

**UJIAN AKHIR SEMESTER
PEMROGRAMAN BERORIENTASI OBJEK
SISTEM ADMINISTRASI PEMINJAMAN BUKU
“RAKBUKU”**



Disusun oleh:

**Diaz Amaliana
16/395389/TK/44681**

**PROGRAM STUDI TEKNOLOGI INFORMASI
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
2020**

DAFTAR ISI

A. DESKRIPSI APLIKASI	3
B. IDENTIFIKASI KEBUTUHAN SISTEM	3
1) Kebutuhan Fungsional	3
2) Kebutuhan Non-Fungsional	3
C. DESAIN MODEL	4
1) Rancangan <i>Use Case Diagram</i>	4
2) Rancangan <i>Class Diagram</i>	5
3) Rancangan <i>User Interface (UI)</i>	6
D. IMPLEMENTASI	11
1) Library Java Swing	11
2) Library JDBC MySql	11
3) Implementasi Kode	12
E. TESTING APLIKASI	13
F. SOURCE CODE	17
1) home.java	17
2) home2.java	26
3) login.java	32
4) user.java	35
5) about.java	37
6) add_buku.java	38
7) add_mahasiswa.java	41
8) add_peminjaman.java	43
9) edit_buku.java	48
10) edit_mahasiswa.java	51
11) edit_peminjaman.java	53

A. DESKRIPSI APLIKASI

Perpustakaan merupakan salah satu fasilitas lembaga pendidikan dan memegang peranan penting. Peran perpustakaan di kampus diibaratkan gudang ilmu, karena perpustakaan bisa memenuhi kebutuhan pengetahuan, berita, informasi, dan referensi untuk mahasiswa. Koleksi dokumen yang dimilikinya harus menjalankan manajemen yang terorganisir untuk memberikan layanan yang memuaskan kepada anggota perpustakaan (pustakawan).

Aplikasi perpustakaan yang akan dikembangkan adalah pemrograman sistem administrasi peminjaman buku sederhana berbasis GUI. Aplikasi menggunakan *database* sebagai tempat penyimpanan data, sehingga data yang di *input* dapat disimpan dan tertata dengan rapi. Aplikasi yang dinamakan “RakBuku” ini dikembangkan berdasarkan fungsi-fungsi yang disesuaikan dengan kegiatan-kegiatan yang ada di perpustakaan DTETI FT UGM.

B. IDENTIFIKASI KEBUTUHAN SISTEM

Dalam pengembangan aplikasi “RakBuku” analisa kebutuhan sistem dibagi menjadi dua golongan :

1) Kebutuhan Fungsional

Analisis kebutuhan fungsional mencakup analisis fungsi-fungsi yang dibutuhkan dalam pengembangan sistem. Fungsi minimal yang dibutuhkan dalam pengembangan aplikasi “RakBuku” sebagai berikut:

- a. Fungsi pendaftaran dan pengelolaan anggota perpustakaan.
- b. Fungsi pengelolaan dan pencarian data koleksi buku perpustakaan
- c. Fungsi pengelolaan transaksi peminjaman dan pengembalian buku

2) Kebutuhan Non-Fungsional

Kebutuhan teknis yang mencakup kebutuhan *hardware* serta *software* dalam proses pengembangan aplikasi adalah sebagai berikut:

- a. Laptop atau PC desktop yang memiliki sistem operasi Windows 7/10.
- b. IDE Netbeans 8.2
- c. XAMPP
- d. *Database MySQL*

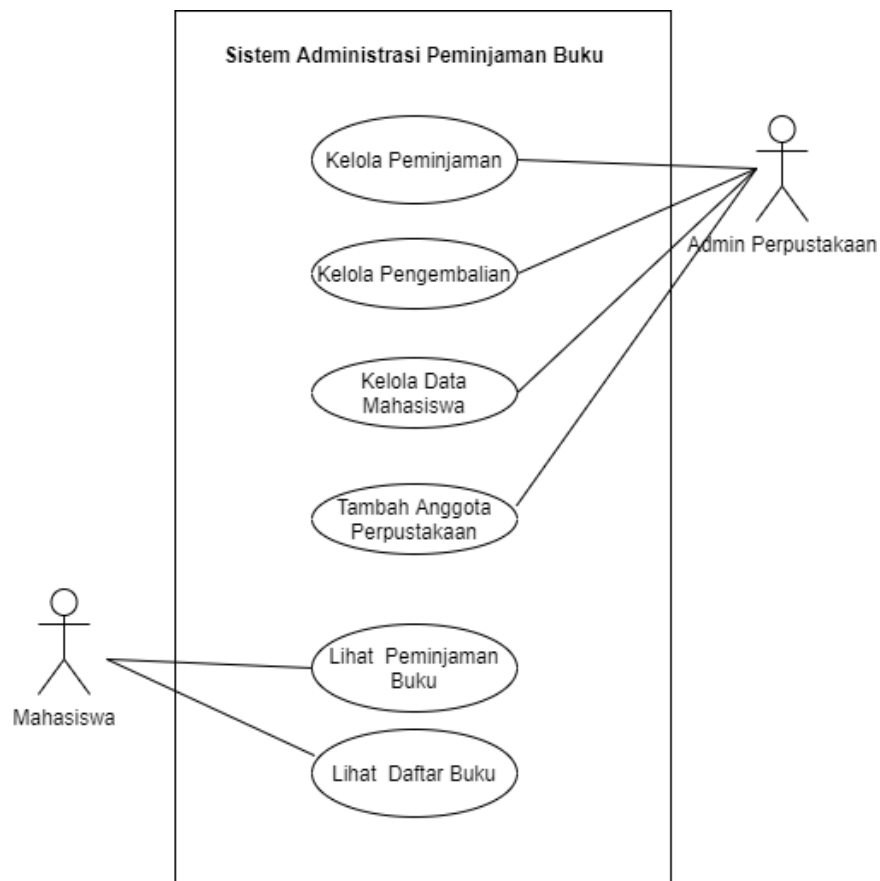
Dalam pengembangan aplikasi ini pengguna yang akan menggunakan aplikasi “RakBuku” berbasis GUI ini dibagi menjadi dua, yakni admin perpustakaan (admin), mahasiswa(*user*). Kedudukan admin lebih tinggi dari pada *user* karena admin sebagai

pengolah data-data sedangkan *user* hanya sebagai pengguna akhir dari sistem informasi perpustakaan yang akan dikembangkan. oleh karena itu admin berhak atas hak akses yang lebih luas daripada *user*.

C. DESAIN MODEL

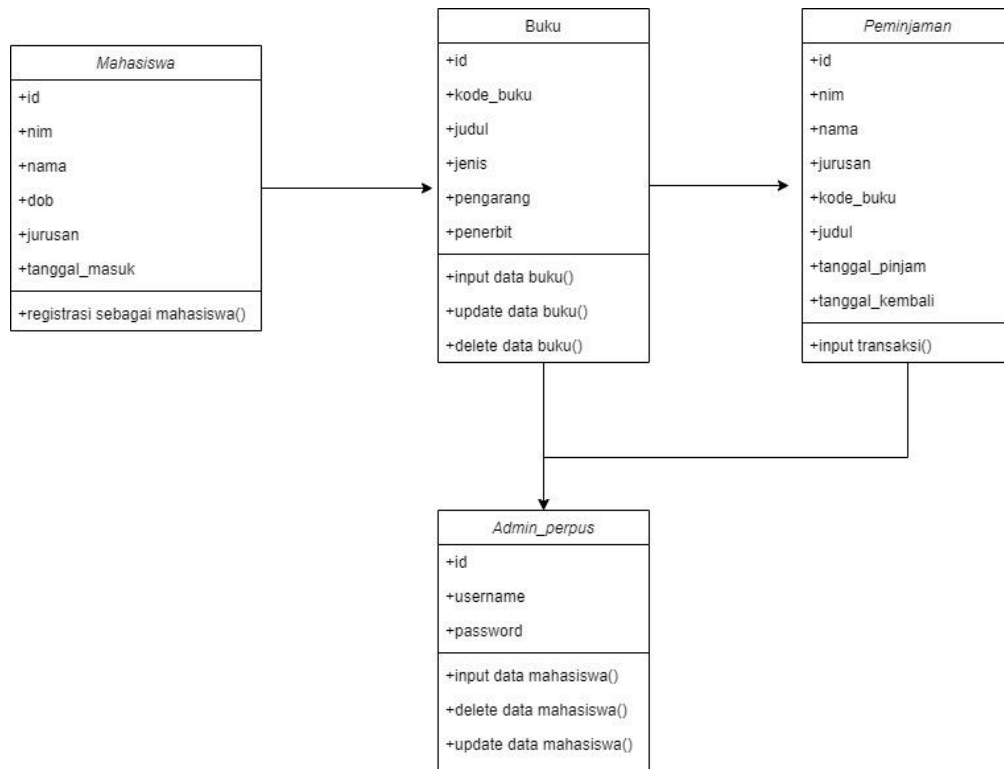
1) Rancangan *Use Case Diagram*

Gambar 1 menjelaskan bahwa aktor yang ada pada *use case* diagram memiliki dua aktor, yaitu terdiri dari Admin Perpustakaan dan Mahasiswa. Admin berhak untuk melakukan penginputan dan pengelolaan anggota, buku, dan peminjaman. Mahasiswa hanya berhak mengakses menu daftar peminjaman buku dan daftar buku.



Gambar 1 Use Case Diagram RakBuku

2) Rancangan *Class Diagram*



Gambar 2 Class Diagram RakBuku

Alur :

- Mahasiswa harus mendaftar sebagai anggota perpustakaan terlebih dahulu untuk meminjam buku. Pendaftaran sebagai anggota perpustakaan dilakukan oleh Admin_Perpustakaan.
- Kemudian Mahasiswa mempunyai relasi composition dengan class buku. Maksudnya ialah mahasiswa mencari buku mana yang akan di pinjam.
- Lalu dari class buku ke class Admin_Perpustakaan menggunakan relasi association karena untuk peminjaman buku ataupun pengembalian buku.
- Dari class buku ke peminjaman ada relasi composition. Karena harus adanya transaksi peminjaman tentang buku apa yang akan di pinjamkan.
- Kemudian dari class peminjaman ke class petugas perpustakaan menggunakan relasi composition, karena Mahasiswa meminjam/bertransaksi dengan Admin_Perpustakaan.

3) Rancangan *User Interface* (UI)

Design Preview [login]

RakBuku Hari, Tanggal Bulan Tahun Jam:Menit:Detik

Administrator
RakBuku

MASUK

Bukan Admin?

Gambar 3 UI Halaman Login

Design Preview [home]

RakBuku Now

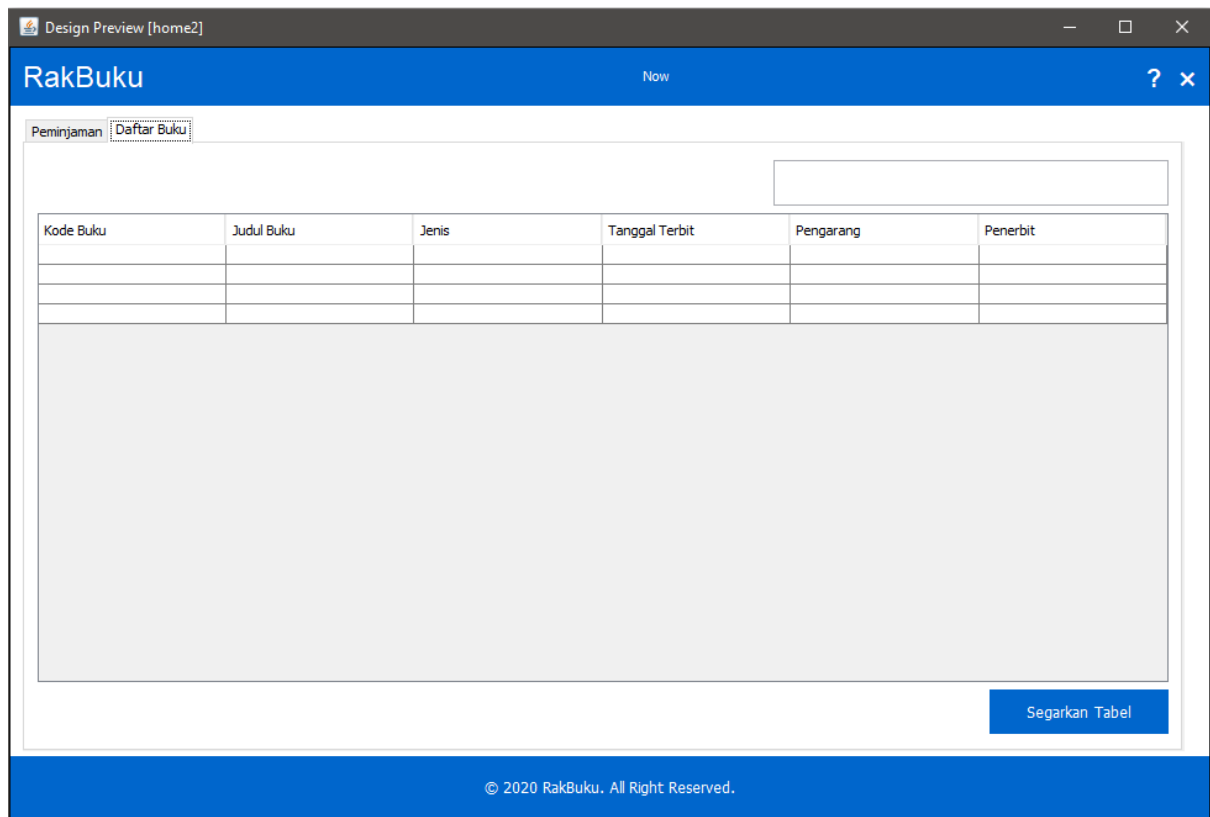
Peminjaman Kelola Buku Kelola Mahasiswa Pengaturan Admin

NIM	Nama	Jurusan	Kode Buku	Judul Buku	Tanggal Pinjam	Tanggal Kembali

Tambah Data Ubah Data Hapus Data Segarkan Tabel

© 2020 RakBuku. All Right Reserved.

