addColumn("Jenis");
377                 listMobilTBL.setModel(modelMobil);
378                 if (resultSet.next()) {
379                     //mengambil data dari database
380                     try {
381                         modelMobil.addRow(new Object[]{
382                             resultSet.getString("nomorKendaraan"),
383                             resultSet.getString("merek"),

```

383	resultSet.getString("jenis"),
384	});
385	} catch(SQLException ex) {
386	JOptionPane.showMessageDialog(null, "Error: "+
	ex.getMessage(), "ERROR", ERROR);
387	ex.printStackTrace();
388	}
389	} else {
390	JOptionPane.showMessageDialog(null, "Data Tidak
	Ditemukan");
391	}
392	}
393	
394	} catch (SQLException ex) {
395	JOptionPane.showMessageDialog(null, ex.getMessage());
396	throw new RuntimeException(ex);
397	}
398	} else {
399	JOptionPane.showMessageDialog(null, "Data tidak ditemukan");
400	}
401	}
402	
403	private void hapusBTActionPerformed(java.awt.event.ActionEvent evt) {
404	// TODO add your handling code here:
405	hapusRecord();
406	showDBtoTB();
407	}
408	
409	private void tambahBTActionPerformed(java.awt.event.ActionEvent
	evt) {
410	// TODO add your handling code here:
411	tambahRecord();
412	showDBtoTB();
413	clearTextField();
414	}
415	
416	private void kembaliLBMouseClicked(java.awt.event.MouseEvent evt) {
417	// TODO add your handling code here:
418	HalUtama halUtama = new HalUtama();
419	dispose();
420	setVisible(false);
421	halUtama.setVisible(true);
422	}
423	
	private void cariNopolTFActionPerformed(java.awt.event.ActionEvent
424	evt) {
425	// TODO add your handling code here:
426	}
427	

```

428 private void cariBTNActionPerformed(java.awt.event.ActionEvent evt) {
429     // TODO add your handling code here:
430     cariDataMobil();
431     cariNopolTF.setText("");
432 }
433
434 private void listBTNActionPerformed(java.awt.event.ActionEvent evt) {
435     // TODO add your handling code here:
436     showDBtoTB();
437 }
438
439 public static void main(String args[]) {
440     /* Set the Nimbus look and feel */
441     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
442     /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
443     * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
444     */
445     try {
446         for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
447             if ("Nimbus".equals(info.getName())) {
448
449 javax.swing.UIManager.setLookAndFeel(info.getClassName());
450                 break;
451             }
452         }
453     } catch (ClassNotFoundException ex) {
454         java.util.logging.Logger.getLogger(HalMobil.class.getName()).log(java.util.
l.logging.Level.SEVERE, null, ex);
455     } catch (InstantiationException ex) {
456         java.util.logging.Logger.getLogger(HalMobil.class.getName()).log(java.util.
l.logging.Level.SEVERE, null, ex);
457     } catch (IllegalAccessException ex) {
458         java.util.logging.Logger.getLogger(HalMobil.class.getName()).log(java.util.
l.logging.Level.SEVERE, null, ex);
459     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
460         java.util.logging.Logger.getLogger(HalMobil.class.getName()).log(java.util.
