



REPUBLIK INDONESIA  
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

# SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202114188, 4 Maret 2021

## Pencipta

Nama : **Drs. Faisal Syafar, M.Si., M.InfTech., Ph.D.**  
Alamat : Kompleks Tabaria Tower E10/23 Kelurahan Mannuruki, Kecamatan Tamalate, Makassar, SULAWESI SELATAN, 90221  
Kewarganegaraan : Indonesia

## Pemegang Hak Cipta

Nama : **Drs. Faisal Syafar., M.Si., M.InfTech., Ph.D.**  
Alamat : Kompleks Tabaria Tower E10/23 Kelurahan Mannuruki, Kecamatan Tamalate, Makassar, SULAWESI SELATAN, 90221  
Kewarganegaraan : Indonesia  
Jenis Ciptaan : **Program Komputer**  
Judul Ciptaan : **Desain Sistem Informasi Management Perpustakaan Berbasis Java**  
Tanggal dan tempat diumumkan untuk pertama kali : 23 Oktober 2015, di Makassar  
di wilayah Indonesia atau di luar wilayah Indonesia  
Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.  
Nomor pencatatan : 000240767

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA  
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL



Dr. Freddy Harris, S.H., LL.M., ACCS.  
NIP. 196611181994031001

## Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, menteri berwenang untuk mencabut surat pencatatan permohonan.

# **PROGRAM KOMPUTER**

## **DESAIN SISTEM INFORMASI MANAJEMEN PERPUSTAKAAN BERBASIS JAVA**

**Drs. Faisal Syafar, M.Si., M.InfTech., Ph.D.**

**MAKASSAR**

**Oktober 2015**

## Desain Sistem Manajemen Perpustakaan Berbasis Java

Sistem manajemen perpustakaan (SMP) adalah salah satu system yang saat ini populer yang dirancang menggunakan Bahasa pemrograman Java. Desain SMP ini dilakukan berdasarkan tahapan sebagai berikut.

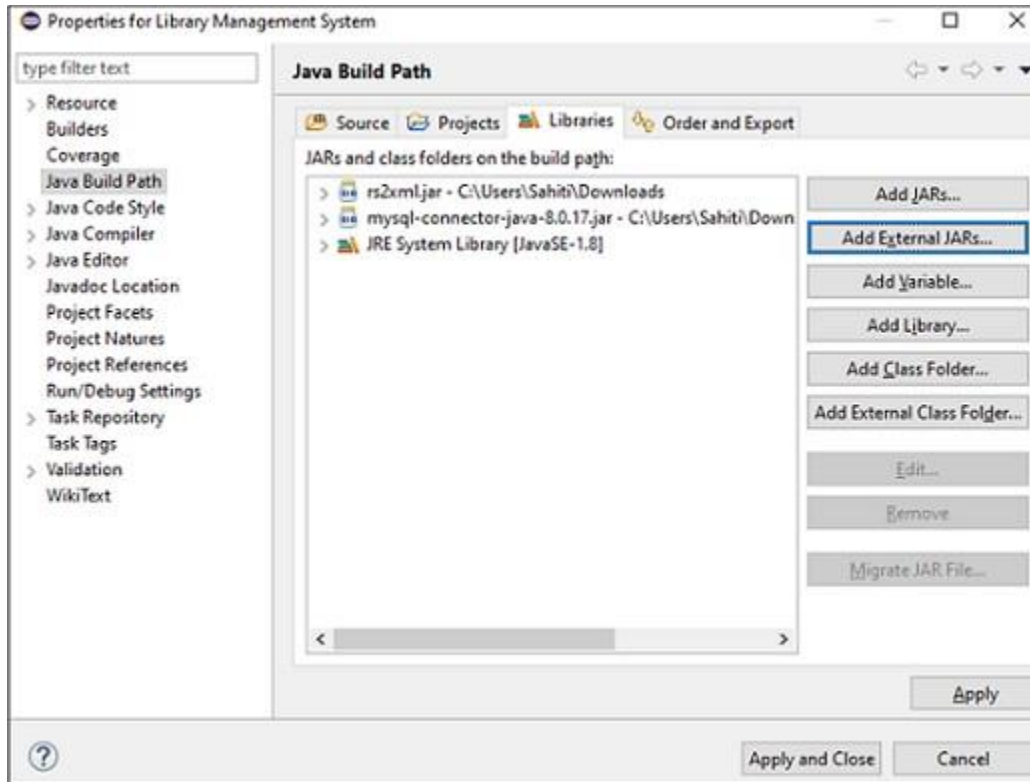
### System Requirements

Untuk menyelesaikan desain proyek SMP ini, berikut adalah spesifikasi yang diperlukan diikuti dengan langkah demi langkah pelaksanaannya:

- 1) MySQL Community Server
- 2) MySQL JDBC Connector
- 3) Java
- 4) Eclipse IDE
- 5) rs2xml.jar

Program rs2xml.jar digunakan untuk menampilkan data dalam format table. Jadi proyek yang dibuat menggunakan Eclipse IDE, selanjutnya diimport ke rs2xml.jar dan JDBC Connector.

Langkahnya, **right-click on the project**, pilih **Build Path -> Configure Build Path**. pada dialog box, yang sudah terbuka, pilih **Add External JARs**, dan tambahkan file-file JAR. Setelah itu klik **Apply and Close** dan akan muncul table berikut



## Tabel

Untuk proyek SMP ini, kami mempertimbangkan tiga table, yaitu:

- **Users** -> terdiri dari kolom {UID, Username, Password, Admin}
- **Books**-> terdiri dari kolom columns {BID, Book name, Price,Genre}
- **Issue** -> terdiri dari kolom {IID, UID, BID, IssueDate, Period,ReturnDate, Fine}

Langkah selanjutnya adalah skema tabel

## Library Management System Project Code

Code dalam proyek berbasis Bahasa pemrograman Java ini, akan kami jelaskan fungsi-fungsinya yang terbagi atas:

- Login
- Connect
- Create/Reset
- User Menu
- Admin Menu

Juga, untuk membuat GUI, saya akan menggunakan Swing. Swing adalah pustaka atau sekumpulan komponen program yang digunakan untuk membuat komponen antarmuka pengguna grafis seperti scroll bar, tombol, kotak dialog, dll.