Gambar 4 UI Halaman Utama untuk Admin (Home.java)



Gambar 5 Halaman Utama untuk Mahasiswa (Home2.java)

Cari Mahasiswa...

Cari Buku...

Cari Mahasiswa

NIM	Nama	Jurusan
81	Diaz Amaliana	S1 Teknologi Infor...
83	Estri Puji Lestari	S1 Teknologi Infor...
44694	Lely Monalisa	S1 Teknik Elektro
44467	Faishal Reza	S1 Teknik Elektro

Data Mahasiswa

NIM

Nama

Jurusan

Data Buku

Kode Buku

Jenis Buku

Buku Teks

Judul Buku

Tanggal Peminjaman (YYYY-MM-DD)

2020-12-02

Tanggal Pengembalian (YYYY-MM-DD)

2020-12-09

Kode Buku	Judul Buku	Tanggal Terbit
24MF0J6T	Pemograman Ber...	2020-11-26
10T67PKK	Pengembangan Apl...	2018-11-26
D04JFVY3	Pengantar Teknol...	2018-11-26
50MCIOKZ	Teknik Referensi ...	2014-12-01

Simpan

Gambar 6 UI Halaman add_peminjaman.java

Cari Mahasiswa...

Cari Buku...

Data Mahasiswa

NIM

44694

Nama

Lely Monalisa

Jurusan

S1 Teknik Elektro

Data Buku

Kode Buku

24MF0J6T

Jenis Buku

Ensiklopedia

Judul Buku

Pemograman Berorientasi Objek dengan Java

Tanggal Peminjaman (YYYY-MM-DD)

2020-12-02

Tanggal Kembali (YYYY-MM-DD)

2020-12-09

NIM	Nama	Jurusan
44681	Diaz Amaliana	S1 Teknologi Infor...
44683	Estri Puji Lestari	S1 Teknologi Infor...
44694	Lely Monalisa	S1 Teknik Elektro
44467	Faishal Reza	S1 Teknik Elektro

Kode Buku	Judul Buku	Tanggal Terbit
24MF0J6T	Pemograman Ber...	2020-11-26
10T67PKK	Pengembangan Apl...	2018-11-26
D04JFVY3	Pengantar Teknol...	2018-11-26
50MCIOKZ	Teknik Referensi ...	2014-12-01

Reset

Simpan

Gambar 8 UI Halaman edit_peminjaman.java

Kode Buku ?

LOROOAVM

Judul Buku

Jenis Buku

Buku Teks

Pengarang

Tanggal Terbit (YYYY-MM-DD)

2020-12-02

Penerbit

Simpan

Gambar 7 UI Halaman add_buku.java

Kode Buku: D04JFVY3

Pengarang: M Suryanto

Judul Buku: Pengantar Teknologi Informasi

Tanggal Terbit (YYYY-MM-DD): 2018-11-26

Jenis Buku: Buku Teks

Penerbit: Gamapress

Buttons: Reset, Simpan

Gambar 9 UI Halaman edit_buku.java

NIM:

Nama Lengkap:

Jurusan: S1 Teknik Elektro

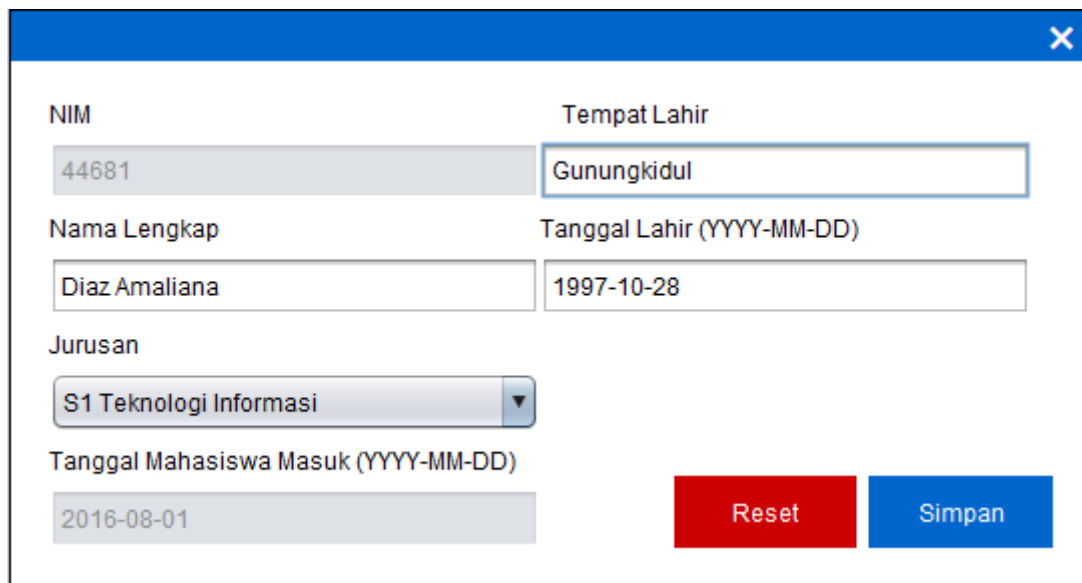
Tempat Lahir:

Tanggal Lahir (YYYY-MM-DD):

Tanggal Mahasiswa Masuk (YYYY-MM-DD):

Button: Simpan

Gambar 10 UI Halaman add_mahasiswa.java

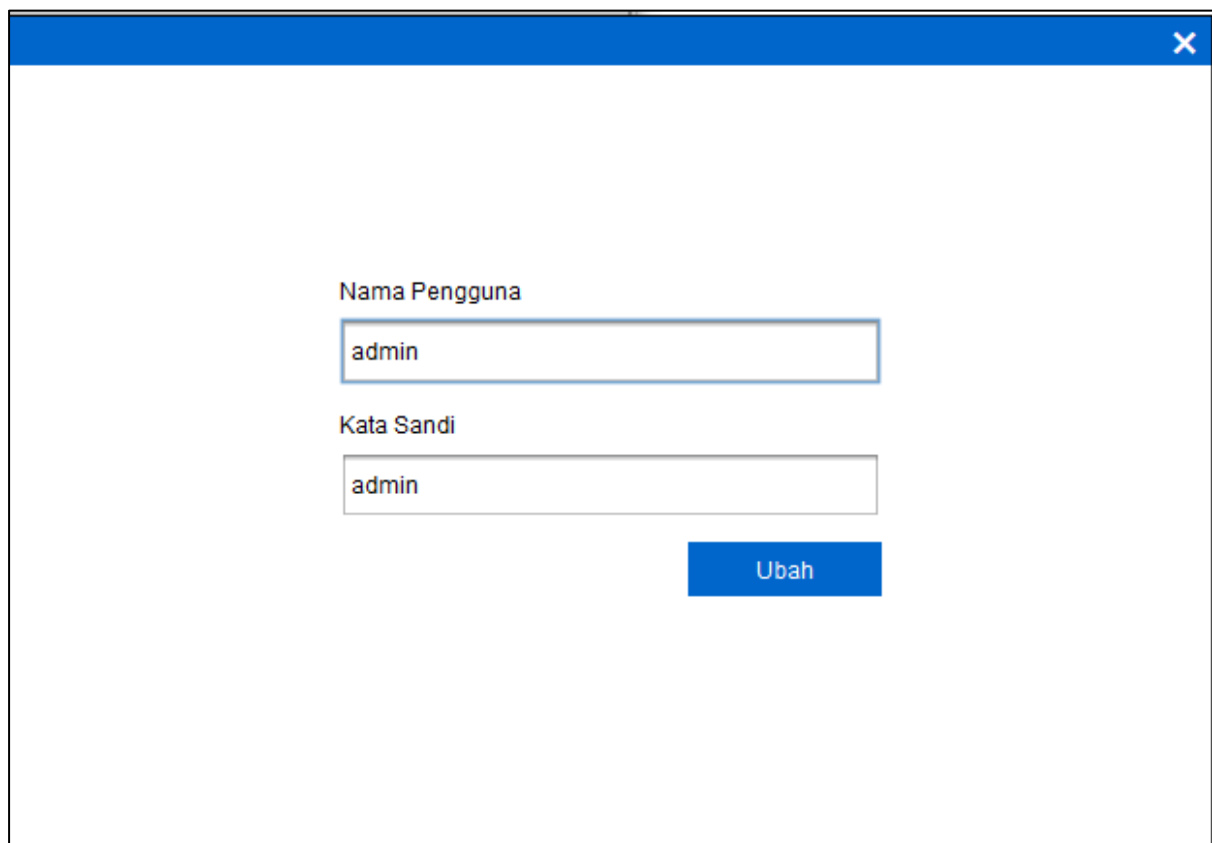


A screenshot of a Java Swing window titled "edit_mahasiswa.java". The window has a blue title bar with a close button (X) in the top right corner. The form contains several input fields and two buttons. The fields are arranged in a grid-like fashion. The "NIM" field is a text box with the value "44681". The "Tempat Lahir" field is a text box with the value "Gunungkidul". The "Nama Lengkap" field is a text box with the value "Diaz Amaliana". The "Tanggal Lahir (YYYY-MM-DD)" field is a text box with the value "1997-10-28". The "Jurusan" field is a dropdown menu with the selected value "S1 Teknologi Informasi". The "Tanggal Mahasiswa Masuk (YYYY-MM-DD)" field is a text box with the value "2016-08-01". At the bottom right of the form are two buttons: a red "Reset" button and a blue "Simpan" button.

NIM	Tempat Lahir
44681	Gunungkidul
Nama Lengkap	Tanggal Lahir (YYYY-MM-DD)
Diaz Amaliana	1997-10-28
Jurusan	
S1 Teknologi Informasi	
Tanggal Mahasiswa Masuk (YYYY-MM-DD)	
2016-08-01	

Reset Simpan

Gambar 11 UI Halaman edit_mahasiswa.java

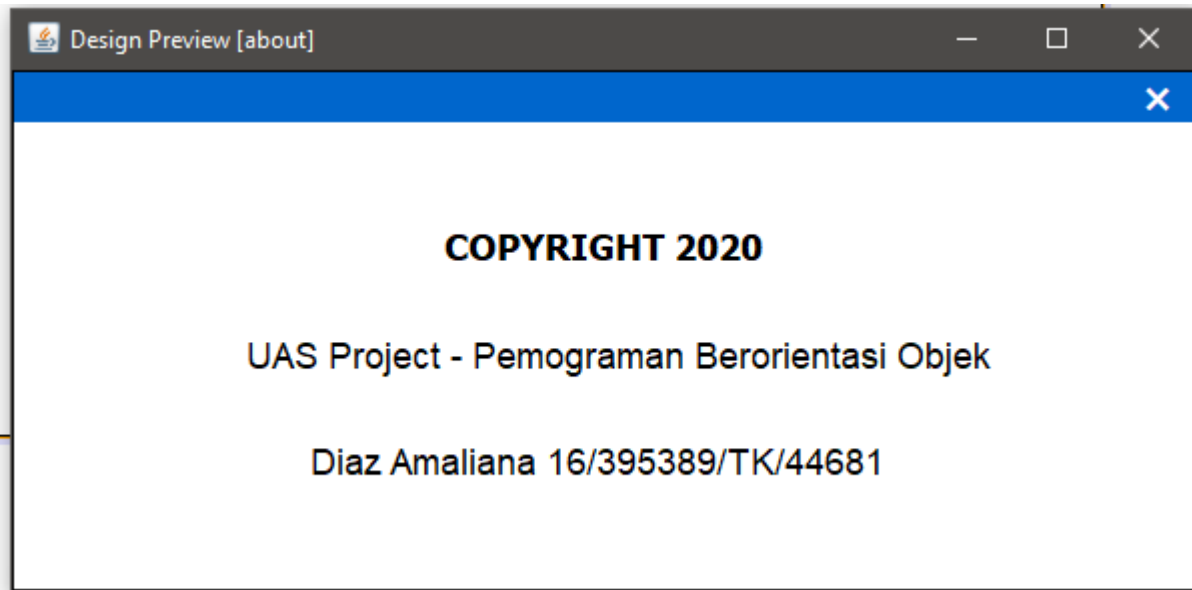


A screenshot of a Java Swing window for an admin password change. The window has a blue title bar with a close button (X) in the top right corner. The form contains two text input fields and one button. The first field is labeled "Nama Pengguna" and contains the text "admin". The second field is labeled "Kata Sandi" and contains the text "admin". Below the second field is a blue button labeled "Ubah".