l.logging.Level.SEVERE, null, ex);
461     }
462     //</editor-fold>
463
464     /* Create and display the form */
465     java.awt.EventQueue.invokeLater(new Runnable() {
466         public void run() {
467             new HalMobil().setVisible(true);
468         }

```

469	});
470	}
471	
472	// Variables declaration - do not modify
473	private javax.swing.JButton cariBTN;
474	private javax.swing.JTextField cariNopolTF;
475	private javax.swing.JButton hapusBT;
476	private javax.swing.JLabel jLabel1;
477	private javax.swing.JLabel jLabel2;
478	private javax.swing.JLabel jLabel3;
479	private javax.swing.JLabel jLabel4;
480	private javax.swing.JLabel jLabel5;
481	private javax.swing.JLabel jLabel6;
482	private javax.swing.JPanel jPanel1;
483	private javax.swing.JPanel jPanel2;
484	private javax.swing.JPanel jPanel3;
485	private javax.swing.JScrollPane jScrollPane1;
486	private javax.swing.JTextField jenisTF;
487	private javax.swing.JLabel kembaliLB;
488	private javax.swing.JButton listBTN;
489	private javax.swing.JTable listMobilTBL;
490	private javax.swing.JTextField merekTF;
491	private javax.swing.JTextField nomorKendaraanTF;
492	private javax.swing.JButton tambahBT;
493	// End of variables declaration
494	}

#### 2.5.5 Class HalTambahSewa

	HalTambahSewa
1	package package1;
2	
3	import packageConfig.DatabaseConfig;
4	import java.sql.*;
5	import java.text.SimpleDateFormat;
6	import javax.swing.JOptionPane;
7	
8	public class HalTambahSewa extends javax.swing.JFrame {
9	
10	public HalTambahSewa() {
11	initComponents();
12	setLocationRelativeTo(null);
13	ItemNomorKendaraan();
14	}
15	
16	@ SuppressWarnings("unchecked")

17	Generated Code
251	
252	public void ItemNomorKendaraan() {
253	String sql = ""
254	SELECT * FROM mobil
255	"";
256	try (
257	Connection connection = new
258	DatabaseConfig().getConnection();
259	PreparedStatement statement = connection.prepareStatement(sql)
260	) {
261	try (ResultSet res = statement.executeQuery()) {
262	while (res.next()) {
263	noKendaraanCB.addItem(res.getString("nomorKendaraan"));
264	}
265	}
266	} catch (SQLException e) {
267	throw new RuntimeException(e.getMessage()); // Error
268	}
269	}
270	
271	private void tambahRecord(){
272	String nama = namaTF.getText();
273	String ktp = ktpTF.getText();
274	String telepon = teleponTF.getText();
275	String noKendaraan = noKendaraanCB.getSelectedItem().toString();
276	int hargaPerhari = Integer.parseInt(hargaTF.getText());
277	int lamaSewa = Integer.parseInt(lamaSewaTF.getText());
278	String tglPengambilan = new SimpleDateFormat("yyyy-MM-
279	dd").format(pengambilanDC.getDate());
280	String tglPengembalian = new SimpleDateFormat("yyyy-MM-
281	dd").format(pengembalianDC.getDate());
282	String keterangan = "BELUM DIBAYAR";
283	
284	int totalHarga = hargaPerhari * lamaSewa;
285	
286	//memasukkan data dari textfield ke database
287	String sql = ""
288	INSERT INTO sewa(nama_Peminjam, no_ktp, no_telepon,
289	no_kendaraan, lama_sewa,
290	pengambilan, pengembalian, harga, keterangan) VALUES(?, ?,
291	?, ?, ?, ?, ?, ?)