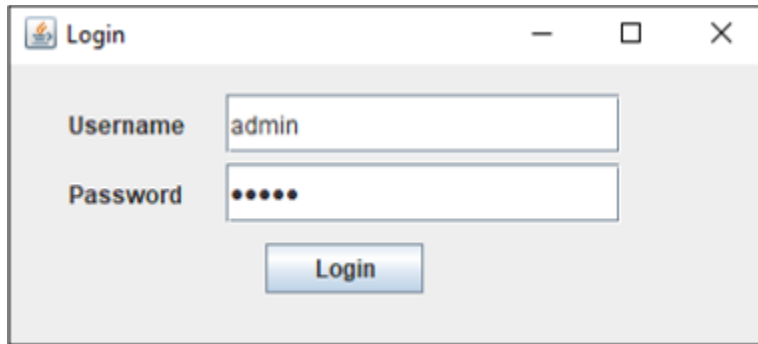
Sebelum saya membahas kode fungsi dengan Anda, izinkan saya menunjukkan kepada Anda kode untuk kelas utama dan perpustakaan yang akan diimpor:

```
1
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.sql.*;
6 import java.text.DateFormat;
7 import java.text.ParseException;
8 import java.text.SimpleDateFormat;
9 import java.util.ArrayList;
10 import java.util.Date;
11 import java.util.Locale;
12 import java.util.concurrent.TimeUnit;
13
14 import javax.swing.*;
15 import net.proteanit.sql.DbUtils;
16
17 public class main {
18
19     public static class ex{
20         public static int days=0;
21     }
22
23     public static void main(String[] args) {
24         login();
25         //create();
26     }
27 }
```

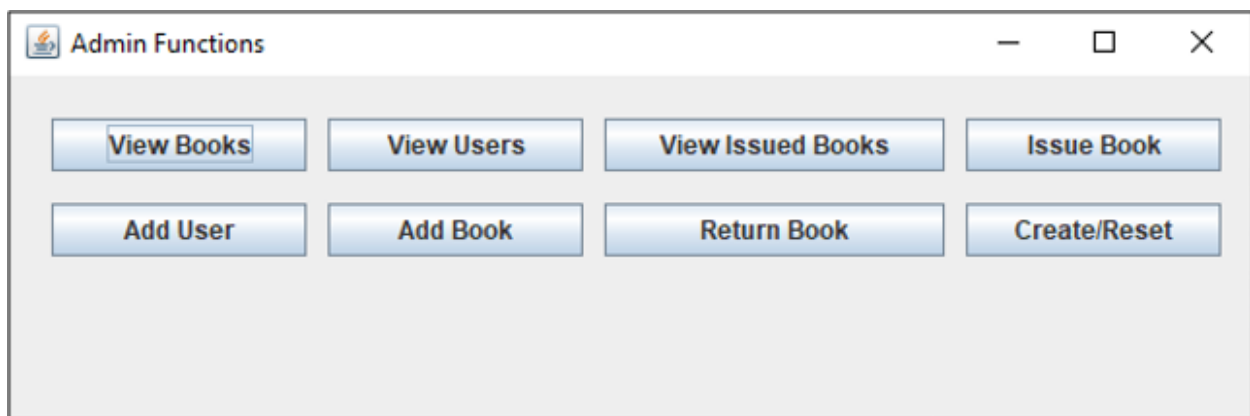
Selanjutnya mari kita pahami kode dari semua fungsi di atas.

## Login

Saya telah membuat fungsi ini untuk mengaktifkan pengguna dan login admin. Jadi, awalnya saat pengguna login untuk pertama kali, pengguna tersebut secara default akan menjadi admin, dan nama pengguna serta sandi adalah {admin, admin}. Lihat di bawah.



Untuk skema ini, saya hanya mempertimbangkan satu admin. Jadi, setelah pengguna login sebagai admin, dia akan diarahkan ke menu admin seperti di bawah ini. Saya akan membahas fungsi admin di bagian menu admin.



Kembali ke fungsi login, lihat kode di bawah ini:

```
1 public static void login() {
2
3     JFrame f=new JFrame("Login");//creating instance of JFrame
4     JLabel l1,l2;
5     l1=new JLabel("Username"); //Create label Username
6     l1.setBounds(30,15, 100,30); //x axis, y axis, width, height
7
8     l2=new JLabel("Password"); //Create label Password
9     l2.setBounds(30,50, 100,30);
10
11    JTextField F_user = new JTextField(); //Create text field for username
12    F_user.setBounds(110, 15, 200, 30);
13
14    JPasswordField F_pass=new JPasswordField(); //Create text field for password
15    F_pass.setBounds(110, 50, 200, 30);
16
17    JButton login_but=new JButton("Login");//creating instance of JButton for Login B
18    login_but.setBounds(130,90,80,25);//Dimensions for button
19    login_but.addActionListener(new ActionListener() { //Perform action
```

```

19     public void actionPerformed(ActionEvent e){
20
21         String username = F_user.getText(); //Store username entered by the user in
22         String password = F_pass.getText(); //Store password entered by the user in
23
24         if(username.equals("")) //If username is null
25         {
26             JOptionPane.showMessageDialog(null,"Please enter username"); //Display d
27         }
28         else if(password.equals("")) //If password is null
29         {
30             JOptionPane.showMessageDialog(null,"Please enter password"); //Display d
31         }
32         else{ //If both the fields are present then to login the user, check wether
33             //System.out.println("Login connect");
34             Connection connection=connect(); //Connect to the database
35             try
36             {
37                 Statement stmt = connection.createStatement();
38                 stmt.executeUpdate("USE LIBRARY"); //Use the database with the name "L
39                 String st = ("SELECT * FROM USERS WHERE USERNAME='"+username+"' AND PA
40 from users
41                 ResultSet rs = stmt.executeQuery(st); //Execute query
42                 if(rs.next()==false) { //Move pointer below
43                     System.out.print("No user");
44                     JOptionPane.showMessageDialog(null,"Wrong Username/Password!"); //
45                 }
46                 else {
47                     f.dispose();
48                     rs.beforeFirst(); //Move the pointer above
49                     while(rs.next())
50                     {
51                         String admin = rs.getString("ADMIN"); //user is admin
52                         //System.out.println(admin);
53                         String UID = rs.getString("UID"); //Get user ID of the user
54                         if(admin.equals("1")) { //If boolean value 1
55                             admin menu(); //redirect to admin menu
56                         }
57                         else{
58                             user menu(UID); //redirect to user menu for that user ID
59                         }
60                     }
61                 }
62             }
63             catch (Exception ex) {
64                 ex.printStackTrace();
65             }
66         }
67     }
68 }
69 });
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

65     f.add(l1); // add label1 i.e. for username
66     f.add(l2); // add label2 i.e. for password
67
68     f.setSize(400,180);//400 width and 500 height
69     f.setLayout(null);//using no layout managers
70     f.setVisible(true);//making the frame visible
71     f.setLocationRelativeTo(null);
72

```

## Connect

Fungsi koneksi digunakan untuk menghubungkan database ke GUI. Jadi, untuk melakukan itu, saya telah menyebutkan kode di bawah ini:

```

1
2 public static Connection connect ()
3 {
4     try {
5         Class.forName("com.mysql.cj.jdbc.Driver");
6         //System.out.println("Loaded driver");
7         Connection con = DriverManager.getConnection("jdbc:mysql://localhost/mysql?u
8         //System.out.println("Connected to MySQL");
9         return con;
10    }
11    catch (Exception ex) {
12        ex.printStackTrace();
13    }
14    return null;
15 }

```

Dalam fungsi di atas, kami menghubungkan **database MySQL** kami dengan **username "root"** dan kata sandi **"edureka"** ke aplikasi kami. Sekarang, setelah aplikasi terhubung ke database, langkah selanjutnya adalah membuat atau mengatur ulang database. Jadi, selanjutnya dalam artikel ini tentang Proyek Sistem Manajemen Perpustakaan di Java, mari kita bahas fungsi Buat ..