Nama Pengguna
admin
Kata Sandi
admin

Ubah

Gambar 12 UI Halaman untuk admin mengganti password(user.java)



Gambar 13 UI Halaman about.java

D. IMPLEMENTASI

1) Library Java Swing

Swing adalah API (*Application Programming Interface*) yang digunakan oleh Java untuk membuat GUI (*Graphical User Interface*) yang menarik. Dengan Swing, kita dapat membuat tampilan aplikasi hanya dengan *drag and drop* komponen Swing dari panel. IDE Netbeans sudah menggunakan library *javax.swing* sehingga untuk menggunakan komponen - komponen Swing pada Netbeans sangat mudah.

2) Library JDBC MySql

Java menyediakan API standar (*Application Programming Interface*) untuk mengembangkan aplikasi *database* yang disebut JDBC API. JDBC adalah Java API yang digunakan untuk memanipulasi database. Dengan menggunakan JDBC API, pengembang aplikasi dan applet Java dapat dengan mudah mengakses berbagai jenis *database* dari berbagai vendor database (seperti MySQL Server, SQL Server, Oracle, Sybase, dll.).

Agar program Java dapat berinteraksi dengan basis data MySQL, harus digunakan JDBC driver dari MySQL. MySQL JDBC Driver disebut juga dengan MySQL Connector/J. Setelah memasang *library JDBC MySql*, diperlukan integrasi untuk menghubungkan aplikasi dengan *database*. Implementasi kodenya sebagai berikut :

Tabel 1 Script Kode connection.java

```
package database;

import java.sql.*;
import javax.swing.*;

/**
 *
 * @author diazam
 */
public class connection {
    static Connection mysqlconfig;
    static Statement mysqlstm;
    public Connection configDB() {
        try {
            String url = "jdbc:mysql://localhost:3306/pbo";
            String user = "root";
            String pass = "";
            DriverManager.registerDriver(new com.mysql.jdbc.Driver());
            mysqlconfig = DriverManager.getConnection(url, user, pass);
            mysqlstm = mysqlconfig.createStatement();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Koneksi GAGAL!\n"+e);
        }
        return mysqlconfig;
    }
}
```

3) Implementasi Kode

Tabel 2 Implementasi Source Code Aplikasi

No.	Nama File	Keterangan
1	home.java	Implementasi kode untuk menampilkan halaman utama admin.
2	home2.java	Implementasi kode untuk menampilkan halaman utama mahasiswa.

3	login.java	Implementasi kode untuk menampilkan halaman <i>login</i> .
4	user.java	Implementasi kode untuk menampilkan halaman penggantian <i>password</i> admin.
5	about.java	Implementasi kode untuk menampilkan profil pembuat aplikasi.
6	add_buku.java	Implementasi kode untuk menampilkan halaman ketika admin menambahkan buku ke <i>database</i> .
7	add_mahasiswa.java	Implementasi kode untuk menampilkan halaman ketika admin mendaftarkan anggota (mahasiswa) ke <i>database</i> .
8	add_peminjaman.java	Implementasi kode untuk menampilkan halaman ketika admin menambahkan daftar peminjaman buku ke <i>database</i> .
9	edit_buku.java	Implementasi kode untuk menampilkan halaman ketika admin menyunting data buku yang tersimpan.
10	edit_mahasiswa.java	Implementasi kode untuk menampilkan halaman ketika admin menyunting data mahasiswa yang tersimpan.
11	edit_peminjaman.java	Implementasi kode untuk menampilkan halaman ketika admin menyunting data peminjaman yang tersimpan.

E. TESTING APLIKASI

NO.	Skenario	Hasil yang Diharapkan	Hasil yang Diujikan	Keterangan
1	Memasukkan password dan username admin pada halaman login	Halaman Home	Sesuai yang diharapkan.	Valid.
2	Mengakses tombol “Tambah Data” pada tab	Halaman Add_peminjaman	Sesuai yang diharapkan.	Valid.

	peminjaman Halaman Home			
3	Menambahkan data peminjaman buku sesuai dengan mengisi form	Pop up <i>message</i> “Data BERHASIL ditambahkan!”, data masuk ke dalam <i>database</i>	Sesuai yang diharapkan.	Valid.
4	Mengakses tombol “Ubah Data” pada tab peminjaman Halaman Home	Pop up <i>message</i> “Anda belum memilih data yang akan diubah!”	Sesuai yang diharapkan.	Valid.
5	Mengakses tombol “Hapus Data” pada tab peminjaman Halaman Home	Pop up <i>message</i> “Anda belum memilih data yang akan diubah!”	Sesuai yang diharapkan.	Valid.
6	Memilih data pada tab peminjaman Halaman Home, kemudian mengakses tombol “Ubah Data”	Halaman Edit_peminjaman	Sesuai yang diharapkan.	Valid.
7	Memilih data pada tab peminjaman Halaman Home, kemudian mengakses tombol “Hapus Data”	Pop up <i>message</i> konfirmasi penghapusan data, data dihapus dari <i>database</i>	Sesuai yang diharapkan.	Valid.
8	Mengakses tombol “Tambah Data” pada tab Kelola Buku Halaman Home	Halaman Add_buku	Sesuai yang diharapkan.	Valid.
9	Menambahkan data buku dengan mengisi form	Pop up <i>message</i> “Data BERHASIL ditambahkan!”, data masuk ke dalam <i>database</i>	Sesuai yang diharapkan.	Valid.
10	Mengakses tombol “Ubah Data” pada tab Kelola Buku Halaman Home	Pop up <i>message</i> “Anda belum memilih data yang akan diubah!”	Sesuai yang diharapkan.	Valid.

11	Mengakses tombol “Hapus Data” pada tab Kelola Buku Halaman Home	Pop up <i>message</i> “Anda belum memilih data yang akan diubah!”	Sesuai yang diharapkan.	Valid.
12	Memilih data pada tab Kelola Buku Halaman Home, kemudian mengakses tombol “Ubah Data”	Halaman Edit_buku	Sesuai yang diharapkan.	Valid.
13	Memilih data pada tab Kelola Buku Halaman Home, kemudian mengakses tombol “Hapus Data”	Pop up <i>message</i> konfirmasi penghapusan data, data dihapus dari <i>database</i>	Sesuai yang diharapkan.	Valid.
14	Mengakses tombol “Tambah Data” pada tab Kelola Mahasiswa Halaman Home	Halaman Add_mahasiswa	Sesuai yang diharapkan.	Valid.
15	Menambahkan data mahasiswa dengan mengisi form	Pop up <i>message</i> “Data BERHASIL ditambahkan!”, data masuk ke dalam <i>database</i>	Sesuai yang diharapkan.	Valid.
16	Mengakses tombol “Ubah Data” pada tab Kelola Mahasiswa Halaman Home	Pop up <i>message</i> “Anda belum memilih data yang akan diubah!”	Sesuai yang diharapkan.	Valid.
17	Mengakses tombol “Hapus Data” pada tab Kelola Mahasiswa Halaman Home	Pop up <i>message</i> “Anda belum memilih data yang akan diubah!”	Sesuai yang diharapkan.	Valid.
18	Memilih data pada tab Kelola Mahasiswa Halaman Home, kemudian mengakses tombol “Ubah Data”	Halaman Edit_mahasiswa	Sesuai yang diharapkan.	Valid.
19	Memilih data pada tab Kelola Mahasiswa Halaman Home, kemudian mengakses tombol “Hapus Data”	Pop up <i>message</i> konfirmasi penghapusan data, data dihapus dari <i>database</i>	Sesuai yang diharapkan.	Valid.

20	Mengakses tab Pengaturan admin untuk penggantian password	Halaman user, Pop up <i>message</i> “Data login berhasil diubah!”	Sesuai yang diharapkan.	Valid.
21	Mengakses tab Peminjaman kemudian melakukan pencarian data berdasarkan kata kunci yang dimasukkan.	Hanya menampilkan data yang sesuai dengan kata kunci	Sesuai yang diharapkan.	Valid.
22	Mengakses tab Kelola Buku kemudian melakukan pencarian data berdasarkan kata kunci yang dimasukkan.	Hanya menampilkan data yang sesuai dengan kata kunci	Sesuai yang diharapkan.	Valid.
23	Mengakses tab Kelola Mahasiswa kemudian melakukan pencarian data berdasarkan kata kunci yang dimasukkan.	Hanya menampilkan data yang sesuai dengan kata kunci	Sesuai yang diharapkan.	Valid.
24	Mengakses button link “Bukan Admin?” pada halaman login	Menampilkan Halaman Home2	Sesuai yang diharapkan.	Valid.
25	Mengakses icon “x”	Pop up <i>message</i> “Anda yakin ingin keluar dari program?”	Sesuai yang diharapkan.	Valid.
26	Mengakses icon “?”	Menampilkan halaman about	Sesuai yang diharapkan.	Valid.

F. SOURCE CODE

1) Home.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
import java.util.logging.*;
import javax.swing.*;
import javax.swing.table.*;
import pages.*;

/*
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
*/

/**
 *
 * @author User
 */
public class home extends javax.swing.JFrame {

    DefaultTableModel tbl1 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    DefaultTableModel tbl2 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    DefaultTableModel tbl3 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };

    /**
     * Creates new form home
     */
    public home() {
        super("RakBuku");
        Dimension screen=Toolkit.getDefaultToolkit().getScreenSize();
        this.setSize(screen.width,screen.height);
        setExtendedState(java.awt.Frame.MAXIMIZED_BOTH);
        setResizable(false);
        this.setUndecorated(true);

        initComponents();
        setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);

        timenow();
        copyright();

        tampil_peminjaman();
        tampil_buku();
        tampil_mhsiswa();
    }

    private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        int x = JOptionPane.showConfirmDialog(null, "Anda yakin ingin keluar dari program?", "Keluar",
        JOptionPane.YES_NO_OPTION, JOptionPane.ERROR_MESSAGE);
        if (x == JOptionPane.YES_OPTION) {
            System.exit(0);
            dispose();
        }
    }
}
```

```

    }
}

private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    int x = JOptionPane.showConfirmDialog(null, "Anda yakin ingin keluar dari program?", "Keluar",
JOptionPane.YES_NO_OPTION, JOptionPane.ERROR_MESSAGE);
    if (x == JOptionPane.YES_OPTION) {
        System.exit(0);
        dispose();
    }
}

private void search_peminjamanKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    search_peminjaman();
}

private void search_peminjamanKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        search_peminjaman();
    }
}

private void aboutMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    new pages.about(this, rootPaneCheckingEnabled).setVisible(true);
}

private void segarkan1_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    search_peminjaman.setText("");
    result_peminjaman.setText("");
    tampil_peminjaman();
}

private void hapus1_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int baris = tb_peminjaman.getSelectedRow();
    if (baris >= 0) {
        String nim = tbl1.getValueAt(baris, 0).toString();
        String kode = tbl1.getValueAt(baris, 3).toString();
        String pinjam = tbl1.getValueAt(baris, 5).toString();

        int x = JOptionPane.showConfirmDialog(null, "Data Peminjaman Buku dengan\n\nKode Buku:
"+kode+"\nNIM Mahasiswa: "+nim+"\nTanggal Peminjaman: "+pinjam+"\n\nnakan Anda hapus, yakin?",
"Konfirmasi", JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
        if (x == JOptionPane.YES_OPTION) {
            try {
                Connection con = new database.connection().configDB();
                String sql = "DELETE FROM peminjaman WHERE nim = '"+nim+"' AND kode = '"+kode+"'
AND pinjam = '"+pinjam+"'";
                java.sql.Statement st = con.createStatement();
                st.executeUpdate(sql);
                JOptionPane.showMessageDialog(null, "Data berhasil dihapus!", "Informasi",
JOptionPane.INFORMATION_MESSAGE);
                tampil_peminjaman();
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "Gagal menghapus data!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
            }
        }
    }
    } else {
        JOptionPane.showMessageDialog(null, "Anda belum memilih data yang akan dihapus!",
"Informasi", JOptionPane.INFORMATION_MESSAGE);
    }
}

private void ubah1_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int baris = tb_peminjaman.getSelectedRow();
    if (baris >= 0) {
        String nim = tbl1.getValueAt(baris, 0).toString();

```