292	"";
	try (
	Connection connection = new
	DatabaseConfig().getConnection();
	PreparedStatement statement = connection.prepareStatement(sql)
	) {



293	statement.setString(1, nama);
294	statement.setString(2, ktp);
295	statement.setString(3, telepon);
296	statement.setString(4, noKendaraan);
297	statement.setInt(5, lamaSewa);
298	statement.setString(6, tglPengambilan);
299	statement.setString(7, tglPengembalian);
300	statement.setInt(8, totalHarga);
301	statement.setString(9, keterangan);
302	statement.executeUpdate();
303	JOptionPane.showMessageDialog(null, "Data berhasil
304	ditambahkan");
	} catch (SQLException e) {
305	throw new RuntimeException(e.getMessage()); // Error
306	}
307	}
308	
309	private void clearTextField(){
310	namaTF.setText("");
311	ktpTF.setText("");
312	teleponTF.setText("");
313	lamaSewaTF.setText("");
314	hargaTF.setText("");
315	pengambilanDC.setCalendar(null);
316	pengembalianDC.setCalendar(null);
317	}
318	
319	
320	
321	private void hargaTFActionPerformed(java.awt.event.ActionEvent evt)
322	{
323	// TODO add your handling code here:
324	}
325	
	private void lamaSewaTFActionPerformed(java.awt.event.ActionEvent
326	evt) {
327	// TODO add your handling code here:
328	}
329	
	private void
330	noKendaraanCBActionPerformed(java.awt.event.ActionEvent evt) {
331	// TODO add your handling code here:
332	}
333	
	private void teleponTFActionPerformed(java.awt.event.ActionEvent
334	evt) {
335	// TODO add your handling code here:
336	}
337	

```

338 private void ktpTFActionPerformed(java.awt.event.ActionEvent evt) {
339     // TODO add your handling code here:
340 }
341
342 private void namaTFActionPerformed(java.awt.event.ActionEvent evt) {
343     // TODO add your handling code here:
344 }
345
346 private void kembaliLBMouseClicked(java.awt.event.MouseEvent evt) {
347     // TODO add your handling code here:
348     HalUtama halUtama = new HalUtama();
349     dispose();
350     setVisible(false);
351     halUtama.setVisible(true);
352 }
353
354 private void tambahBTActionPerformed(java.awt.event.ActionEvent
355 evt) {
356     // TODO add your handling code here:
357     tambahRecord();
358     clearTextField();
359 }
360
361 public static void main(String args[]) {
362     /* Set the Nimbus look and feel */
363     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
364     code (optional) ">
365     /* If Nimbus (introduced in Java SE 6) is not available, stay with the
366     default look and feel.
367     * For details see
368     http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
369     */
370     try {
371         for (javax.swing.UIManager.LookAndFeelInfo info :
372             javax.swing.UIManager.getInstalledLookAndFeels()) {
373             if ("Nimbus".equals(info.getName())) {
374                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
375                 break;
376             }
377         }
378     } catch (ClassNotFoundException ex) {
379         java.util.logging.Logger.getLogger(HalTambahSewa.class.getName()).log(
380             java.util.logging.Level.SEVERE, null, ex);
381     } catch (InstantiationException ex) {
382         java.util.logging.Logger.getLogger(HalTambahSewa.class.getName()).log(
383             java.util.logging.Level.SEVERE, null, ex);
384     } catch (IllegalAccessException ex) {
385         java.util.logging.Logger.getLogger(HalTambahSewa.class.getName()).log(

```

379	java.util.logging.Level.SEVERE, null, ex);
	} catch (javax.swing.UnsupportedLookAndFeelException ex) {
380	java.util.logging.Logger.getLogger(HalTambahSewa.class.getName()).log(
381	java.util.logging.Level.SEVERE, null, ex);
382	}
383	//</editor-fold>
384	
385	/* Create and display the form */
386	java.awt.EventQueue.invokeLater(new Runnable() {
387	public void run() {
388	new HalTambahSewa().setVisible(true);
389	}
390	});
391	}
392	
393	// Variables declaration - do not modify
394	private javax.swing.JTextField hargaTF;
395	private javax.swing.JLabel jLabel1;
396	private javax.swing.JLabel jLabel10;
397	private javax.swing.JLabel jLabel3;
398	private javax.swing.JLabel jLabel4;
399	private javax.swing.JLabel jLabel5;
400	private javax.swing.JLabel jLabel6;
401	private javax.swing.JLabel jLabel7;
402	private javax.swing.JLabel jLabel8;
403	private javax.swing.JLabel jLabel9;
404	private javax.swing.JPanel jPanel1;
405	private javax.swing.JPanel jPanel2;
406	private javax.swing.JLabel kembaliLB;
407	private javax.swing.JTextField ktpTF;
408	private javax.swing.JTextField lamaSewaTF;
409	private javax.swing.JTextField namaTF;
410	private javax.swing.JComboBox<String> noKendaraanCB;
411	private com.toedter.calendar.JDateChooser pengambilanDC;
412	private com.toedter.calendar.JDateChooser pengembalianDC;
413	private javax.swing.JButton tambahBT;
414	private javax.swing.JTextField teleponTF;
415	// End of variables declaration
	}

#### 2.5.6 Class HalBayar

	HalBayar
1	package package1;
2	
3	import java.sql.Connection;
4	import java.sql.PreparedStatement;

5	import java.sql.ResultSet;
6	import java.sql.SQLException;
7	import javax.swing.JOptionPane;
8	
9	import packageConfig.DatabaseConfig;
10	
11	public class HalBayar extends javax.swing.JFrame {
12	
13	String ktp = "";
14	String keteranganSewa = "";
15	long hargaSewa = 0L;
16	
17	public HalBayar() {
18	initComponents();
19	setLocationRelativeTo(null);