## Create

Fungsi create digunakan untuk membuat database, tabel dan menambahkan data ke dalam tabel tersebut. Jadi, untuk melakukan itu, pernyataan SQL akan digunakan seperti di bawah ini.

```

1 public static void create() {
2     try {
3         Connection connection=connect();
4         ResultSet resultSet = connection.getMetaData().getCatalogs();
5         //iterate each catalog in the ResultSet
6         while (resultSet.next()) {
7             // Get the database name, which is at position 1
8             String databaseName = resultSet.getString(1);
9             if(databaseName.equals("library")) {
10                //System.out.print("yes");

```



```

9         Statement stmt = connection.createStatement();
10        //Drop database if it pre-exists to reset the complete database
11        String sql = "DROP DATABASE library";
12        stmt.executeUpdate(sql);
13    }
14    }
15    Statement stmt = connection.createStatement();
16    String sql = "CREATE DATABASE LIBRARY"; //Create Database
17    stmt.executeUpdate(sql);
18    stmt.executeUpdate("USE LIBRARY"); //Use Database
19    //Create Users Table
20    String sql1 = "CREATE TABLE USERS (UID INT NOT NULL AUTO INCREMENT PRIMARY
21    BOOLEAN) ";
22    stmt.executeUpdate(sql1);
23    //Insert into users table
24    stmt.executeUpdate("INSERT INTO USERS (USERNAME, PASSWORD, ADMIN) VALUES ('a
25    //Create Books table
26    stmt.executeUpdate("CREATE TABLE BOOKS (BID INT NOT NULL AUTO INCREMENT PRI
27    INT)");
28    //Create Issued Table
29    stmt.executeUpdate("CREATE TABLE ISSUED (IID INT NOT NULL AUTO_INCREMENT PR
30    RETURN_DATE VARCHAR(20), PERIOD INT, FINE INT)");
31    //Insert into books table
32    stmt.executeUpdate("INSERT INTO BOOKS (BNAME, GENRE, PRICE) VALUES ('War an
33    300), ('The Perfect Murder', 'Mystery', 150), ('Accidental Presidents', 'Biography',
34    resultSet.close();
35    }
36    catch (Exception ex) {
37        ex.printStackTrace();
38    }
39    }

```

Sekarang, setelah kita membuat database, terhubung dengan GUI dan mengaktifkan fungsi login, selanjutnya dalam artikel ini tentang Proyek Sistem Manajemen Perpustakaan di Java, mari kita sekarang membahas fungsi Menu Pengguna.

## User Menu

Menu Pengguna dirancang untuk menampilkan detail dari semua buku yang ada di perpustakaan dan buku yang diterbitkan oleh pengguna.

```

1    public static void user_menu(String UID) {
2
3        JFrame f=new JFrame("User Functions"); //Give dialog box name as User functions
4        //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //Exit user menu on closing
5        JButton view_but=new JButton("View Books"); //creating instance of JButton
6        view_but.setBounds(20,20,120,25); //x axis, y axis, width, height
7        view_but.addActionListener(new ActionListener() {
8            public void actionPerformed(ActionEvent e){
9
10               JFrame f = new JFrame("Books Available"); //View books stored in database

```

```

10         //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11
12
13         Connection connection = connect();
14         String sql="select * from BOOKS"; //Retreive data from database
15         try{
16             Statement stmt = connection.createStatement(); //connect to database
17             stmt.executeUpdate("USE LIBRARY"); // use librabry
18             stmt=connection.createStatement();
19             ResultSet rs=stmt.executeQuery(sql);
20             JTable book_list= new JTable(); //show data in table format
21             book_list.setModel(DbUtils.resultSetToTableModel(rs));
22
23             JScrollPane scrollPane = new JScrollPane(book_list); //enable scroll
24
25             f.add(scrollPane); //add scroll bar
26             f.setSize(800, 400); //set dimensions of view books frame
27             f.setVisible(true);
28             f.setLocationRelativeTo(null);
29         } catch (SQLException e1) {
30             // TODO Auto-generated catch block
31             JOptionPane.showMessageDialog(null, e1);
32         }
33     }
34     }
35     );
36
37     JButton my_book=new JButton("My Books");//creating instance of JButton
38     my_book.setBounds(150,20,120,25);//x axis, y axis, width, height
39     my_book.addActionListener(new ActionListener() { //Perform action
40         public void actionPerformed(ActionEvent e){
41
42             JFrame f = new JFrame("My Books"); //View books issued by user
43             //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44             int UID_int = Integer.parseInt(UID); //Pass user ID
45
46             //.iid,issued.uid,issued.bid,issued.issued_date,issued.return_date,issue
47             Connection connection = connect(); //connect to database
48             //retrieve data
49             String sql="select distinct issued.*,books.bname,books.genre,books.price
50 and (books.bid in (select bid from issued where issued.uid="+UID_int+")) group by i
51 String sql1 = "select bid from issued where uid="+UID_int;
52         try{
53             Statement stmt = connection.createStatement();
54             //use database
55             stmt.executeUpdate("USE LIBRARY");
56             stmt=connection.createStatement();
57             //store in array
58             ArrayList books_list = new ArrayList();

```

```

56         ResultSet rs=stmt.executeQuery(sql);
57         JTable book_list= new JTable(); //store data in table format
58         book_list.setModel(DbUtils.resultSetToTableModel(rs));
59         //enable scroll bar
60         JScrollPane scrollPane = new JScrollPane(book list);
61
62         f.add(scrollPane); //add scroll bar
63         f.setSize(800, 400); //set dimensions of my books frame
64         f.setVisible(true);
65         f.setLocationRelativeTo(null);
66     } catch (SQLException e1) {
67         // TODO Auto-generated catch block
68         JOptionPane.showMessageDialog(null, e1);
69     }
70 }
71 }
72 }
73 }
74 f.add(my_book); //add my books
75 f.add(view_but); // add view books
76 f.setSize(300,100);//400 width and 500 height
77 f.setLayout(null);//using no layout managers
78 f.setVisible(true);//making the frame visible
79 f.setLocationRelativeTo(null);
80 }

```

Selanjutnya, pada artikel tentang Proyek Sistem Manajemen Perpustakaan di Java ini, mari kita bahas kode untuk fungsi Menu Admin.

## Admin Menu

Menu Admin dirancang untuk menampilkan detail pengguna, buku, buku terbitan, menambah buku, mengembalikan buku, menambah pengguna, dan membuat atau mengatur ulang database.

```

1     public static void admin_menu() {
2
3
4         JFrame f=new JFrame("Admin Functions"); //Give dialog box name as admin function
5         //f.setDefaultCloseOperation(JFrame.EXIT ON CLOSE); //
6
7         JButton create_but=new JButton("Create/Reset");//creating instance of JButton to
8         create_but.setBounds(450,60,120,25);//x axis, y axis, width, height
9         create_but.addActionListener(new ActionListener() { //Perform action
10            public void actionPerformed(ActionEvent e) {
11
12                create(); //Call create function
13                JOptionPane.showMessageDialog(null,"Database Created/Reset!"); //Open a