```

String kode = tbl1.getValueAt(baris, 3).toString();
String pinjam = tbl1.getValueAt(baris, 5).toString();

try {
    Connection con = new database.connection().configDB();
    String sql = "SELECT * FROM peminjaman WHERE nim = '"+nim+"' AND kode = '"+kode+"' AND
pinjam = '"+pinjam+"'";
    java.sql.Statement st = con.createStatement();
    java.sql.ResultSet rs = st.executeQuery(sql);

    if (rs.next()) {
        String id = rs.getString("id");
        new pages.edit_peminjaman(this, rootPaneCheckingEnabled, id).setVisible(true);
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Gagal menyeleksi data!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
}
} else {
    JOptionPane.showMessageDialog(null, "Anda belum memilih data yang akan diubah!",
"Informasi", JOptionPane.INFORMATION_MESSAGE);
}
}

private void tambah1_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    new pages.add_peminjaman(this, rootPaneCheckingEnabled).setVisible(true);
}

private void search_bukuKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        search_buku();
    }
}

private void search_bukuKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    search_buku();
}

private void tambah2_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    new pages.add_buku(this, rootPaneCheckingEnabled).setVisible(true);
}

private void ubah2_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int baris = tb_buku.getSelectedRow();
    if (baris >= 0) {
        String kode = tbl2.getValueAt(baris, 0).toString();
        try {
            Connection con = new database.connection().configDB();
            String sql = "SELECT * FROM buku WHERE kode = '"+kode+"'";
            java.sql.Statement st = con.createStatement();
            java.sql.ResultSet rs = st.executeQuery(sql);

            if (rs.next()) {
                String id = rs.getString("id");
                new pages.edit_buku(this, rootPaneCheckingEnabled, id).setVisible(true);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Gagal menyeleksi data!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(null, "Anda belum memilih data yang akan diubah!",
"Informasi", JOptionPane.INFORMATION_MESSAGE);
    }
}

private void hapus2_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int baris = tb_buku.getSelectedRow();

```

```

        if (baris >= 0) {
            String kode = tbl2.getValueAt(baris, 0).toString();
            String judul = tbl2.getValueAt(baris, 1).toString();

            int x = JOptionPane.showConfirmDialog(null, "Data Buku dengan\n\nJudul: "+judul+"\nKode
Buku: "+kode+"\n\nAnda hapus, yakin?\nData peminjaman dengan BUKU ini juga akan ikut terhapus!",
"Konfirmasi", JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
            if (x == JOptionPane.YES_OPTION) {
                try {
                    Connection con = new database.connection().configDB();
                    String sql = "DELETE FROM buku WHERE kode = '"+kode+"'";
                    String sql2 = "DELETE FROM peminjaman WHERE kode = '"+kode+"' AND judul =
'"+judul+"'";
                    java.sql.Statement st = con.createStatement();
                    st.executeUpdate(sql);
                    st.executeUpdate(sql2);
                    JOptionPane.showMessageDialog(null, "Data berhasil dihapus!", "Informasi",
JOptionPane.INFORMATION_MESSAGE);
                    tampil_buku();
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null, "Gagal menghapus data!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
                }
            } else {
                JOptionPane.showMessageDialog(null, "Anda belum memilih data yang akan dihapus!",
"Informasi", JOptionPane.INFORMATION_MESSAGE);
            }
        }

        private void segarkan2_valueMouseClicked(java.awt.event.MouseEvent evt) {
            // TODO add your handling code here:
            search_buku.setText("");
            result_buku.setText("");
            tampil_buku();
        }

        private void search_mhsiswaKeyPressed(java.awt.event.KeyEvent evt) {
            // TODO add your handling code here:
            if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
                search_mhsiswa();
            }
        }

        private void search_mhsiswaKeyTyped(java.awt.event.KeyEvent evt) {
            // TODO add your handling code here:
            search_mhsiswa();
        }

        private void tambah3_valueMouseClicked(java.awt.event.MouseEvent evt) {
            // TODO add your handling code here:
            new pages.add_mahasiswa(this, rootPaneCheckingEnabled).setVisible(true);
        }

        private void ubah3_valueMouseClicked(java.awt.event.MouseEvent evt) {
            // TODO add your handling code here:
            int baris = tb_mhsiswa.getSelectedRow();
            if (baris >= 0) {
                String nim = tbl3.getValueAt(baris, 0).toString();
                try {
                    Connection con = new database.connection().configDB();
                    String sql = "SELECT * FROM mahasiswa WHERE nim = '"+nim+"'";
                    java.sql.Statement st = con.createStatement();
                    java.sql.ResultSet rs = st.executeQuery(sql);

                    if (rs.next()) {
                        String id = rs.getString("id");
                        new pages.edit_mahasiswa(this, rootPaneCheckingEnabled, id).setVisible(true);
                    }
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null, "Gagal menyeleksi data!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
                }
            } else {

```

```

        JOptionPane.showMessageDialog(null, "Anda belum memilih data yang akan diubah!",
        "Informasi", JOptionPane.INFORMATION_MESSAGE);
    }

    private void hapus3_valueMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        int baris = tb_mhsiswa.getSelectedRow();
        if (baris >= 0) {
            String nim = tbl3.getValueAt(baris, 0).toString();
            String nama = tbl3.getValueAt(baris, 1).toString();

            int x = JOptionPane.showConfirmDialog(null, "Data Mahasiswa dengan\n\nNIM: "+nim+"\nNama
            Lengkap: "+nama+"\n\nnakan Anda hapus, yakin?\nData peminjaman dengan SISWA ini juga akan ikut
            terhapus!", "Konfirmasi", JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
            if (x == JOptionPane.YES_OPTION) {
                try {
                    Connection con = new database.connection().configDB();
                    String sql = "DELETE FROM mahasiswa WHERE nim = '"+nim+"'";
                    String sql2 = "DELETE FROM peminjaman WHERE nim = '"+nim+"'";
                    java.sql.Statement st = con.createStatement();
                    st.executeUpdate(sql);
                    st.executeUpdate(sql2);
                    JOptionPane.showMessageDialog(null, "Data berhasil dihapus!", "Informasi",
                    JOptionPane.INFORMATION_MESSAGE);
                    tampil_mhsiswa();
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null, "Gagal menghapus data!", "Kesalahan",
                    JOptionPane.ERROR_MESSAGE);
                }
            } else {
                JOptionPane.showMessageDialog(null, "Anda belum memilih data yang akan dihapus!",
                "Informasi", JOptionPane.INFORMATION_MESSAGE);
            }
        }

        private void segarkan3_valueMouseClicked(java.awt.event.MouseEvent evt) {
            // TODO add your handling code here:
            search_mhsiswa.setText("");
            result_mhsiswa.setText("");
            tampil_mhsiswa();
        }

        private void userinfoMouseClicked(java.awt.event.MouseEvent evt) {
            // TODO add your handling code here:
        }

        private void userinfoComponentShown(java.awt.event.ComponentEvent evt) {
            // TODO add your handling code here:
            this.dispose();
            new user(this, rootPaneCheckingEnabled).setVisible(true);
        }

        /**
         * @param args the command line arguments
         */
        public static void main(String args[]) {
            /* Set the Nimbus look and feel */
            //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
            /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
             * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
             */
            try {
                for
                (javax.swing.UIManager.LookAndFeelInfo info
                : javax.swing.UIManager.getInstalledLookAndFeels()) {
                    if ("Nimbus".equals(info.getName())) {
                        javax.swing.UIManager.setLookAndFeel(info.getClassName());
                        break;
                    }
                }
            } catch (ClassNotFoundException ex) {

```

```

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new home().setVisible(true);
        }
    });
}

private void timenow() {
    // ActionListener untuk Keperluan Timer
    ActionListener taskPerformer = new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            String nol_jam = "";
            String nol_menit = "";
            String nol_detik = "";
            // Membuat Date
            java.util.Date dt = new java.util.Date();
            // Membuat date format
            java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat ("EEEE, dd MMMMM yyyy")
;

            // Mengambil nilai JAM, MENIT, dan DETIK Sekarang
            int nilai_jam = dt.getHours();
            int nilai_menit = dt.getMinutes();
            int nilai_detik = dt.getSeconds();
            // Jika nilai JAM lebih kecil dari 10 (hanya 1 digit)
            if (nilai_jam <= 9) {
                // Tambahkan "0" didepannya
                nol_jam = "0";
            }
            // Jika nilai MENIT lebih kecil dari 10 (hanya 1 digit)
            if (nilai_menit <= 9) {
                // Tambahkan "0" didepannya
                nol_menit = "0";
            }
            // Jika nilai DETIK lebih kecil dari 10 (hanya 1 digit)
            if (nilai_detik <= 9) {
                // Tambahkan "0" didepannya
                nol_detik = "0";
            }
            // Membuat String JAM, MENIT, DETIK
            String jam = nol_jam + Integer.toString(nilai_jam);
            String menit = nol_menit + Integer.toString(nilai_menit);
            String detik = nol_detik + Integer.toString(nilai_detik);
            // Menampilkan pada Layar
            timenow.setText(kal.format(dt)+"          "+jam + ":" + menit + ":" + detik+"
");
        }
    };
    // Timer
    new javax.swing.Timer(1000, taskPerformer).start();
}

private void copyright() {
    java.util.Date sekarang = new java.util.Date();
    java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat("yyyy") ;
    copyright.setText("© "+kal.format(sekarang)+" Tugas Akhir. All Right Reserved.");
}

```

```

    }

    private void tampil_peminjaman() {
        Object []baris = {"NIM", "Nama", "Jurusan", "Kode Buku", "Judul Buku", "Tanggal Peminjaman",
"Tanggal Pengembalian"};
        tbl1 = new DefaultTableModel(null, baris) {
            public boolean isCellEditable(int row, int column) {
                return false;
            }
        };
        tb_peminjaman.setModel(tbl1);
        try {
            Connection con = new database.connection().configDB();
            String sql = "SELECT * FROM peminjaman ORDER BY id ASC";
            java.sql.Statement stat = con.createStatement();
            java.sql.ResultSet hasil = stat.executeQuery(sql);

            while (hasil.next()) {
                String nim = hasil.getString("nim");
                String nama = hasil.getString("nama");
                String jurusan = hasil.getString("jurusan");
                String kode = hasil.getString("kode");
                String judul = hasil.getString("judul");
                String pinjam = hasil.getString("pinjam");
                String kembali = hasil.getString("kembali");

                String[] data = {nim, nama, jurusan, kode, judul, pinjam, kembali};
                tbl1.addRow(data);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
            System.exit(0);
            dispose();
        }
    }

    private void tampil_buku() {
        Object []baris = {"Kode Buku", "Judul Buku", "Jenis", "Tanggal Terbit", "Pengarang",
"Penerbit"};
        tbl2 = new DefaultTableModel(null, baris) {
            public boolean isCellEditable(int row, int column) {
                return false;
            }
        };
        tb_buku.setModel(tbl2);
        try {
            Connection con = new database.connection().configDB();
            String sql = "SELECT * FROM buku ORDER BY id ASC";
            java.sql.Statement stat = con.createStatement();
            java.sql.ResultSet hasil = stat.executeQuery(sql);

            while (hasil.next()) {
                String jenis = null;
                String kode = hasil.getString("kode");
                String judul = hasil.getString("judul");
                String tanggal = hasil.getString("tanggal");
                String pengarang = hasil.getString("pengarang");
                String penerbit = hasil.getString("penerbit");

                if (hasil.getString("jenis").equals("1")) {
                    jenis = "Buku Teks";
                } else if (hasil.getString("jenis").equals("2")) {
                    jenis = "Referensi";
                } else if (hasil.getString("jenis").equals("3")) {
                    jenis = "Laporan Kerja Praktik";
                } else if (hasil.getString("jenis").equals("4")) {
                    jenis = "Jurnal";
                } else if (hasil.getString("jenis").equals("5")) {
                    jenis = "Majalah";
                } else if (hasil.getString("jenis").equals("6")) {
                    jenis = "Tugas Akhir";
                }
            }
        }
    }

```

```

        String[] data = {kode, judul, jenis, tanggal, pengarang, penerbit};
        tbl2.addRow(data);
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
    System.exit(0);
    dispose();
}
}

private void tampil_mhsiswa() {
    Object []baris = {"NIM", "Nama Lengkap", "Tempat Lahir", "Tanggal Lahir", "Jurusan", "Tanggal
Masuk"};
    tbl3 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tb_mhsiswa.setModel(tbl3);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM mahasiswa ORDER BY id ASC";
        java.sql.Statement stat = con.createStatement();
        java.sql.ResultSet hasil = stat.executeQuery(sql);

        while (hasil.next()) {
            String nim = hasil.getString("nim");
            String nama = hasil.getString("nama");
            String tempat_lahir = hasil.getString("tempat_lahir");
            String tanggal_lahir = hasil.getString("tanggal_lahir");
            String jurusan = hasil.getString("jurusan");
            String tanggal = hasil.getString("tanggal");

            String[] data = {nim, nama, tempat_lahir, tanggal_lahir, jurusan, tanggal};
            tbl3.addRow(data);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        System.exit(0);
        dispose();
    }
}

private void search_peminjaman() {
    String query = search_peminjaman.getText();

    Object []baris = {"NIM", "Nama", "Jurusan", "Kode Buku", "Judul Buku", "Tanggal Peminjaman",
"Tanggal Pengembalian"};
    tbl1 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tb_peminjaman.setModel(tbl1);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM peminjaman WHERE BINARY "
            + "nim LIKE '%" + query + "%' "
            + "OR nama LIKE '%" + query + "%' "
            + "OR jurusan LIKE '%" + query + "%' "
            + "OR kode LIKE '%" + query + "%' "
            + "OR judul LIKE '%" + query + "%' "
            + "OR pinjam LIKE '%" + query + "%' "
            + "OR kembali LIKE '%" + query + "%' "
            + "ORDER BY id ASC";
        java.sql.Statement stmt = con.createStatement();
        java.sql.ResultSet hasil = stmt.executeQuery(sql);

        while (hasil.next()) {
            String nim = hasil.getString("nim");

```



```

        String nama = hasil.getString("nama");
        String jurusan = hasil.getString("jurusan");
        String kode = hasil.getString("kode");
        String judul = hasil.getString("judul");
        String pinjam = hasil.getString("pinjam");
        String kembali = hasil.getString("kembali");