20	}
21	
22	private boolean isDataExist(String noKTP) {
23	String sql = ""
24	SELECT no_ktp FROM sewa WHERE no_ktp= ?
25	"";
26	
27	try (Connection connection = new DatabaseConfig().getConnection();
28	PreparedStatement statement = connection.prepareStatement(sql)
29	){
30	statement.setString(1, noKTP);
31	try (ResultSet resultSet = statement.executeQuery()) {
32	return resultSet.next();
33	}
34	}catch (SQLException ex) {
35	throw new RuntimeException(ex);
36	}
37	}
38	
39	public void cariData() {
40	String noKTP = inputKTPPTF.getText();
41	if (isDataExist(noKTP)) {
42	String sql = ""
43	SELECT * FROM sewa WHERE no_ktp=?
44	"";
45	try (Connection connection = new
46	DatabaseConfig().getConnection();
47	PreparedStatement statement =
48	connection.prepareStatement(sql)
49	) {
50	statement.setString(1, noKTP);
51	try (ResultSet resultSet = statement.executeQuery()) {

52	if (resultSet.next()) {
53	keteranganSewa = resultSet.getString("keterangan");
54	if (!keteranganSewa.equals("SUDAH DIBAYAR")) {
55	hasilNamaLabel.setText(resultSet.getString("nama_peminjam"));
56	hasilNopolLabel.setText(resultSet.getString("no_kendaraan"));
57	hasilHargaLabel.setText("Rp."+resultSet.getLong("harga"));
58	
59	ktp = resultSet.getString("no_ktp");
60	hargaSewa = resultSet.getLong("harga");
61	} else {
62	JOptionPane.showMessageDialog(null, "DATA SEWA
	SUDAH LUNAS");
63	}
64	} else {
65	JOptionPane.showMessageDialog(null, "Data Tidak
	Ditemukan");
66	}
67	}
68	
69	} catch (SQLException ex) {
70	JOptionPane.showMessageDialog(null, ex.getMessage());
71	throw new RuntimeException(ex);
72	}
73	} else {
74	JOptionPane.showMessageDialog(null, "Data tidak ditemukan");
75	}
76	}
77	
78	
79	public void hitungBayar() {
80	long inputBayar = Long.parseLong(inputBayarTF.getText());
81	long kembalian;
82	if (inputBayar >= hargaSewa && hargaSewa != 0) {
83	kembalian = inputBayar - hargaSewa;
84	kembalianLabel.setText("Rp."+kembalian);
85	keteranganSewa = "SUDAH DIBAYAR";