```



```

60         JTable book_list= new JTable();
61         book_list.setModel(DbUtils.resultSetToTableModel(rs));
62         //mention scroll bar
63         JScrollPane scrollPane = new JScrollPane(book_list);
64
65         f.add(scrollPane); //add scrollpane
66         f.setSize(800, 400); //set size for frame
67         f.setVisible(true);
68         f.setLocationRelativeTo(null);
69     } catch (SQLException e1) {
70         // TODO Auto-generated catch block
71         JOptionPane.showMessageDialog(null, e1);
72     }
73 }
74 );
75
76 JButton issued_but=new JButton("View Issued Books");//creating instance of JButton
77 issued_but.setBounds(280,20,160,25);//x axis, y axis, width, height
78 issued_but.addActionListener(new ActionListener() {
79     public void actionPerformed(ActionEvent e) {
80
81         JFrame f = new JFrame("Users List");
82         //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
83
84         Connection connection = connect();
85         String sql="select * from issued";
86         try {
87             Statement stmt = connection.createStatement();
88             stmt.executeUpdate("USE LIBRARY");
89             stmt=connection.createStatement();
90             ResultSet rs=stmt.executeQuery(sql);
91             JTable book_list= new JTable();
92             book_list.setModel(DbUtils.resultSetToTableModel(rs));
93
94             JScrollPane scrollPane = new JScrollPane(book_list);
95
96             f.add(scrollPane);
97             f.setSize(800, 400);
98             f.setVisible(true);
99             f.setLocationRelativeTo(null);
100        } catch (SQLException e1) {
101            // TODO Auto-generated catch block
102            JOptionPane.showMessageDialog(null, e1);
103        }
104    }
105 );

```

```

106 JButton add_user=new JButton("Add User"); //creating instance of JButton to add
107 add_user.setBounds(20,60,120,25); //set dimensions for button
108
109 add_user.addActionListener(new ActionListener() {
110     public void actionPerformed(ActionEvent e){
111
112         JFrame g = new JFrame("Enter User Details"); //Frame to enter user d
113         //g.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
114         //Create label
115         JLabel l1,l2;
116         l1=new JLabel("Username"); //label 1 for username
117         l1.setBounds(30,15, 100,30);
118
119         l2=new JLabel("Password"); //label 2 for password
120         l2.setBounds(30,50, 100,30);
121
122         //set text field for username
123         JTextField F_user = new JTextField();
124         F_user.setBounds(110, 15, 200, 30);
125
126         //set text field for password
127         JPasswordField F_pass=new JPasswordField();
128         F_pass.setBounds(110, 50, 200, 30);
129         //set radio button for admin
130         JRadioButton a1 = new JRadioButton("Admin");
131         a1.setBounds(55, 80, 200,30);
132         //set radio button for user
133         JRadioButton a2 = new JRadioButton("User");
134         a2.setBounds(130, 80, 200,30);
135         //add radio buttons
136         ButtonGroup bg=new ButtonGroup();
137         bg.add(a1);bg.add(a2);
138
139         JButton create_but=new JButton("Create");//creating instance of JBut
140         create_but.setBounds(130,130,80,25);//x axis, y axis, width, height
141         create_but.addActionListener(new ActionListener() {
142
143             public void actionPerformed(ActionEvent e){
144
145                 String username = F_user.getText();
146                 String password = F_pass.getText();
147                 Boolean admin = false;
148
149                 if(a1.isSelected()) {
150                     admin=true;
151                 }
152
153                 Connection connection = connect();
154
155                 try{
156                     Statement stmt = connection.createStatement();

```

```

152         stmt.executeUpdate("USE LIBRARY");
153         stmt.executeUpdate("INSERT INTO USERS (USERNAME,PASSWORD,ADMIN)
154         JOptionPane.showMessageDialog(null,"User added!");
155         g.dispose();
156     }
157
158     catch (SQLException e1) {
159         // TODO Auto-generated catch block
160         JOptionPane.showMessageDialog(null, e1);
161     }
162 }
163
164 });
165
166
167     g.add(create_but);
168     g.add(a2);
169     g.add(a1);
170     g.add(l1);
171     g.add(l2);
172     g.add(F_user);
173     g.add(F_pass);
174     g.setSize(350,200);//400 width and 500 height
175     g.setLayout(null);//using no layout managers
176     g.setVisible(true);//making the frame visible
177     g.setLocationRelativeTo(null);
178 }
179 });
180
181
182 JButton add_book=new JButton("Add Book"); //creating instance of JButton for add
183 add_book.setBounds(150,60,120,25);
184
185 add_book.addActionListener(new ActionListener() {
186     public void actionPerformed(ActionEvent e){
187         //set frame wot enter book details
188         JFrame g = new JFrame("Enter Book Details");
189         //g.setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
190         // set labels
191         JLabel l1,l2,l3;
192         l1=new JLabel("Book Name"); //lebel 1 for book name
193         l1.setBounds(30,15, 100,30);
194
195         l2=new JLabel("Genre"); //label 2 for genre
196         l2.setBounds(30,53, 100,30);
197
198         l3=new JLabel("Price"); //label 2 for price
199         l3.setBounds(30,90, 100,30);

```

```

198
199 //set text field for book name
200 JTextField F_bname = new JTextField();
201 F_bname.setBounds(110, 15, 200, 30);
202
203 //set text field for genre
204 JTextField F_genre=new JTextField();
205 F_genre.setBounds(110, 53, 200, 30);
206 //set text field for price
207 JTextField F_price=new JTextField();
208 F_price.setBounds(110, 90, 200, 30);
209
210 JButton create_but=new JButton("Submit");//creating instance of JButton
211 create_but.setBounds(130,130,80,25);//x axis, y axis, width, height
212 create_but.addActionListener(new ActionListener() {
213
214     public void actionPerformed(ActionEvent e) {
215         // assign the book name, genre, price
216         String bname = F_bname.getText();
217         String genre = F_genre.getText();
218         String price = F_price.getText();
219         //convert price of integer to int
220         int price_int = Integer.parseInt(price);
221
222         Connection connection = connect();
223
224         try{
225             Statement stmt = connection.createStatement();
226             stmt.executeUpdate("USE LIBRARY");
227             stmt.executeUpdate("INSERT INTO BOOKS (BNAME,GENRE,PRICE) VALUE
228             JOptionPane.showMessageDialog(null,"Book added!");
229             g.dispose();
230         }
231
232         catch (SQLException e1) {
233             // TODO Auto-generated catch block
234             JOptionPane.showMessageDialog(null, e1);
235         }
236     }
237
238 });
239
240 g.add(l3);
241 g.add(create_but);
242 g.add(l1);
243 g.add(l2);
244 g.add(F_bname);
245 g.add(F_genre);
246 g.add(F_price);
247 g.setSize(350,200);//400 width and 500 height
248 g.setLayout(null);//using no layout managers

```



```

244         g.setVisible(true); //making the frame visible
245         g.setLocationRelativeTo(null);
246     }
247 });
248
249
250 JButton issue_book=new JButton("Issue Book"); //creating instance of JButton to
251 issue_book.setBounds(450,20,120,25);
252
253 issue_book.addActionListener(new ActionListener() {
254     public void actionPerformed(ActionEvent e) {
255         //enter details
256         JFrame g = new JFrame("Enter Details");
257         //g.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
258         //create labels
259         JLabel l1,l2,l3,l4;
260         l1=new JLabel("Book ID(BID)"); // Label 1 for Book ID
261         l1.setBounds(30,15, 100,30);
262
263         l2=new JLabel("User ID(UID)"); //Label 2 for user ID
264         l2.setBounds(30,53, 100,30);
265
266         l3=new JLabel("Period(days)"); //Label 3 for period
267         l3.setBounds(30,90, 100,30);
268
269         l4=new JLabel("Issued Date(DD-MM-YYYY)"); //Label 4 for issue date
270         l4.setBounds(30,127, 150,30);
271
272         JTextField F_bid = new JTextField();
273         F_bid.setBounds(110, 15, 200, 30);
274
275         JTextField F_uid=new JTextField();
276         F_uid.setBounds(110, 53, 200, 30);
277
278         JTextField F_period=new JTextField();
279         F_period.setBounds(110, 90, 200, 30);
280
281         JTextField F_issue=new JTextField();
282         F_issue.setBounds(180, 130, 130, 30);
283
284         JButton create_but=new JButton("Submit");//creating instance of JBut
285         create_but.setBounds(130,170,80,25);//x axis, y axis, width, height
286         create_but.addActionListener(new ActionListener() {
287
288             public void actionPerformed(ActionEvent e) {
289
290                 String uid = F_uid.getText();
291                 String bid = F_bid.getText();