        String[] data = {nim, nama, jurusan, kode, judul, pinjam, kembali};
        tbl1.addRow(data);
    }

    if (!query.equals("")) {
        result_peminjaman.setText("Hasil pencarian dari: '"+query+"'");
    } else {
        result_peminjaman.setText("");
    }
} catch (Exception e) {}
}

private void search_buku() {
    String query = search_buku.getText();

    Object []baris = {"Kode Buku", "Judul Buku", "Jenis", "Tanggal Terbit", "Pengarang",
"Penerbit"};
    tbl2 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tb_buku.setModel(tbl2);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM buku WHERE BINARY "
            + "kode LIKE '%" + query + "%' "
            + "OR judul LIKE '%" + query + "%' "
            + "OR tanggal LIKE '%" + query + "%' "
            + "OR pengarang LIKE '%" + query + "%' "
            + "OR penerbit LIKE '%" + query + "%' "
            + "ORDER BY id ASC";
        java.sql.Statement stmt = con.createStatement();
        java.sql.ResultSet hasil = stmt.executeQuery(sql);

        while (hasil.next()) {
            String jenis = null;
            String kode = hasil.getString("kode");
            String judul = hasil.getString("judul");
            String tanggal = hasil.getString("tanggal");
            String pengarang = hasil.getString("pengarang");
            String penerbit = hasil.getString("penerbit");

            if (hasil.getString("jenis").equals("1")) {
                jenis = "Buku Teks";
            } else if (hasil.getString("jenis").equals("2")) {
                jenis = "Referensi";
            } else if (hasil.getString("jenis").equals("3")) {
                jenis = "Laporan Kerja Praktik";
            } else if (hasil.getString("jenis").equals("4")) {
                jenis = "Jurnal";
            } else if (hasil.getString("jenis").equals("5")) {
                jenis = "Majalah";
            } else if (hasil.getString("jenis").equals("6")) {
                jenis = "Tugas Akhir";
            }

            String[] data = {kode, judul, jenis, tanggal, pengarang, penerbit};
            tbl2.addRow(data);
        }

        if (!query.equals("")) {
            result_buku.setText("Hasil pencarian dari: '"+query+"'");
        } else {
            result_buku.setText("");
        }
    }
}

```

```

    }
    } catch (Exception e) {}
}

private void search_mhsiswa() {
    String query = search_mhsiswa.getText();

    Object []baris = {"NIM", "Nama Lengkap", "Tempat Lahir", "Tanggal Lahir", "Jurusan", "Tanggal
Masuk"};
    tbl3 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tb_mhsiswa.setModel(tbl3);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM mahasiswa WHERE BINARY "
            + "nim LIKE '%" + query + "%' "
            + "OR nama LIKE '%" + query + "%' "
            + "OR tempat_lahir LIKE '%" + query + "%' "
            + "OR tanggal_lahir LIKE '%" + query + "%' "
            + "OR jurusan LIKE '%" + query + "%' "
            + "OR tanggal LIKE '%" + query + "%' "
            + "ORDER BY id ASC";
        java.sql.Statement stmt = con.createStatement();
        java.sql.ResultSet hasil = stmt.executeQuery(sql);

        while (hasil.next()) {
            String nim = hasil.getString("nim");
            String nama = hasil.getString("nama");
            String tempat_lahir = hasil.getString("tempat_lahir");
            String tanggal_lahir = hasil.getString("tanggal_lahir");
            String jurusan = hasil.getString("jurusan");
            String tanggal = hasil.getString("tanggal");

            String[] data = {nim, nama, tempat_lahir, tanggal_lahir, jurusan, tanggal};
            tbl3.addRow(data);
        }

        if (!query.equals("")) {
            result_mhsiswa.setText("Hasil pencarian dari: '" + query + "'");
        } else {
            result_mhsiswa.setText("");
        }
    } catch (Exception e) {}
}

/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/
...

```

2) Home2.java

```

',',',
public class home2 extends javax.swing.JFrame {

    DefaultTableModel tbl1 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    DefaultTableModel tbl2 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
}

```

```

    };
    DefaultTableModel tbl3 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
};

/**
 * Creates new form home
 */
public home2() {
    super("RakBuku");
    Dimension screen=Toolkit.getDefaultToolkit().getScreenSize();
    this.setSize(screen.width,screen.height);
    setExtendedState(java.awt.Frame.MAXIMIZED_BOTH);
    setResizable(false);
    this.setUndecorated(true);

    initComponents();
    setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);

    timenow();
    copyright();

    tampil_peminjaman();
    tampil_buku();
}
private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int x = JOptionPane.showConfirmDialog(null, "Anda yakin ingin keluar dari program?",
    "Keluar", JOptionPane.YES_NO_OPTION, JOptionPane.ERROR_MESSAGE);
    if (x == JOptionPane.YES_OPTION) {
        System.exit(0);
        dispose();
    }
}

private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    int x = JOptionPane.showConfirmDialog(null, "Anda yakin ingin keluar dari program?",
    "Keluar", JOptionPane.YES_NO_OPTION, JOptionPane.ERROR_MESSAGE);
    if (x == JOptionPane.YES_OPTION) {
        System.exit(0);
        dispose();
    }
}

private void aboutMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    new pages.about(this, rootPaneCheckingEnabled).setVisible(true);
}

private void segarkan2_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    search_buku.setText("");
    result_buku.setText("");
    tampil_buku();
}

private void search_bukuKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    search_buku();
}

private void search_bukuKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        search_buku();
    }
}

private void segarkan1_valueMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    search_peminjaman.setText("");
}

```

```

        result_peminjaman.setText("");
        tampil_peminjaman();
    }

    private void search_peminjamanKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        search_peminjaman();
    }

    private void search_peminjamanKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
            search_peminjaman();
        }
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //

```

```

        int nilai_jam = dt.getHours();
        int nilai_menit = dt.getMinutes();
        int nilai_detik = dt.getSeconds();
        // Jika nilai JAM lebih kecil dari 10 (hanya 1 digit)
        if (nilai_jam <= 9) {
            // Tambahkan "0" didepannya
            nol_jam = "0";
        }
        // Jika nilai MENIT lebih kecil dari 10 (hanya 1 digit)
        if (nilai_menit <= 9) {
            // Tambahkan "0" didepannya
            nol_menit = "0";
        }
        // Jika nilai DETIK lebih kecil dari 10 (hanya 1 digit)
        if (nilai_detik <= 9) {
            // Tambahkan "0" didepannya
            nol_detik = "0";
        }
        // Membuat String JAM, MENIT, DETIK
        String jam = nol_jam + Integer.toString(nilai_jam);
        String menit = nol_menit + Integer.toString(nilai_menit);
        String detik = nol_detik + Integer.toString(nilai_detik);
        // Menampilkan pada Layar
        timenow.setText(kal.format(dt)+"          "+jam + ":" + menit + ":" + detik+"
");
    }
};
// Timer
new javax.swing.Timer(1000, taskPerformer).start();
}

private void copyright() {
    java.util.Date sekarang = new java.util.Date();
    java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat("yyyy") ;
    copyright.setText("@ "+kal.format(sekarang)+" Tugas Akhir. All Right Reserved.");
}

private void tampil_peminjaman() {
    Object []baris = {"NIM", "Nama", "Jurusan", "Kode Buku", "Judul Buku", "Tanggal Peminjaman",
    "Tanggal Pengembalian"};
    tbl1 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tb_peminjaman.setModel(tbl1);
    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM peminjaman ORDER BY id ASC";
        java.sql.Statement stat = con.createStatement();
        java.sql.ResultSet hasil = stat.executeQuery(sql);

        while (hasil.next()) {
            String nim = hasil.getString("nim");
            String nama = hasil.getString("nama");
            String jurusan = hasil.getString("jurusan");
            String kode = hasil.getString("kode");
            String judul = hasil.getString("judul");
            String pinjam = hasil.getString("pinjam");
            String kembali = hasil.getString("kembali");

            String[] data = {nim, nama, jurusan, kode, judul, pinjam, kembali};
            tbl1.addRow(data);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
        JOptionPane.ERROR_MESSAGE);
        System.exit(0);
        dispose();
    }
}

private void tampil_buku() {

```

```

        Object []baris = {"Kode Buku", "Judul Buku", "Jenis", "Tanggal Terbit", "Pengarang",
        "Penerbit"};
        tbl2 = new DefaultTableModel(null, baris) {
            public boolean isCellEditable(int row, int column) {
                return false;
            }
        };
        tb_buku.setModel(tbl2);
        try {
            Connection con = new database.connection().configDB();
            String sql = "SELECT * FROM buku ORDER BY id ASC";
            java.sql.Statement stat = con.createStatement();
            java.sql.ResultSet hasil = stat.executeQuery(sql);

            while (hasil.next()) {
                String jenis = null;
                String kode = hasil.getString("kode");
                String judul = hasil.getString("judul");
                String tanggal = hasil.getString("tanggal");
                String pengarang = hasil.getString("pengarang");
                String penerbit = hasil.getString("penerbit");

                if (hasil.getString("jenis").equals("1")) {
                    jenis = "Buku Teks";
                } else if (hasil.getString("jenis").equals("2")) {
                    jenis = "Referensi";
                } else if (hasil.getString("jenis").equals("3")) {
                    jenis = "Laporan Kerja Praktik";
                } else if (hasil.getString("jenis").equals("4")) {
                    jenis = "Jurnal";
                } else if (hasil.getString("jenis").equals("5")) {
                    jenis = "Majalah";
                } else if (hasil.getString("jenis").equals("6")) {
                    jenis = "Tugas Akhir";
                }

                String[] data = {kode, judul, jenis, tanggal, pengarang, penerbit};
                tbl2.addRow(data);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
            JOptionPane.ERROR_MESSAGE);
            System.exit(0);
            dispose();
        }
    }

    private void search_peminjaman() {
        String query = search_peminjaman.getText();

        Object []baris = {"NIM", "Nama", "Jurusan", "Kode Buku", "Judul Buku", "Tanggal Peminjaman",
        "Tanggal Pengembalian"};
        tbl1 = new DefaultTableModel(null, baris) {
            public boolean isCellEditable(int row, int column) {
                return false;
            }
        };
        tb_peminjaman.setModel(tbl1);

        try {
            Connection con = new database.connection().configDB();
            String sql = "SELECT * FROM peminjaman WHERE BINARY "
                + "nim LIKE '%" + query + "%' "
                + "OR nama LIKE '%" + query + "%' "
                + "OR jurusan LIKE '%" + query + "%' "
                + "OR kode LIKE '%" + query + "%' "
                + "OR judul LIKE '%" + query + "%' "
                + "OR pinjam LIKE '%" + query + "%' "
                + "OR kembali LIKE '%" + query + "%' "
                + "ORDER BY id ASC";
            java.sql.Statement stmt = con.createStatement();
            java.sql.ResultSet hasil = stmt.executeQuery(sql);

```

```

        while (hasil.next()) {
            String nim = hasil.getString("nim");
            String nama = hasil.getString("nama");
            String jurusan = hasil.getString("jurusan");
            String kode = hasil.getString("kode");
            String judul = hasil.getString("judul");
            String pinjam = hasil.getString("pinjam");
            String kembali = hasil.getString("kembali");

            String[] data = {nim, nama, jurusan, kode, judul, pinjam, kembali};
            tbl1.addRow(data);
        }

        if (!query.equals("")) {
            result_peminjaman.setText("Hasil pencarian dari: '"+query+"'");
        } else {
            result_peminjaman.setText("");
        }
    } catch (Exception e) {}
}

private void search_buku() {
    String query = search_buku.getText();

    Object []baris = {"Kode Buku", "Judul Buku", "Jenis", "Tanggal Terbit", "Pengarang",
"Penerbit"};
    tbl2 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tb_buku.setModel(tbl2);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM buku WHERE BINARY "
            + "kode LIKE '%" + query + "%' "
            + "OR judul LIKE '%" + query + "%' "
            + "OR tanggal LIKE '%" + query + "%' "
            + "OR pengarang LIKE '%" + query + "%' "
            + "OR penerbit LIKE '%" + query + "%' "
            + "ORDER BY id ASC";
        java.sql.Statement stmt = con.createStatement();
        java.sql.ResultSet hasil = stmt.executeQuery(sql);

        while (hasil.next()) {
            String jenis = null;
            String kode = hasil.getString("kode");
            String judul = hasil.getString("judul");
            String tanggal = hasil.getString("tanggal");
            String pengarang = hasil.getString("pengarang");
            String penerbit = hasil.getString("penerbit");

            if (hasil.getString("jenis").equals("1")) {
                jenis = "Buku Teks";
            } else if (hasil.getString("jenis").equals("2")) {
                jenis = "Referensi";
            } else if (hasil.getString("jenis").equals("3")) {
                jenis = "Laporan Kerja Praktik";
            } else if (hasil.getString("jenis").equals("4")) {
                jenis = "Jurnal";
            } else if (hasil.getString("jenis").equals("5")) {
                jenis = "Majalah";
            } else if (hasil.getString("jenis").equals("6")) {
                jenis = "Tugas Akhir";
            }