86	} else {
87	JOptionPane.showMessageDialog(null, "Uang tidak cukup");
88	}
89	}
90	
91	public void updateKeterangan(String keterangan, String ktp) {
92	String sql = ""
93	UPDATE sewa SET keterangan=? WHERE no_ktp=?
94	"";
95	try (
96	Connection connection = new
	DatabaseConfig().getConnection();
97	PreparedStatement statement = connection.prepareStatement(sql)

98	) {
99	statement.setString(1, keterangan);
100	statement.setString(2, ktp);
101	statement.executeUpdate();
102	JOptionPane.showMessageDialog(null, "Berhasil disimpan");
103	
104	} catch (SQLException e) {
105	throw new RuntimeException(e.getMessage()); // Error
106	}
107	}
108	
109	public void clearTextField() {
110	hasilNamaLabel.setText("");
111	hasilHargaLabel.setText("");
112	hasilNopolLabel.setText("");
113	kembalianLabel.setText("");
114	inputBayarTF.setText("");
115	}
116	
117	@SuppressWarnings("unchecked")
118	Generated Code
322	
323	private void inputKTPTFActionPerformed(java.awt.event.ActionEvent
324	evt) {
325	// TODO add your handling code here:
326	}
327	private void cariBTNActionPerformed(java.awt.event.ActionEvent evt)
328	{
329	// TODO add your handling code here:
330	cariData();
331	}
332	private void inputBayarTFActionPerformed(java.awt.event.ActionEvent
333	evt) {
334	// TODO add your handling code here:
335	}
336	private void bayarBTNActionPerformed(java.awt.event.ActionEvent
337	evt) {
338	// TODO add your handling code here:
339	hitungBayar();
340	}
341	private void simpanBTNActionPerformed(java.awt.event.ActionEvent
342	evt) {
343	// TODO add your handling code here:
344	updateKeterangan(keteranganSewa, ktp);
	clearTextField();

```

345     }
346
347     private void kembaliLBMouseClicked(java.awt.event.MouseEvent evt) {
348         // TODO add your handling code here:
349         HalUtama halUtama = new HalUtama();
350         dispose();
351         setVisible(false);
352         halUtama.setVisible(true);
353     }
354
355     public static void main(String args[]) {
356         /* Set the Nimbus look and feel */
357         //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
358         /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
359         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
*/
360         try {
361             for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
362                 if ("Nimbus".equals(info.getName())) {
363                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
364                     break;
365                 }
366             }
367         } catch (ClassNotFoundException ex) {
368             java.util.logging.Logger.getLogger(HalBayar.class.getName()).log(java.uti
l.logging.Level.SEVERE, null, ex);
369         } catch (InstantiationException ex) {
370             java.util.logging.Logger.getLogger(HalBayar.class.getName()).log(java.uti
l.logging.Level.SEVERE, null, ex);
371         } catch (IllegalAccessException ex) {
372             java.util.logging.Logger.getLogger(HalBayar.class.getName()).log(java.uti