```

```

290         String period = F_period.getText();
291         String issued_date = F_issue.getText();
292
293         int period_int = Integer.parseInt(period);
294
295         Connection connection = connect();
296
297         try {
298             Statement stmt = connection.createStatement();
299             stmt.executeUpdate("USE LIBRARY");
300             stmt.executeUpdate("INSERT INTO ISSUED(UID,BID,ISSUED DATE,PER
('"+uid+"','"+bid+"','"+issued date+"','"+period int+"")");
301             JOptionPane.showMessageDialog(null, "Book Issued!");
302             g.dispose();
303         }
304
305         catch (SQLException e1) {
306             // TODO Auto-generated catch block
307             JOptionPane.showMessageDialog(null, e1);
308         }
309     }
310
311     });
312
313
314         g.add(l3);
315         g.add(l4);
316         g.add(create_but);
317         g.add(l1);
318         g.add(l2);
319         g.add(F_uid);
320         g.add(F_bid);
321         g.add(F_period);
322         g.add(F_issue);
323         g.setSize(350,250);//400 width and 500 height
324         g.setLayout(null);//using no layout managers
325         g.setVisible(true);//making the frame visible
326         g.setLocationRelativeTo(null);
327     }
328     });
329
330     JButton return_book=new JButton("Return Book"); //creating instance of JButton t
331     return_book.setBounds(280,60,160,25);
332
333     return_book.addActionListener(new ActionListener() {
334         public void actionPerformed(ActionEvent e) {
335
336                 JFrame g = new JFrame("Enter Details");

```

```

336 //g.setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
337 //set labels
338 JLabel l1,l2,l3,l4;
339 l1=new JLabel("Issue ID(IID)"); //Label 1 for Issue ID
340 l1.setBounds(30,15, 100,30);
341
342
343 l4=new JLabel("Return Date(DD-MM-YYYY)");
344 l4.setBounds(30,50, 150,30);
345
346 JTextField F_iid = new JTextField();
347 F_iid.setBounds(110, 15, 200, 30);
348
349 JTextField F_return=new JTextField();
350 F_return.setBounds(180, 50, 130, 30);
351
352 JButton create_but=new JButton("Return");//creating instance of JButton
353 create_but.setBounds(130,170,80,25);//x axis, y axis, width, height
354 create_but.addActionListener(new ActionListener() {
355
356     public void actionPerformed(ActionEvent e) {
357
358         String iid = F_iid.getText();
359         String return_date = F_return.getText();
360
361         Connection connection = connect();
362
363         try {
364             Statement stmt = connection.createStatement();
365             stmt.executeUpdate("USE LIBRARY");
366             //Intialize date1 with NULL value
367             String date1=null;
368             String date2=return_date; //Intialize date2 with return date
369
370             //select issue date
371             ResultSet rs = stmt.executeQuery("SELECT ISSUED_DATE FROM ISSU
372             while (rs.next()) {
373                 date1 = rs.getString(1);
374             }
375
376             try {
377                 Date date_1=new SimpleDateFormat("dd-MM-yyyy").parse(date1);
378                 Date date_2=new SimpleDateFormat("dd-MM-yyyy").parse(date2);
379                 //subtract the dates and store in diff
380                 long diff = date_2.getTime() - date_1.getTime();
381                 //Convert diff from milliseconds to days
382                 ex.days=(int) (TimeUnit.DAYS.convert(diff, TimeUnit.MILL

```

```

382         // TODO Auto-generated catch block
383         e1.printStackTrace();
384     }
385
386     //update return date
387     stmt.executeUpdate("UPDATE ISSUED SET RETURN_DATE='"+return_da
388     g.dispose();
389
390
391     Connection connection1 = connect();
392     Statement stmt1 = connection1.createStatement();
393     stmt1.executeUpdate("USE LIBRARY");
394     ResultSet rs1 = stmt1.executeQuery("SELECT PERIOD FROM ISSUED W
395     String diff=null;
396     while (rs1.next()) {
397         diff = rs1.getString(1);
398     }
399     int diff_int = Integer.parseInt(diff);
400     if(ex.days&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&gt;diff_int)
401     calculcate fine
402
403     //System.out.println(ex.days);
404     int fine = (ex.days-diff_int)*10; //fine for every day after
405     //update fine in the system
406     stmt1.executeUpdate("UPDATE ISSUED SET FINE="+fine+" WHERE
407     String fine_str = ("Fine: Rs. "+fine);
408     JOptionPane.showMessageDialog(null,fine_str);
409
410     }
411
412     JOptionPane.showMessageDialog(null,"Book Returned!");
413
414     }
415
416     catch (SQLException e1) {
417         // TODO Auto-generated catch block
418         JOptionPane.showMessageDialog(null, e1);
419     }
420
421     });
422     g.add(l4);
423     g.add(create_but);
424     g.add(l1);
425     g.add(F_iid);
426     g.add(F_return);
427     g.setSize(350,250);//400 width and 500 height
428     g.setLayout(null);//using no layout managers
429     g.setVisible(true);//making the frame visible

```

```

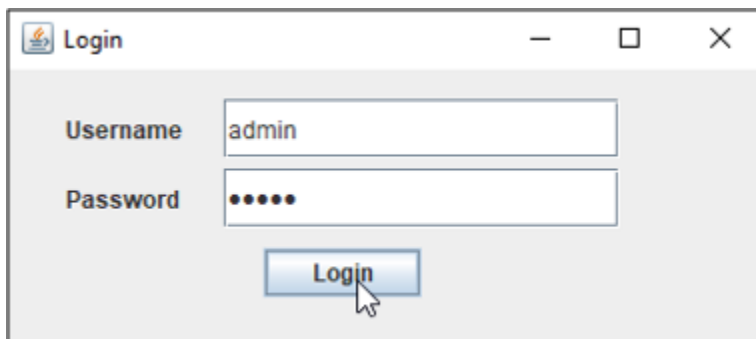
428         g.setLocationRelativeTo(null);
429     }
430 });
431     f.add(create_but);
432     f.add(return_book);
433     f.add(issue_book);
434     f.add(add_book);
435     f.add(issued_but);
436     f.add(users_but);
437     f.add(view_but);
438     f.add(add_user);
439     f.setSize(600,200);//400 width and 500 height
440     f.setLayout(null);//using no layout managers
441     f.setVisible(true);//making the frame visible
442     f.setLocationRelativeTo(null);
443 }