            String[] data = {kode, judul, jenis, tanggal, pengarang, penerbit};
            tbl2.addRow(data);
        }

        if (!query.equals("")) {

```

```

        result_buku.setText("Hasil pencarian dari: '"+query+"'");
    } else {
        result_buku.setText("");
    }
} catch (Exception e) {}
}

/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/
. . .

```

3) Login.java

```

. . .
/**
 *
 * @author User
 */
public class login extends javax.swing.JFrame {

    private static Point point = new Point();

    /**
     * Creates new form login
     */
    public login() {
        super("RakBuku");
        setSize(642, 443);
        setResizable(false);
        setLocationRelativeTo(null);
        this.setUndecorated(true);

        this.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                point.x = e.getX();
                point.y = e.getY();
            }
        });
        this.addMouseMotionListener(new MouseMotionAdapter() {
            public void mouseDragged(MouseEvent e) {
                Point p = getLocation();
                setLocation(p.x + e.getX() - point.x, p.y + e.getY() - point.y);
            }
        });

        initComponents();
        setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);

        thisday();
        timenow();
    }
    private void submitMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        submit();
    }

    private void passwordKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
            submit();
        }
    }

    private void usernameKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
            submit();
        }
    }
}

```



```

private void usernameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    int x = JOptionPane.showConfirmDialog(null, "Anda yakin ingin keluar dari program?",
"Keluar", JOptionPane.YES_NO_OPTION, JOptionPane.ERROR_MESSAGE);
    if (x == JOptionPane.YES_OPTION) {
        System.exit(0);
        dispose();
    }
}

private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int x = JOptionPane.showConfirmDialog(null, "Anda yakin ingin keluar dari program?",
"Keluar", JOptionPane.YES_NO_OPTION, JOptionPane.ERROR_MESSAGE);
    if (x == JOptionPane.YES_OPTION) {
        System.exit(0);
        dispose();
    }
}

private void bukanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new home2().setVisible(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //

```

```

    }

    private void submit() {
        if (username.getText().equals("") || password.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "Nama pengguna dan Kata sandi tidak boleh kosong!",
            "Peringatan", JOptionPane.WARNING_MESSAGE);
        } else {
            try {
                String sql = "SELECT * FROM admin WHERE username = '"+username.getText()+"' AND
password = '"+password.getText()+"'";
                Connection con = new database.connection().configDB();
                java.sql.Statement stat = con.createStatement();
                java.sql.ResultSet rs = stat.executeQuery(sql);

                if (rs.next()) {
                    if (username.getText().equals(rs.getString("username"))      &&
password.getText().equals(rs.getString("password"))) {
                        home n = new home();
                        n.setVisible(true);
                        this.setVisible(false);
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "Nama pengguna atau Kata sandi yang Anda
masukkan salah.", "Kesalahan", JOptionPane.ERROR_MESSAGE);
                }
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, e.getMessage());
            }
        }
    }

    private void thisday() {
        java.util.Date sekarang = new java.util.Date();
        java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat("EEEE, dd MMMMM yyyy") ;
        thisday.setText(""+kal.format(sekarang));
    }

    private void timenow() {
        // ActionListener untuk Keperluan Timer
        ActionListener taskPerformer = new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                String nol_jam = "";
                String nol_menit = "";
                String nol_detik = "";
                // Membuat Date
                java.util.Date dt = new java.util.Date();
                // Mengambil nilai JAM, MENIT, dan DETIK Sekarang
                int nilai_jam = dt.getHours();
                int nilai_menit = dt.getMinutes();
                int nilai_detik = dt.getSeconds();
                // Jika nilai JAM lebih kecil dari 10 (hanya 1 digit)
                if (nilai_jam <= 9) {
                    // Tambahkan "0" didepannya
                    nol_jam = "0";
                }
                // Jika nilai MENIT lebih kecil dari 10 (hanya 1 digit)
                if (nilai_menit <= 9) {
                    // Tambahkan "0" didepannya
                    nol_menit = "0";
                }
                // Jika nilai DETIK lebih kecil dari 10 (hanya 1 digit)
                if (nilai_detik <= 9) {
                    // Tambahkan "0" didepannya
                    nol_detik = "0";
                }
                // Membuat String JAM, MENIT, DETIK
                String jam = nol_jam + Integer.toString(nilai_jam);
                String menit = nol_menit + Integer.toString(nilai_menit);
                String detik = nol_detik + Integer.toString(nilai_detik);
                // Menampilkan pada Layar
                timenow.setText(jam + ":" + menit + ":" + detik);
            }
        };
        // Timer
    }

```

```

        new javax.swing.Timer(1000, taskPerformer).start();

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    . . .

```

4) User.java

```

. . .
public class user extends javax.swing.JDialog {

    /**
     * Creates new form userinfo
     */
    public user(java.awt.Frame parent, boolean modal) {
        super(parent, modal);

        setSize(319, 235);
        setResizable(false);
        setLocationRelativeTo(null);
        this.setUndecorated(true);

        initComponents();

        username.setEditable(false);
        password.setEditable(false);
        userlawas.setVisible(false);
        passlawas.setVisible(false);
        tampil();
    }

    private void close_btn2KeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        home n = new home();
        this.setVisible(false);
        n.setVisible(true);
    }

    private void edit_saveMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        if (edit_save.getText().equals("Ubah")) {
            username.setEditable(true);
            password.setEditable(true);
            edit_save.setText("Simpan");
        } else if (edit_save.getText().equals("Simpan")) {
            String pengguna = userlawas.getText();
            String sandi = passlawas.getText();
            String user = username.getText();
            String pass = password.getText();

            try {
                Connection con = new database.connection().configDB();
                String sql = "UPDATE admin SET username = ?, password = ? WHERE username = '"+pengguna+"' AND password = '"+sandi+"'";
                java.sql.PreparedStatement stat = con.prepareStatement(sql);
                stat.setString(1, user);
                stat.setString(2, pass);
                stat.executeUpdate();

                JOptionPane.showMessageDialog(null, "Data login BERHASIL Diubah", "Informasi",
                JOptionPane.INFORMATION_MESSAGE);

                login n = new login();
                this.setVisible(false);
                n.setVisible(true);
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "Data login GAGAL Diubah", "Kesalahan",
                JOptionPane.ERROR_MESSAGE);

                home n = new home();

```

```

        this.setVisible(false);
        n.setVisible(true);
    }
}

private void usernameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())
        {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(user.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(user.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(user.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(user.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            user dialog = new user(new javax.swing.JFrame(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

private void tampil() {
    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM admin";
        java.sql.Statement stat = con.createStatement();
        java.sql.ResultSet hasil = stat.executeQuery(sql);

        if (hasil.next()) {
            username.setText(hasil.getString("username"));
            password.setText(hasil.getString("password"));
            userlawas.setText(hasil.getString("username"));
            passlawas.setText(hasil.getString("password"));
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
    }
}

```

```

        System.exit(0);
        dispose();
    }
}
/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/
...

```

5) About.java

```

...
public class about extends javax.swing.JDialog {

    private static Point point = new Point();

    /**
     * Creates new form about_
     */
    public about(java.awt.Frame parent, boolean modal) {
        super(parent, modal);

        setSize(456, 260);
        setResizable(false);
        setLocationRelativeTo(null);
        this.setUndecorated(true);

        this.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                point.x = e.getX();
                point.y = e.getY();
            }
        });
        this.addMouseMotionListener(new MouseMotionAdapter() {
            public void mouseDragged(MouseEvent e) {
                Point p = getLocation();
                setLocation(p.x + e.getX() - point.x, p.y + e.getY() - point.y);
            }
        });

        initComponents();
    }
    private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        this.setVisible(false);
    }

    private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        this.setVisible(false);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

```

```

        java.util.logging.Logger.getLogger(about.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(about.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(about.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(about.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            about dialog = new about(new javax.swing.JFrame(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}
/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/
...

```

6) Add_buku.java

```

...
public class add_buku extends javax.swing.JDialog {

    private static Point point = new Point();

    private char[] chars = "1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ".toCharArray();
    private StringBuilder stringBuilder = new StringBuilder();
    private Random random = new Random();
    private String output;

    /**
     * Creates new form add_buku
     */
    public add_buku(java.awt.Frame parent, boolean modal) {
        super(parent, modal);

        setSize(540, 225);
        setResizable(false);
        setLocationRelativeTo(null);
        this.setUndecorated(true);

        this.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                point.x = e.getX();
                point.y = e.getY();
            }
        });
        this.addMouseMotionListener(new MouseMotionAdapter() {
            public void mouseDragged(MouseEvent e) {
                Point p = getLocation();
                setLocation(p.x + e.getX() - point.x, p.y + e.getY() - point.y);
            }
        });
    }

    initComponents();
}

```

```

        thisdate();

        kode.setText(getRandom());
        kode.setEnabled(false);
    }
    private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        this.setVisible(false);
    }

    private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        this.setVisible(false);
    }

    private void saveMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        String getkode, getjudul, getpengarang, gettanggal, getpenerbit;

        getkode = kode.getText();
        getjudul = judul.getText();
        getpengarang = pengarang.getText();
        gettanggal = tanggal.getText();
        getpenerbit = penerbit.getText();
        int getjenis = jenis.getSelectedIndex()+1;

        if (empty(getjudul) || empty(gettanggal) || empty(getpengarang) || empty(getpenerbit)) {
            JOptionPane.showMessageDialog(null, "Data yang Anda masukkan belum lengkap!",
"Peringatan!", JOptionPane.WARNING_MESSAGE);
        } else {
            try {
                Connection con = new database.connection().configDB();
                String get = "SELECT * FROM buku WHERE judul = '"+getjudul+"'";
                java.sql.Statement st = con.createStatement();
                java.sql.ResultSet hasil = st.executeQuery(get);

                if (hasil.next()) {
                    JOptionPane.showMessageDialog(null, "Judul buku tidak boleh ada yang sama.",
"Kesalahan", JOptionPane.ERROR_MESSAGE);
                } else {
                    try {
                        String sql = "INSERT INTO buku VALUES (null, ?, ?, ?, ?, ?,?)";
                        java.sql.PreparedStatement stat =
(PreparedStatement)con.prepareStatement(sql);
                        stat.setString(1, getkode);
                        stat.setString(2, getjenis+"");
                        stat.setString(3, getjudul);
                        stat.setString(4, gettanggal);
                        stat.setString(5, getpengarang);
                        stat.setString(6, getpenerbit);
                        stat.executeUpdate();

                        this.setVisible(false);
                        JOptionPane.showMessageDialog(null, "Data BERHASIL Ditambahkan",
"Informasi", JOptionPane.INFORMATION_MESSAGE);
                    } catch (Exception e) {
                        JOptionPane.showMessageDialog(null, "GAGAL Menambahkan Data",
"Kesalahan", JOptionPane.ERROR_MESSAGE);
                    }
                }
            } catch (Exception e) {}
        }
    }

    private void jenisActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void pengarangActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

/**
 * @param args the command line arguments

```

```

    */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
        *
        * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(add_buku.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(add_buku.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(add_buku.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(add_buku.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>

        /* Create and display the dialog */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                add_buku dialog = new add_buku(new javax.swing.JFrame(), true);
                dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                    @Override
                    public void windowClosing(java.awt.event.WindowEvent e) {
                        System.exit(0);
                    }
                });
                dialog.setVisible(true);
            }
        });

        private void thisdate() {
            java.util.Date sekarang = new java.util.Date();
            java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat("yyyy-MM-dd");
            tanggal.setText(""+kal.format(sekarang));
        }

        private String getRandom() {
            for (int length = 0; length < 8; length++) {
                Character character = chars[random.nextInt(chars.length)];
                stringBuilder.append(character);
            }
            output = stringBuilder.toString();
            stringBuilder.delete(0, 9);
            return output;
        }

        private static boolean empty(final String s) {
            return s == null || s.trim().isEmpty();
        }
    }
}

/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/

```



```
*/
. . .
```

7) Add_mahasiswa.java

```
...
public class add_mahasiswa extends javax.swing.JDialog {

    private static Point point = new Point();

    /**
     * Creates new form add_siswa
     */
    public add_mahasiswa(java.awt.Frame parent, boolean modal) {
        super(parent, modal);

        setSize(540, 281);
        setResizable(false);
        setLocationRelativeTo(null);
        this.setUndecorated(true);

        this.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                point.x = e.getX();
                point.y = e.getY();
            }
        });
        this.addMouseMotionListener(new MouseMotionAdapter() {
            public void mouseDragged(MouseEvent e) {
                Point p = getLocation();
                setLocation(p.x + e.getX() - point.x, p.y + e.getY() - point.y);
            }
        });

        initComponents();

        ((AbstractDocument)nim.getDocument()).setDocumentFilter(new DocumentFilter() {
            Pattern regex = Pattern.compile("\\d*");