l.logging.Level.SEVERE, null, ex);
373         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
374             java.util.logging.Logger.getLogger(HalBayar.class.getName()).log(java.uti
l.logging.Level.SEVERE, null, ex);
375         }
376     }
377     //</editor-fold>
378
379     /* Create and display the form */
380     java.awt.EventQueue.invokeLater(new Runnable() {
381         public void run() {
382             new HalBayar().setVisible(true);
383         }
384     });
385 }

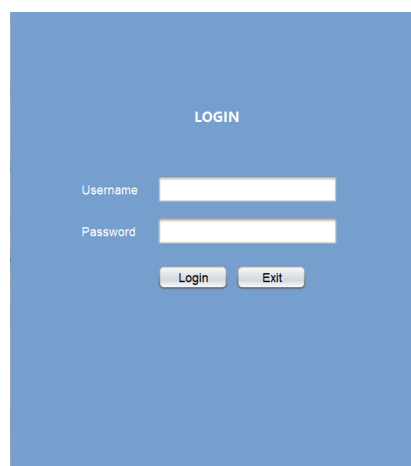
```

386	
387	// Variables declaration - do not modify
388	private javax.swing.JButton bayarBTN;
389	private javax.swing.JButton cariBTN;
390	private javax.swing.JLabel hasilHargaLabel;
391	private javax.swing.JLabel hasilNamaLabel;
392	private javax.swing.JLabel hasilNopolLabel;
393	private javax.swing.JTextField inputBayarTF;
394	private javax.swing.JTextField inputKTPTF;
395	private javax.swing.JLabel jLabel1;
396	private javax.swing.JLabel jLabel3;
397	private javax.swing.JLabel jLabel4;
398	private javax.swing.JLabel jLabel5;
399	private javax.swing.JLabel jLabel6;
400	private javax.swing.JLabel jLabel7;
401	private javax.swing.JLabel jLabel8;
402	private javax.swing.JPanel jPanel1;
403	private javax.swing.JPanel jPanel2;
404	private javax.swing.JLabel kembaliLB;
405	private javax.swing.JLabel kembalianLabel;
406	private javax.swing.JButton simpanBTN;
407	// End of variables declaration
408	}
409	

## 2.6 Screenshot Program Yang Dijalankan

### 2.6.1 Halaman Login

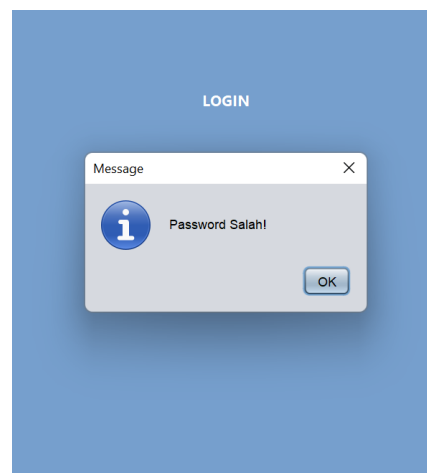
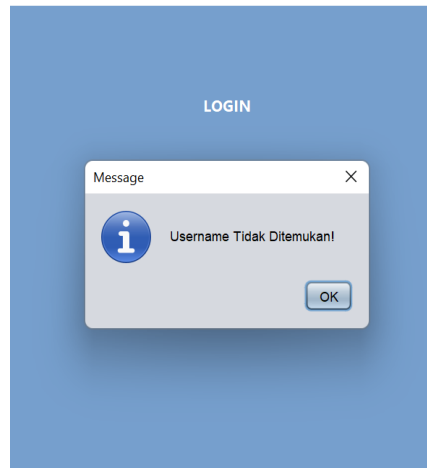
Halaman Login saat dijalankan akan seperti gambar dibawah. Terdapat Textfield yang harus diisi. Yaitu username dan password.



Apabila username yang dimasukkan berbeda dengan apa yang ada di database, maka akan tampil pesan “username tidak ditemukan”. Apabila username benar, tetapi password salah, maka akan tampil pesan “password



salah”. Apabila username dan password sudah sesuai, maka akan ditampilkan halaman utama.



## 2.6.2 Halaman Utama

Pada halaman utama digunakan untuk menampilkan data atau informasi tentang penyewa yang disajikan dalam bentuk tabel.

DASHBOARD								
Nama	No. KTP	No. Telepon	No. Kendaraan	Lama Sewa	Pengambilan	Pengembalian	Harga	Keterangan
Tarida Grace	2110631170109	0123456789	B111AAA	1	2022-12-01	2022-12-02	300000	BELUM DIBAY...
Hadli Royan	2110631170079	0123456788	B222BBB	3	2022-12-01	2022-12-03	900000	BELUM DIBAY...
Wanda Putra	2110631170115	0123456787	B333CCC	3	2022-12-01	2022-12-03	900000	BELUM DIBAY...

Hapus

List

Mobil

Tambah Sewa

Bayar

No. KTP

Cari

Tombol Hapus digunakan untuk menghapus salah satu record yang dipilih. Sedangkan tombol cari digunakan untuk mencari record berdasarkan Nomor KTP yang diisi pada textfield yang disediakan. Contohnya seperti gambar dibawah ini.

DASHBOARD

Nama	No. KTP	No. Telepon	No. Kendaraan	Lama Sewa	Pengambilan	Pengembalian	Harga	Keterangan
Tarida Grace	2110631170109	0123456789	B111AAA	1	2022-12-01	2022-12-02	300000	BELUM DIBAY...