```

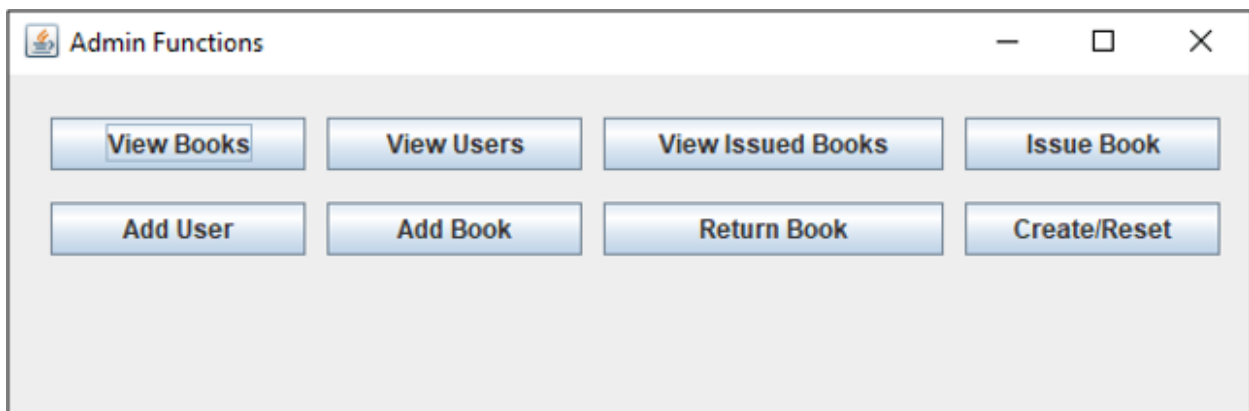
Sekarang Anda telah memahami semua fungsinya, mari kita jalankan proyek sistem manajemen perpustakaan kami di Java dan lihat hasilnya.

### Output:

Jalankan aplikasi dengan mengklik tombol run. Setelah, Anda mengeksekusi Anda akan melihat kotak dialog di bawah ini. Pada kotak dialog di bawah ini, sebutkan nama pengguna dan sandi sebagai {admin, admin}. Kemudian klik tombol Login.



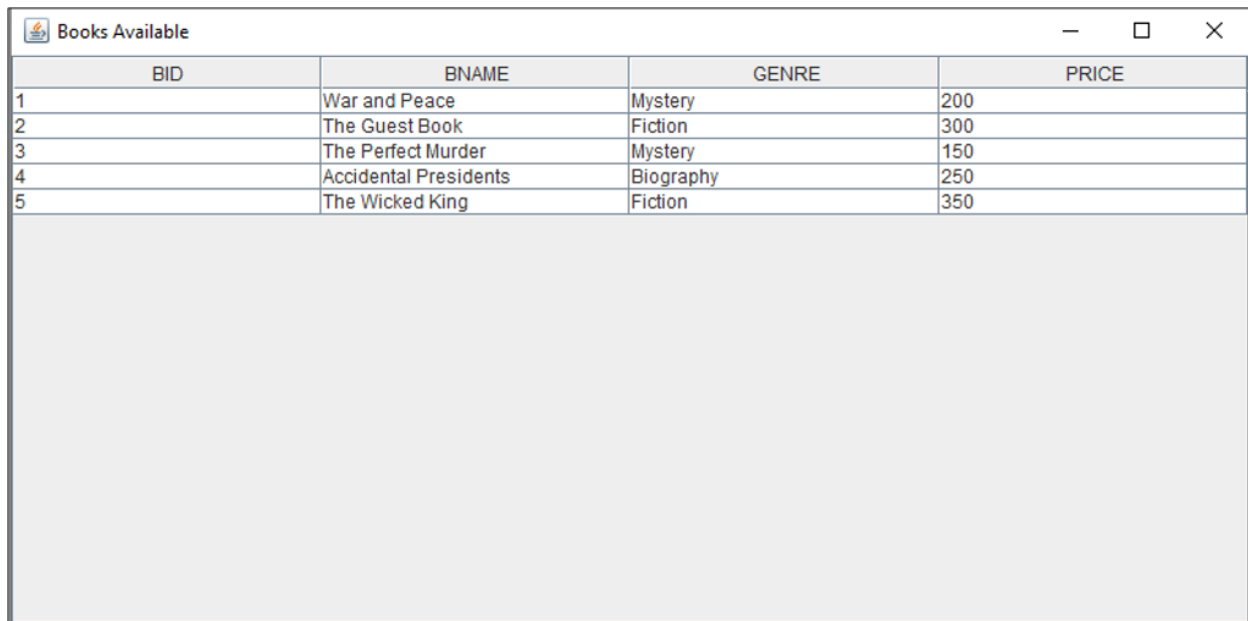
Setelah Anda mengklik tombol Login, Anda akan melihat kotak dialog di bawah ini terbuka.



Di sini Anda memiliki berbagai opsi yang dapat Anda jelajahi. Jadi, mari kita mulai dengan yang pertama:

### View Books

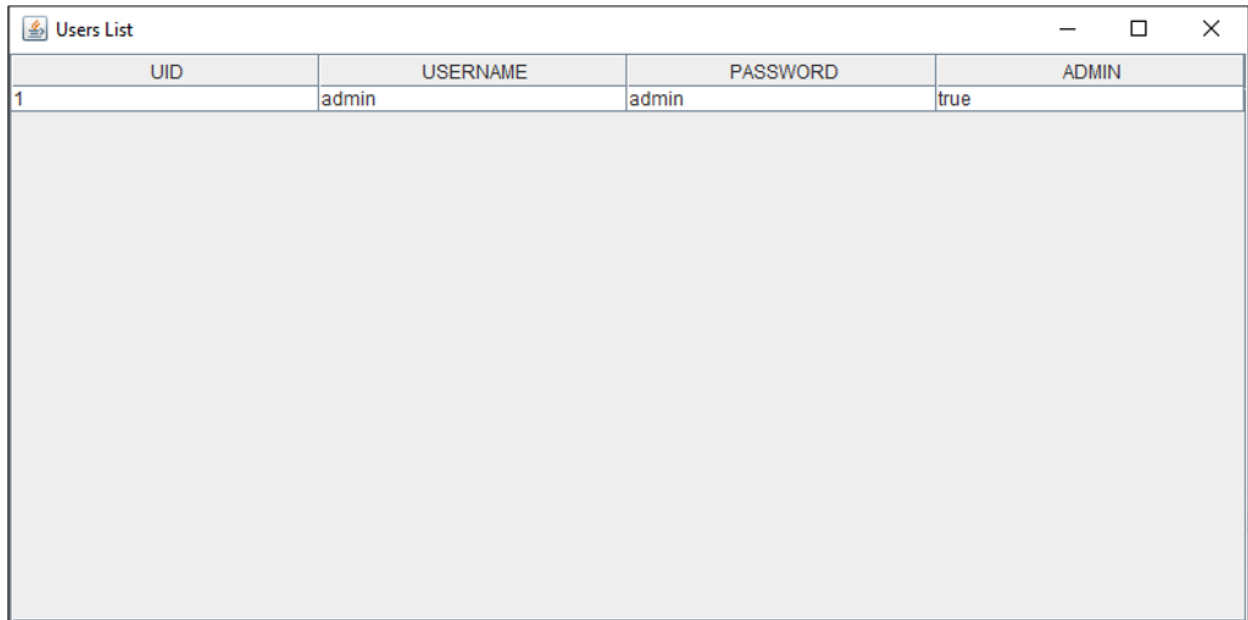
Sekali, Anda mengklik tombol Lihat Buku, Anda akan melihat bingkai di bawah ini menampilkan semua buku yang ada di database, dengan detailnya.