            @Override
            public void replace(FilterBypass fb, int offset, int length, String text, AttributeSet attrs) throws
BadLocationException {
                Matcher matcher = regex.matcher(text);
                if (!matcher.matches()) {
                    return;
                }
                super.replace(fb, offset, length, text, attrs);
            }
        });
    }

    private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        this.setVisible(false);
    }

    private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        this.setVisible(false);
    }

    private void saveMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        String getnim, getnama, gettempat, getttl, getjurusan, gettanggal;
        getnim = nim.getText();
        getnama = nama.getText();
        gettempat = tempat_lahir.getText();
        getttl = tanggal_lahir.getText();
        getjurusan = (String)jurusan.getSelectedItem();
        gettanggal = tanggal.getText();

        if (empty(getnim) || empty(getnama) || empty(gettempat) || empty(getttl) || empty(gettanggal)) {
```

```

        JOptionPane.showMessageDialog(null, "Data yang Anda masukkan belum lengkap!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
    } else if (getnim.length() >= 6 || getnim.length() <= 4) {
        JOptionPane.showMessageDialog(null, "Panjang NIM harus 5 karakter!", "Peringatan!",
JOptionPane.WARNING_MESSAGE);
    } else {
        try {
            Connection con = new database.connection().configDB();
            String get = "SELECT * FROM mahasiswa WHERE nim = '"+getnim+"' OR nama = '"+getnama+"'";
            java.sql.Statement st = con.createStatement();
            java.sql.ResultSet hasil = st.executeQuery(get);

            if (hasil.next()) {
                JOptionPane.showMessageDialog(null, "Data seperti ini sudah pernah Anda masukkan
sebelumnya.", "Kesalahan", JOptionPane.ERROR_MESSAGE);
            } else {
                try{
                    String sql = "INSERT INTO mahasiswa VALUES (null, ?, ?, ?, ?, ?, ?)";
                    java.sql.PreparedStatement stat = (PreparedStatement)con.prepareStatement(sql);
                    stat.setString(1, getnim);
                    stat.setString(2, getnama);
                    stat.setString(3, gettempat);
                    stat.setString(4, getttl);
                    stat.setString(5, getjurusan);
                    stat.setString(6, gettanggal);
                    stat.executeUpdate();

                    this.setVisible(false);
                    JOptionPane.showMessageDialog(null, "Data BERHASIL Ditambahkan", "Informasi",
JOptionPane.INFORMATION_MESSAGE);
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null, "GAGAL Menambahkan Data", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
                }
            } catch (Exception e) {}
        }
    }

    private void nimKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void nimActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(add_mahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(add_mahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (IllegalAccessException ex) {

```

```

java.util.logging.Logger.getLogger(add_mahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(add_mahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            add_mahasiswa dialog = new add_mahasiswa(new javax.swing.JFrame(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

private static boolean empty(final String s) {
    return s == null || s.trim().isEmpty();
}

/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/
    . . .

```

8) Add_peminjaman.java

```

. . .
public class add_peminjaman extends javax.swing.JDialog {

    private static Point point = new Point();

    DefaultTableModel tbl1 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    DefaultTableModel tbl2 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };

    /**
     * Creates new form add_peminjaman
     */
    public add_peminjaman(java.awt.Frame parent, boolean modal) {
        super(parent, modal);

        setSize(760, 578);
        setResizable(false);
        setLocationRelativeTo(null);
        this.setUndecorated(true);

        this.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                point.x = e.getX();
                point.y = e.getY();
            }
        });
        this.addMouseMotionListener(new MouseMotionAdapter() {

```

```

        public void mouseDragged(MouseEvent e) {
            Point p = getLocation();
            setLocation(p.x + e.getX() - point.x, p.y + e.getY() - point.y);
        }
    });

    initComponents();

    nim.setEnabled(false);
    nama.setEnabled(false);
    jurusan.setEnabled(false);
    kode.setEnabled(false);
    judul.setEnabled(false);
    jenis.setEnabled(false);

    tampil_buku();
    tampil_mahasiswa();
    thisdate();
    afterdate();
}

private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
}

private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
}

private void tbl_mahasiswaMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    clicktbl_mahasiswa();
}

private void cari_mahasiswaKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        search_mahasiswa();
    }
}

private void cari_mahasiswaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    search_mahasiswa();
}

private void cari_bukuKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        search_book();
    }
}

private void cari_bukuKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    search_book();
}

private void saveMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    save();
}

private void tbl_bukuMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    clicktbl_buku();
}

private void jurusanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cari_mahasiswaActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    // TODO add your handling code here:
}

private void jenisActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for
            (javax.swing.UIManager.LookAndFeelInfo info
            :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(add_peminjaman.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(add_peminjaman.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(add_peminjaman.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(add_peminjaman.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            add_peminjaman dialog = new add_peminjaman(new javax.swing.JFrame(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

private void thisdate() {
    java.util.Date sekarang = new java.util.Date();
    java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat("yyyy-MM-dd") ;
    pinjam.setText(""+kal.format(sekarang));
    //kembali.setText("0000-00-00");
}

private void afterdate() {
    //java.util.Date seminggu = new java.util.Date();
    java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat("yyyy-MM-dd");
    Calendar c = Calendar.getInstance();
    c.add(Calendar.DATE, 7);
    kembali.setText(""+kal.format(c.getTime()));
}

```

```

private void tampil_buku() {
    Object []baris = {"Kode Buku", "Judul Buku", "Tanggal Terbit"};
    tbl1 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tbl_buku.setModel(tbl1);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM buku ORDER BY id ASC";
        java.sql.Statement st = con.createStatement();
        java.sql.ResultSet rs = st.executeQuery(sql);

        while (rs.next()) {
            String kode = rs.getString("kode");
            String judul = rs.getString("judul");
            String tanggal = rs.getString("tanggal");

            String[] data = {kode, judul, tanggal};
            tbl1.addRow(data);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        this.setVisible(false);
    }
}

private void tampil_mahasiswa() {
    Object []baris = {"NIM", "Nama", "Jurusan"};
    tbl2 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tbl_mahasiswa.setModel(tbl2);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM mahasiswa ORDER BY id ASC";
        java.sql.Statement st = con.createStatement();
        java.sql.ResultSet rs = st.executeQuery(sql);

        while (rs.next()) {
            String nim = rs.getString("nim");
            String nama = rs.getString("nama");
            String jurusan = rs.getString("jurusan");

            String[] data = {nim, nama, jurusan};
            tbl2.addRow(data);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        this.setVisible(false);
    }
}

private void search_book() {
    String query = cari_buku.getText();

    Object []baris = {"Kode Buku", "Judul Buku", "Tanggal Terbit"};
    tbl1 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tbl_buku.setModel(tbl1);

    try {
        Connection con = new database.connection().configDB();

```

```

        String sql = "SELECT * FROM buku WHERE BINARY "
            + "kode LIKE '%" + query + "%' "
            + "OR judul LIKE '%" + query + "%' "
            + "OR tanggal LIKE '%" + query + "%' "
            + "OR pengarang LIKE '%" + query + "%' "
            + "ORDER BY id ASC";
        java.sql.Statement stmt = con.createStatement();
        java.sql.ResultSet hasil = stmt.executeQuery(sql);

        while (hasil.next()) {
            String kode = hasil.getString("kode");
            String judul = hasil.getString("judul");
            String tanggal = hasil.getString("tanggal");

            String[] data = {kode, judul, tanggal};
            tbl1.addRow(data);
        }
    } catch (Exception e) {}
}

private void search_mahasiswa() {
    String query = cari_mahasiswa.getText();

    Object []baris = {"NIM", "Nama", "Jurusan"};
    tbl2 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tbl_mahasiswa.setModel(tbl2);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM mahasiswa WHERE BINARY "
            + "nim LIKE '%" + query + "%' "
            + "OR nama LIKE '%" + query + "%' "
            + "OR jurusan LIKE '%" + query + "%' "
            + "ORDER BY id ASC";
        java.sql.Statement stmt = con.createStatement();
        java.sql.ResultSet hasil = stmt.executeQuery(sql);

        while (hasil.next()) {
            String nim = hasil.getString("nim");
            String nama = hasil.getString("nama");
            String jurusan = hasil.getString("jurusan");

            String[] data = {nim, nama, jurusan};
            tbl2.addRow(data);
        }
    } catch (Exception e) {}
}

private void clicktbl_mahasiswa() {
    int baris = tbl_mahasiswa.getSelectedRow();

    nim.setText((String)tbl2.getValueAt(baris, 0));
    nama.setText((String)tbl2.getValueAt(baris, 1));
    jurusan.setText((String)tbl2.getValueAt(baris, 2));
}

private void clicktbl_buku() {
    int baris = tbl_buku.getSelectedRow();

    kode.setText((String)tbl1.getValueAt(baris, 0));
    judul.setText((String)tbl1.getValueAt(baris, 1));
    thisdate();
    afterdate();

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM buku WHERE kode = '" + (String)tbl1.getValueAt(baris, 0) + "'";
        java.sql.Statement stat = con.createStatement();
        java.sql.ResultSet rs = stat.executeQuery(sql);
    }
}

```

```

        if (rs.next()) {
            jenis.setSelectedIndex(rs.getInt("jenis")-1);
        }

    } catch (Exception e) {}
}

private void save() {
    String id, getnim, getnama, getjurusan, getkode, getjudul, getpinjam, getkembali;

    getnim = nim.getText();
    getnama = nama.getText();
    getjurusan = jurusan.getText();
    getkode = kode.getText();
    getjudul = judul.getText();
    getpinjam = pinjam.getText();
    getkembali = kembali.getText();

    if (empty(getnim) || empty(getkode) || empty(getpinjam) || empty(getkembali)) {
        JOptionPane.showMessageDialog(null, "Data yang Anda masukkan belum lengkap!", "Peringatan!",
JOptionPane.WARNING_MESSAGE);
    } else {
        try {
            Connection con = new database.connection().configDB();
            String get = "SELECT * FROM peminjaman WHERE nim = '"+getnim+"' AND kode = '"+getkode+"'";
            java.sql.Statement st = con.createStatement();
            java.sql.ResultSet hasil = st.executeQuery(get);

            if (hasil.next()) {
                JOptionPane.showMessageDialog(null, "Data seperti ini sudah pernah Anda masukkan
sebelumnya.", "Kesalahan", JOptionPane.ERROR_MESSAGE);
            } else {
                try {
                    String sql = "INSERT INTO peminjaman VALUES (null, ?, ?, ?, ?, ?, ?, ?)";
                    java.sql.PreparedStatement stat = (PreparedStatement)con.prepareStatement(sql);
                    stat.setString(1, getnim);
                    stat.setString(2, getnama);
                    stat.setString(3, getjurusan);
                    stat.setString(4, getkode);
                    stat.setString(5, getjudul);
                    stat.setString(6, getpinjam);
                    stat.setString(7, getkembali);
                    stat.executeUpdate();

                    this.setVisible(false);
                    JOptionPane.showMessageDialog(null, "Data BERHASIL Ditambahkan", "Informasi",
JOptionPane.INFORMATION_MESSAGE);
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null, "GAGAL Menambahkan Data", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
                }
            }
        } catch (Exception e) {}
    }
}

private static boolean empty(final String s) {
    return s == null || s.trim().isEmpty();
}

/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/

```

9) Edit_buku.java

```

...
public class edit_buku extends javax.swing.JDialog {

```



```

private static Point point = new Point();

/**
 * Creates new form edit_buku
 */
public edit_buku(java.awt.Frame parent, boolean modal, String id) {
    super(parent, modal);

    setSize(540, 225);
    setResizable(false);
    setLocationRelativeTo(null);
    this.setUndecorated(true);

    this.addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent e) {
            point.x = e.getX();
            point.y = e.getY();
        }
    });
    this.addMouseMotionListener(new MouseMotionAdapter() {
        public void mouseDragged(MouseEvent e) {
            Point p = getLocation();
            setLocation(p.x + e.getX() - point.x, p.y + e.getY() - point.y);
        }
    });

    initComponents();

    tampilin(id);
    idnya.setText(id);
    idnya.setVisible(false);
    kode.setEnabled(false);
    judul.setEnabled(false);
}

private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
}

private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
}

private void saveMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    String getpengarang, gettanggal;
    getpengarang = pengarang.getText();
    gettanggal = tanggal.getText();
    int getjenis = jenis.getSelectedIndex()+1;

    if (!empty(gettanggal)) {
        try {
            Connection con = new database.connection().configDB();
            String sql = "UPDATE buku SET jenis = ?, tanggal = ?, pengarang = ? WHERE id = "
            +idnya.getText()+"";
            java.sql.PreparedStatement stat = con.prepareStatement(sql);
            stat.setString(1, getjenis+"");
            stat.setString(2, gettanggal);
            stat.setString(3, getpengarang);
            stat.executeUpdate();

            this.setVisible(false);
            JOptionPane.showMessageDialog(null, "Data BERHASIL Diubah", "Informasi",
JOptionPane.INFORMATION_MESSAGE);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "GAGAL Mengubah Data", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(null, "Silahkan masukkan Tanggal Peminjaman!", "Peringatan!",
JOptionPane.WARNING_MESSAGE);
    }
}
}

```