Hapus List

Mobil Tambah Sewa Bayar

No. KTP 2110631170109 Cari

Untuk menampilkan seluruh list seperti semula, hanya perlu klik tombol List. Namun, jika nomor KTP yang dimasukkan tidak ditemukan, maka akan tampil pesan “Data tidak ditemukan”.

DASHBOARD

Nama	No. KTP	No. Telepon	No. Kendaraan	Lama Sewa	Pengambilan	Pengembalian	Harga	Keterangan
Tarida Grace	2110631170109	0123456789	B111AAA	1	2022-12-01	2022-12-02	300000	BELUM DIBAY...
Hadi Royan	2110631170079	0123456788	B222BBB	3	2022-12-01	2022-12-03	900000	BELUM DIBAY...
Wanda Putra	2110631170115	0123456787	B333CCC	3	2022-12-01	2022-12-03	900000	BELUM DIBAY...

Message

Data tidak ditemukan

OK

Hapus List

Mobil Tambah Sewa Bayar

No. KTP 112233 Cari

Untuk tombol lainnya, yaitu tombol Mobil, Tambah Sewa, dan Bayar digunakan untuk menampilkan halamannya masing-masing.

### 2.6.3 Halaman Mobil

Pada Halaman Mobil berisi informasi daftar mobil yang tersedia untuk disewakan.

Kembali **MOBIL**

Nomor Kendaraan	Merek	Jenis
B111AAA	Toyota	Agia
B222BBB	Toyota	Avanza
B333CCC	Suzuki	Ertiga

No. Kendaraan

Tambah List Mobil

Nomor Kendaraan   
Merek   
Jenis

Dalam halaman utama juga terdapat fitur untuk mencari mobil berdasarkan nomor kendaraannya. Lalu klik tombol List untuk kembali menampilkan seluruh list mobil seperti gambar dibawah ini.

Kembali **MOBIL**

Nomor Kendaraan	Merek	Jenis
B111AAA	Toyota	Agia

No. Kendaraan

Tambah List Mobil

Nomor Kendaraan   
Merek   
Jenis

Terdapat juga bagian panel yang digunakan untuk menambahkan mobil ke list. Ketika klik tombol Tambah, maka data yang diisi pada textfield nomor kendaraan, merek, dan jenis akan dimasukkan ke database mobil dan datanya akan ditampilkan di tabel. Contohnya seperti gambar dibawah ini.

Kembali

MOBIL

Nomor Kendaraan	Merek	Jenis
B111AAA	Toyota	Agya
B222BBB	Toyota	Avanza
B333CCC	Suzuki	Ertiga

No. Kendaraan

Cari

List

Hapus

Tambah List Mobil

Nomor Kendaraan

Merek

Jenis

Tambah

Kembali

MOBIL

Nomor Kendaraan	Merek	Jenis
B111AAA	Toyota	Agya
B222BBB	Toyota	Avanza
B333CCC	Suzuki	Ertiga
B444DDD	Hyundai	Stargazer

No. Kendaraan

Cari

List

Hapus

Tambah List Mobil

Nomor Kendaraan

Merek

Jenis

Tambah

#### 2.6.4 Halaman Tambah Sewa

Halaman Tambah Sewa digunakan untuk menambah daftar penyewa yang akan ditampilkan pada halaman utama. Yang perlu dilakukan hanya mengisi data yang disediakan pada halaman tersebut, lalu klik tombol Tambah.

Kembali TAMBAH SEWA

Nama

Nomor KTP

Nomor Telepon

Nomor Kendaraan

Harga per hari

Lama Sewa

Tanggal Pengambilan

Tanggal Pengembalian

Pada bagian pengisian Nomor Kendaraan disediakan dalam bentuk combo box yang berisikan list nomor kendaraan yang tersedia berdasarkan daftar pada halaman mobil sebelumnya.