BID	BNAME	GENRE	PRICE
1	War and Peace	Mystery	200
2	The Guest Book	Fiction	300
3	The Perfect Murder	Mystery	150
4	Accidental Presidents	Biography	250
5	The Wicked King	Fiction	350

### View Users

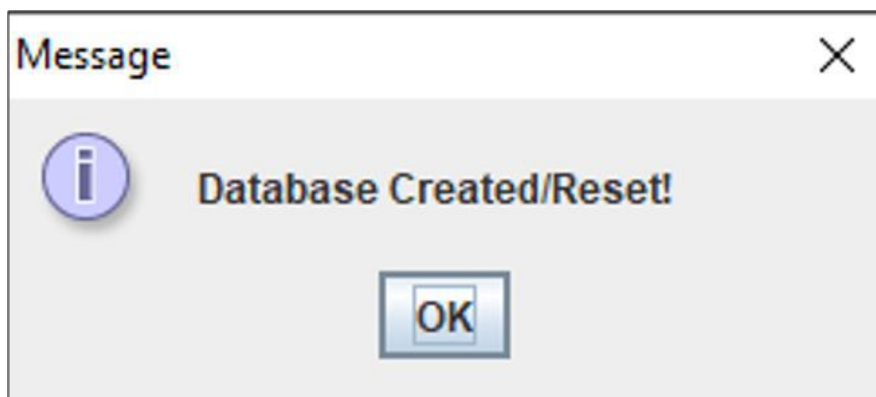
Tombol Lihat Pengguna digunakan untuk melihat pengguna saat ini di sistem. Karena kami hanya memiliki satu pengguna yang hadir yaitu admin, itu akan menunjukkan kepada Anda output seperti di bawah ini:



UID	USERNAME	PASSWORD	ADMIN
1	admin	admin	true

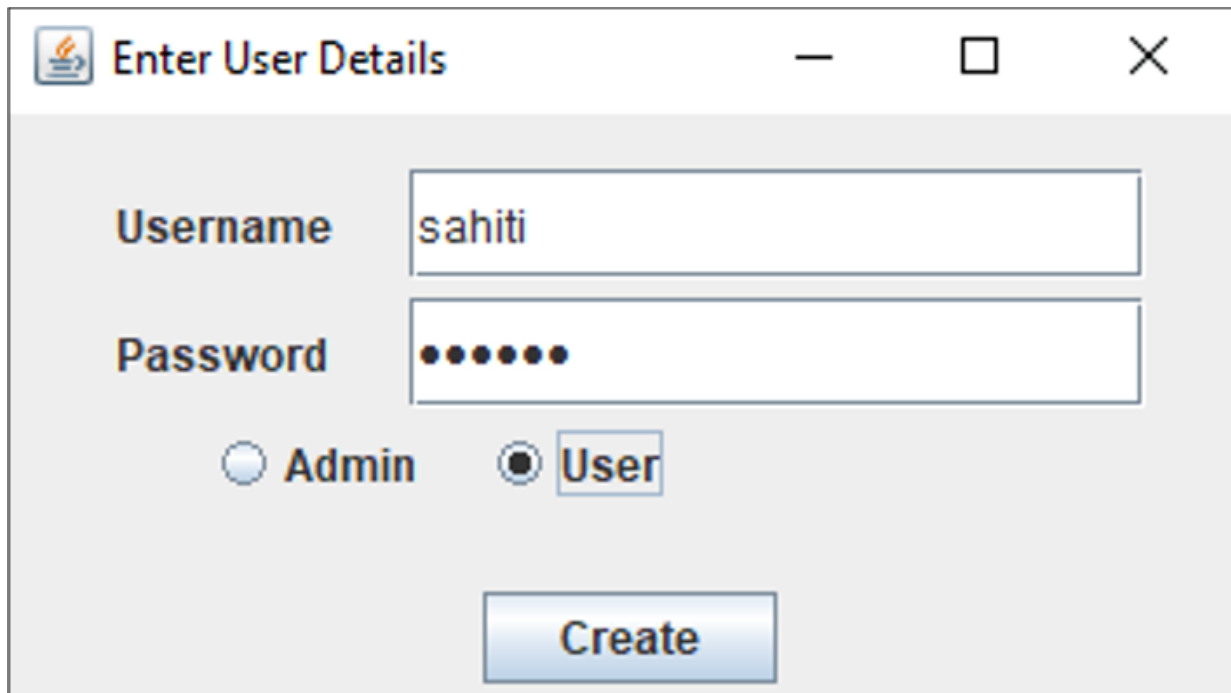
### Create/Reset

Fungsionalitas ini digunakan untuk membuat atau mengatur ulang database. Jadi, setelah Anda mengklik tombol Buat / Istirahat, Anda akan melihat output di bawah ini:



### Add User

Untuk menambahkan pengguna, klik pada opsi "Add User" dan sebutkan detail seperti **username**, **password** and **choose the radio button user or admin**. Secara default, itu akan menjadi pengguna. Kemudian, klik Creat.