Kembali TAMBAH SEWA

Nama

Nomor KTP

Nomor Telepon

Nomor Kendaraan

Harga per hari

Lama Sewa

Tanggal Pengambilan

Tanggal Pengembalian

### 2.6.5 Halaman Bayar

Halaman bayar digunakan untuk mengubah keterangan atau status pembayaran penyewa pada record di halaman utama.

Kembali

BAYAR

Nomor KTP

Cari

Nama

-

Nomor Kendaraan

-

Harga

-

Bayar

Bayar

Kembalian

-

Simpan

Yang harus dilakukan pertama kali yaitu mencari data penyewa yang ingin melakukan pembayaran. Yaitu dengan mencarinya melalui nomor KTP. Kemudian jika sudah klik cari, akan tampil nama penyewa, nomor kendaraan yang disewa, dan harga yang harus dibayarkan penyewa.

Kembali

BAYAR

Nomor KTP

2110631170115

Cari

Nama

Wanda Putra

Nomor Kendaraan

B333CCC

Harga

Rp.900000

Bayar

Bayar

Kembalian

-

Simpan

Selanjutnya admin bisa mengisi uang yang dibayarkan oleh penyewa. Kemudian klik bayar untuk menampilkan apakah ada kembalian atau tidak.

Kembali **BAYAR**

Nomor KTP

2110631170115

Cari

Nama

Wanda Putra

Nomor Kendaraan

B333CCC

Harga

Rp.900000

Bayar

1000000

Bayar

Kembalian

Rp.100000

Simpan

Jika sudah sesuai, selanjutnya admin dapat klik simpan untuk mengubah status pembayaran pada halaman utama menjadi “Sudah Dibayar”. Seperti yang terlihat pada gambar dibawah ini. Pada kolom keterangan record Wanda Putra, keterangannya menjadi “SUDAH DIBAYAR”.

DASHBOARD								
Nama	No. KTP	No. Telepon	No. Kendaraan	Lama Sewa	Pengambilan	Pengembalian	Harga	Keterangan
Tarida Grace	2110631170109	0123456789	B111AAA	1	2022-12-01	2022-12-02	300000	BELUM DIBAY...
Hadli Royan	2110631170079	0123456788	B222BBB	3	2022-12-01	2022-12-03	900000	BELUM DIBAY...
Wanda Putra	2110631170115	0123456787	B333CCC	3	2022-12-01	2022-12-03	900000	SUDAH DIBAY...

Hapus

List

Mobil

Tambah Sewa

Bayar

No. KTP

Cari

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Usaha rental mobil merupakan suatu bentuk usaha yang menawarkan jasa penyewaan mobil kepada banyak orang. Hal ini tentu menjadi sebuah peluang, karena banyaknya orang yang membutuhkan sewa mobil untuk memenuhi berbagai kebutuhan. Untuk meningkatkan kinerja pelayanan terhadap konsumen maka usaha ini perlu melakukan terobosan-terobosan agar mendapatkan feedback yang baik dari para konsumen. Oleh karena itu, kelompok Hawantar berinisiatif untuk membuat sebuah sistem aplikasi penyewaan mobil yang dapat memudahkan seorang admin dalam melayani konsumen ketika melakukan pendataan.

#### **3.2 Saran**

Kami menyadari bahwa proyek ini belum sempurna dan perlu ditingkatkan lagi. Oleh karena itu, diharapkan kedepannya sistem aplikasi ini dapat dikembangkan lagi, seperti dengan menambahkan berbagai fitur agar lebih kompleks dan efektif dan bisa juga dari segi tampilan atau interface dari sistem aplikasi ini sendiri agar lebih menarik dan user friendly.