Enter User Details

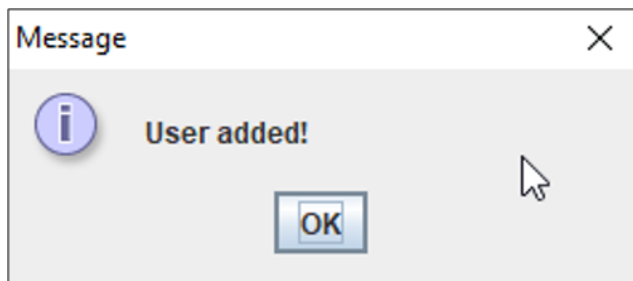
Username: sahiti

Password: ●●●●●●

Admin  User

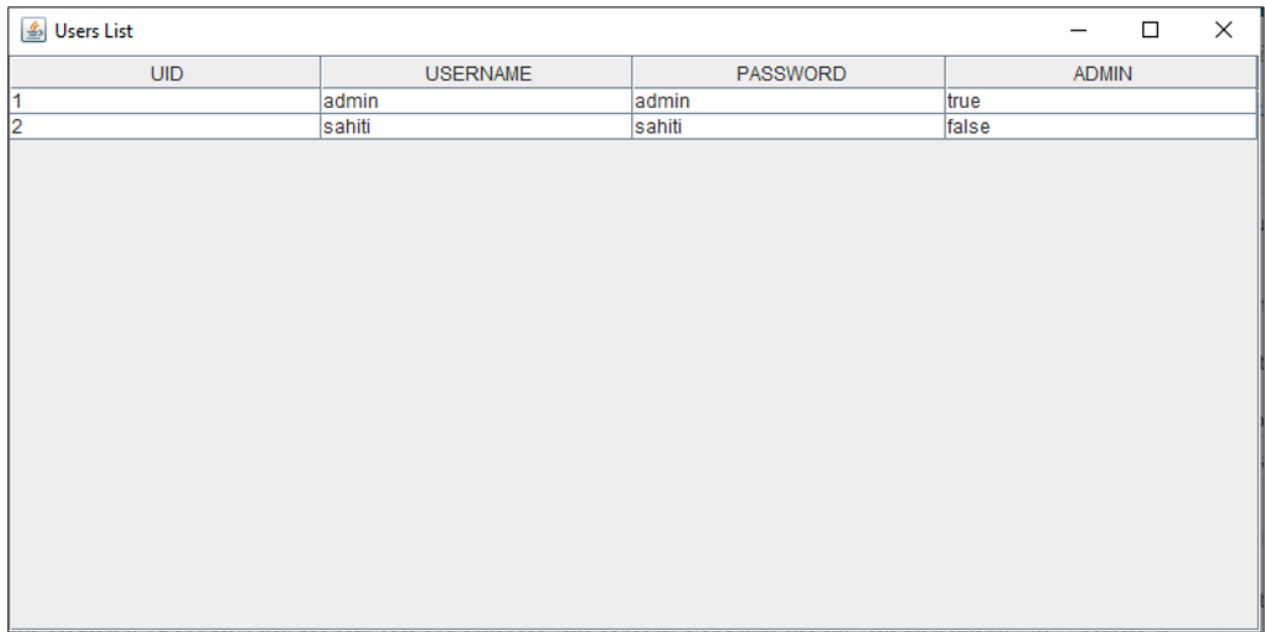
Create

Setelah Uaser dibuat, Anda akan melihat output seperti di bawah ini:



Now, again if you click on **View Users button**, you will see the below output:





UID	USERNAME	PASSWORD	ADMIN
1	admin	admin	true
2	sahti	sahti	false

Baiklah, jadi sekarang kami telah menambahkan pengguna. Katakanlah, pengguna tertentu ingin menerbitkan buku. Untuk melakukan itu, pengguna harus memilih opsi Issue Book.

### **Issue Book**

Suppose, if you are the user, once you click on the **Issue Book button**, you have to mention the **Book ID, User ID, Period(Number of days for issuing the book)**, and the **Issue Date** as follows:

Misalkan, jika Anda adalah pengguna, setelah Anda mengklik tombol **Issue Book**, Anda harus menyebutkan **Book ID, User ID, Period (Number of days for issuing the book)**, dan Tanggal Penerbitan sebagai berikut:

Enter Details

Book ID(BID) 3

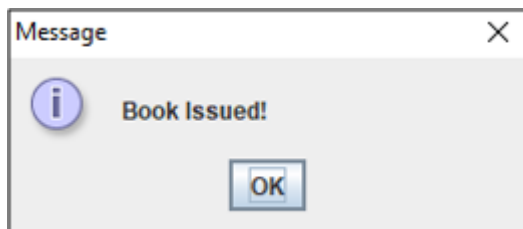
User ID(UID) 2

Period(days) 10

Issued Date(DD-MM-YYYY) 02-09-2019

Submit

Kemudian klik **Submit**. Setelah, Anda mengklik **Submit**, Anda akan melihat kotak dialog di bawah ini:



Sekarang, jika Anda ingin melihat detail buku yang diterbitkan, Anda dapat menggunakan fungsi Lihat Buku yang Diterbitkan.

### **View Issued Books**

Setelah Anda mengklik tombol ini, Anda akan melihat output berikut:

IID	UID	BID	ISSUED_DATE	RETURN_DATE	PERIOD	FINE
1	2	3	02-09-2019		10	

Sekarang jika pengguna masuk ke sistem, menggunakan fungsi login, seperti di bawah ini:

Kemudian pengguna akan melihat di bawah ini Menu pengguna.

Here, the user can **view all the books** in the database by **using the View Books option** and the **books issued by the user** in the **My Books section** as below:

Di sini, pengguna dapat melihat semua buku dalam (**view all the books** database dengan menggunakan opsi **using the View Books** yang diterbitkan oleh pengguna di **My Books section** seperti di bawah ini:

BID	BNAME	GENRE	PRICE
1	War and Peace	Mystery	200
2	The Guest Book	Fiction	300
3	The Perfect Murder	Mystery	150
4	Accidental Presidents	Biography	250
5	The Wicked King	Fiction	350

ID	UID	BID	ISSUED_D	RETURN_D	PERIOD	FINE	bname	genre	price
1	2	3	02-09-2019		10		The Perfect	Mystery	150

Sekarang, jika Anda ingin mengembalikan buku tersebut, maka Anda harus memilih opsi Pengembalian Buku.

## Return Book

Sekali, Anda mengklik Return Book, sebutkan **Issue ID** dan tanggal pengembalian (**the return date** seperti di bawah ini. Kemudian klik **Return**.

**Enter Details**

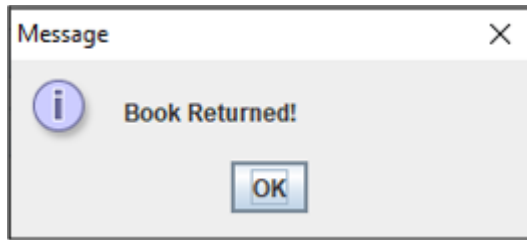
Issue ID(IID)

Return Date(DD-MM-YYYY)

kemudian, Anda melihat kotak pesan ditampilkan **fine**



Setelah itu, Anda kembali melihat kotak dialog, menampilkan pesan " **Book Returned** ". Lihat di bawah.



Sekarang, jika Anda mengklik **View Issued Books**, Anda akan melihat output di bawah ini:

IID	UID	BID	ISSUED_DATE	RETURN_DATE	PERIOD	FINE
1	2	3	02-09-2019	12-10-2019	10	300

Terakhir, jika Anda ingin menambahkan buku, Anda dapat menggunakan opsi Tambah Buku. **Add Book**

Click on the **Add Book button**, and mention the **book name, genre and price**. Then, click on the **Submit button**. Refer below.

Klik pada **Add Book button**, dan sebutkan nama **book name, genre and price**. Kemudian, klik tombol **Submit**. Lihat di bawah.

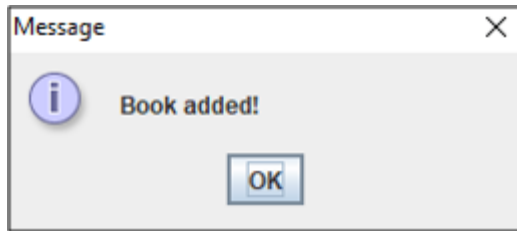
Enter Book Details

Book Name:

Genre:

Price:

Anda akan melihat kotak dialog yang menampilkan pesan di bawah ini:



Selain itu, Anda juga dapat melihat buku yang ditambahkan di bagian **View Books** seperti di bawah ini:

A window titled "Books Available" with a standard Windows title bar (minimize, maximize, close buttons). The window contains a table with four columns: BID, BNAME, GENRE, and PRICE. The table lists six books with their respective IDs, titles, genres, and prices. Below the table is a large, empty grey rectangular area.

BID	BNAME	GENRE	PRICE
1	War and Peace	Mystery	200
2	The Guest Book	Fiction	300
3	The Perfect Murder	Mystery	150
4	Accidental Presidents	Biography	250
5	The Wicked King	Fiction	350
6	Murder on the Orient Express	Thriller	1000

Ini membawa kita ke bagian Tutorial desain manajemen Perpustakaan dengan Bahasa Pemrograman Java. Saya harap penjelasan ini informatif dan menambah nilai pengetahuan Anda.