```

private void resetMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    tampilin(idnya.getText());
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())
        {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(edit_buku.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(edit_buku.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(edit_buku.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(edit_buku.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
    }
    //</editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            edit_buku dialog = new edit_buku(new javax.swing.JFrame(), true, "0");
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

private void tampilin(String id) {
    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM buku WHERE id = '"+id+"'";
        java.sql.Statement stat = con.createStatement();
        java.sql.ResultSet hasil = stat.executeQuery(sql);

        if (hasil.next()) {
            kode.setText(hasil.getString("kode"));
            judul.setText(hasil.getString("judul"));
            pengarang.setText(hasil.getString("pengarang"));
            penerbit.setText(hasil.getString("penerbit"));
            tanggal.setText(hasil.getString("tanggal"));
            jenis.setSelectedIndex(hasil.getInt("jenis")-1);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal menampilkan Frame!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        this.setVisible(false);
    }
}

```

```

        private static boolean empty(final String s) {
            return s == null || s.trim().isEmpty();
        }
    /**
    This method is called from within the constructor to initialize the form.
    WARNING: Do NOT modify this code. The content of this method is always
    regenerated by the Form Editor.
    */
    . . .

```

10) Edit_mahasiswa.java

```

. . .
public class edit_mahasiswa extends javax.swing.JDialog {

    private static Point point = new Point();

    /**
     * Creates new form edit_siswa
     */
    public edit_mahasiswa(java.awt.Frame parent, boolean modal, String id) {
        super(parent, modal);

        setSize(540, 281);
        setResizable(false);
        setLocationRelativeTo(null);
        this.setUndecorated(true);

        this.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                point.x = e.getX();
                point.y = e.getY();
            }
        });
        this.addMouseMotionListener(new MouseMotionAdapter() {
            public void mouseDragged(MouseEvent e) {
                Point p = getLocation();
                setLocation(p.x + e.getX() - point.x, p.y + e.getY() - point.y);
            }
        });

        initComponents();

        ((AbstractDocument)nim.getDocument()).setDocumentFilter(new DocumentFilter() {
            Pattern regex = Pattern.compile("\\d*");

            @Override
            public void replace(DocumentFilter.FilterBypass fb, int offset, int length, String text, AttributeSet
attrs) throws BadLocationException {
                Matcher matcher = regex.matcher(text);
                if (!matcher.matches()) {
                    return;
                }
                super.replace(fb, offset, length, text, attrs);
            }
        });

        nim.setEnabled(false);
        idnya.setVisible(false);
        tanggal.setEnabled(false);
        idnya.setText(id);
        tampil(id);
    }
    private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        this.setVisible(false);
    }

    private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        this.setVisible(false);
    }

```

```

    }

    private void nimKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void saveMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        String getnim = nim.getText();
        String getnama = nama.getText();
        String gettempat = tempat_lahir.getText();
        String getttl = tanggal_lahir.getText();
        String getjurusan = (String)jurusan.getSelectedItem();
        String gettanggal = tanggal.getText();

        if (!empty(getnama) || !empty(gettempat) || !empty(getttl) || !empty(gettanggal)) {
            try {
                Connection con = new database.connection().configDB();
                String sql = "UPDATE mahasiswa SET nama = ?, tempat_lahir = ?, tanggal_lahir = ?, jurusan = ?,
tanggal = ? WHERE id = '"+idnya.getText()+"'";
                java.sql.PreparedStatement stat = con.prepareStatement(sql);
                stat.setString(1, getnama);
                stat.setString(2, gettempat);
                stat.setString(3, getttl);
                stat.setString(4, getjurusan);
                stat.setString(5, gettanggal);
                stat.executeUpdate();
                String sql2 = "UPDATE peminjaman SET nama = ? WHERE nim = '"+getnim+"'";
                java.sql.PreparedStatement stat2 = con.prepareStatement(sql2);
                stat2.setString(1, getnama);
                stat2.executeUpdate();

                this.setVisible(false);
                JOptionPane.showMessageDialog(null, "Data BERHASIL Diubah", "Informasi",
JOptionPane.INFORMATION_MESSAGE);
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "GAGAL Mengubah Data", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    private void resetMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        tampil(idnya.getText());
    }

    private void jurusanActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(edit_mahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(edit_mahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(edit_mahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(edit_mahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            edit_mahasiswa dialog = new edit_mahasiswa(new javax.swing.JFrame(), true, "0");
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

public void tampil(String id) {
    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM mahasiswa WHERE id = '"+id+"'";
        java.sql.Statement stat = con.createStatement();
        java.sql.ResultSet hasil = stat.executeQuery(sql);

        if (hasil.next()) {
            nim.setText(hasil.getString("nim"));
            nama.setText(hasil.getString("nama"));
            tempat_lahir.setText(hasil.getString("tempat_lahir"));
            tanggal_lahir.setText(hasil.getString("tanggal_lahir"));
            jurusan.setSelectedItem(hasil.getString("jurusan"));
            tanggal.setText(hasil.getString("tanggal"));
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal menampilkan Frame!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        this.setVisible(false);
    }
}

private static boolean empty(final String s) {
    return s == null || s.trim().isEmpty();
}
/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/
. . .

```

11) Edit_peminjaman.java

```

. . .
public class edit_peminjaman extends javax.swing.JDialog {

    private static Point point = new Point();

    DefaultTableModel tbl1 = new DefaultTableModel() {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
}

```

```

    }
};
DefaultTableModel tbl2 = new DefaultTableModel() {
    public boolean isCellEditable(int row, int column) {
        return false;
    }
};

/**
 * Creates new form edit_peminjaman
 */
public edit_peminjaman(java.awt.Frame parent, boolean modal, String id) {
    super(parent, modal);

    setSize(760, 578);
    setResizable(false);
    setLocationRelativeTo(null);
    this.setUndecorated(true);

    this.addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent e) {
            point.x = e.getX();
            point.y = e.getY();
        }
    });
    this.addMouseMotionListener(new MouseMotionAdapter() {
        public void mouseDragged(MouseEvent e) {
            Point p = getLocation();
            setLocation(p.x + e.getX() - point.x, p.y + e.getY() - point.y);
        }
    });

    initComponents();

    nim.setEnabled(false);
    nama.setEnabled(false);
    jurusan.setEnabled(false);
    kode.setEnabled(false);
    judul.setEnabled(false);
    jenis.setEnabled(false);
    id_peminjaman.setVisible(false);

    tampil_data(id);
    tampil_buku();
    tampil_mahasiswa();
}

private void cari_bukuKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    search_book();
}

private void cari_bukuKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        search_book();
    }
}

private void cari_mahasiswaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    search_mahasiswa();
}

private void cari_mahasiswaKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        search_mahasiswa();
    }
}

private void tbl_mahasiswaMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    clicktbl_mahasiswa();
}
}

```

```

private void close_btnKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
}

private void close_btnMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
}

private void resetMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    String id = id_peminjaman.getText();
    tampil_data(id);
}

private void tbl_bukuMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    clicktbl_buku();
}

private void saveMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    save();
}

private void kembaliActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    <editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())
        {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(edit_peminjaman.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(edit_peminjaman.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(edit_peminjaman.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(edit_peminjaman.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    </editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            edit_peminjaman dialog = new edit_peminjaman(new javax.swing.JFrame(), true, 0+"");
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {

```

```

        System.exit(0);
    }
    });
    dialog.setVisible(true);
}
});
}

private void tampil_data(String id) {
    id_peminjaman.setText(id);
    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM peminjaman WHERE id = '"+id+"'";
        java.sql.Statement st = con.createStatement();
        java.sql.ResultSet rs = st.executeQuery(sql);

        if (rs.next()) {
            nim.setText(rs.getString("nim"));
            nama.setText(rs.getString("nama"));
            jurusan.setText(rs.getString("jurusan"));
            kode.setText(rs.getString("kode"));
            judul.setText(rs.getString("judul"));
            pinjam.setText(rs.getString("pinjam"));
            kembali.setText(rs.getString("kembali"));
            try {
                String sql2 = "SELECT * FROM buku WHERE kode = '"+rs.getString("kode")+"'";
                java.sql.ResultSet hasil = st.executeQuery(sql2);

                if (hasil.next()) {
                    jenis.setSelectedIndex(hasil.getInt("jenis")-1);
                }
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "Gagal menampilkan Frame!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
                this.setVisible(false);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal menampilkan Frame!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        this.setVisible(false);
    }
}

private void tampil_buku() {
    Object []baris = {"Kode Buku", "Judul Buku", "Tanggal Terbit"};
    tbl1 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tbl_buku.setModel(tbl1);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM buku ORDER BY id ASC";
        java.sql.Statement st = con.createStatement();
        java.sql.ResultSet rs = st.executeQuery(sql);

        while (rs.next()) {
            String kode = rs.getString("kode");
            String judul = rs.getString("judul");
            String tanggal = rs.getString("tanggal");

            String[] data = {kode, judul, tanggal};
            tbl1.addRow(data);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        this.setVisible(false);
    }
}
}

```



```

private void tampil_mahasiswa() {
    Object []baris = {"NIM", "Nama", "Jurusan"};
    tbl2 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tbl_mahasiswa.setModel(tbl2);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM mahasiswa ORDER BY id ASC";
        java.sql.Statement st = con.createStatement();
        java.sql.ResultSet rs = st.executeQuery(sql);

        while (rs.next()) {
            String nim = rs.getString("nim");
            String nama = rs.getString("nama");
            String jurusan = rs.getString("jurusan");

            String[] data = {nim, nama, jurusan};
            tbl2.addRow(data);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal Menampilkan DATA!", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
        this.setVisible(false);
    }
}

private void search_book() {
    String query = cari_buku.getText();

    Object []baris = {"Kode Buku", "Judul Buku", "Tanggal Terbit"};
    tbl1 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tbl_buku.setModel(tbl1);

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM buku WHERE BINARY "
            + "kode LIKE '%" + query + "%' "
            + "OR judul LIKE '%" + query + "%' "
            + "OR tanggal LIKE '%" + query + "%' "
            + "OR pengarang LIKE '%" + query + "%' "
            + "OR penerbit LIKE '%" + query + "%' "
            + "ORDER BY id ASC";
        java.sql.Statement stmt = con.createStatement();
        java.sql.ResultSet hasil = stmt.executeQuery(sql);

        while (hasil.next()) {
            String kode = hasil.getString("kode");
            String judul = hasil.getString("judul");
            String tanggal = hasil.getString("tanggal");

            String[] data = {kode, judul, tanggal};
            tbl1.addRow(data);
        }
    } catch (Exception e) {}
}

private void search_mahasiswa() {
    String query = cari_mahasiswa.getText();

    Object []baris = {"NIM", "Nama", "Jurusan"};
    tbl2 = new DefaultTableModel(null, baris) {
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    tbl_mahasiswa.setModel(tbl2);
}

```

```

try {
    Connection con = new database.connection().configDB();
    String sql = "SELECT * FROM mahasiswa WHERE BINARY "
        + "nim LIKE '%" + query + "%' "
        + "OR nama LIKE '%" + query + "%' "
        + "OR jurusan LIKE '%" + query + "%' "
        + "ORDER BY id ASC";
    java.sql.Statement stmt = con.createStatement();
    java.sql.ResultSet hasil = stmt.executeQuery(sql);

    while (hasil.next()) {
        String nim = hasil.getString("nim");
        String nama = hasil.getString("nama");
        String jurusan = hasil.getString("jurusan");

        String[] data = {nim, nama, jurusan};
        tbl2.addRow(data);
    }
} catch (Exception e) {}
}

private void clicktbl_mahasiswa() {
    int baris = tbl_mahasiswa.getSelectedRow();

    nim.setText((String)tbl2.getValueAt(baris, 0));
    nama.setText((String)tbl2.getValueAt(baris, 1));
    jurusan.setText((String)tbl2.getValueAt(baris, 2));
}

private void clicktbl_buku() {
    int baris = tbl_buku.getSelectedRow();

    kode.setText((String)tbl1.getValueAt(baris, 0));
    judul.setText((String)tbl1.getValueAt(baris, 1));

    try {
        Connection con = new database.connection().configDB();
        String sql = "SELECT * FROM buku WHERE kode = '" + (String)tbl1.getValueAt(baris, 0) + "'";
        java.sql.Statement stat = con.createStatement();
        java.sql.ResultSet rs = stat.executeQuery(sql);

        if (rs.next()) {
            jenis.setSelectedIndex(rs.getInt("jenis")-1);
        }
    } catch (Exception e) {}
}

private void save() {
    String id, getnim, getnama, getjurusan, getkode, getjudul, getpinjam, getkembali;

    getnim = nim.getText();
    getnama = nama.getText();
    getjurusan = jurusan.getText();
    getkode = kode.getText();
    getjudul = judul.getText();
    getpinjam = pinjam.getText();
    getkembali = kembali.getText();

    id = id_peminjaman.getText();

    if (!empty(getpinjam)) {
        try {
            Connection con = new database.connection().configDB();
            String sql = "UPDATE peminjaman SET nim = ?, nama = ?, jurusan = ?, kode = ?, judul = ?, pinjam
= ?, kembali = ? WHERE id = '" + id + "'";
            java.sql.PreparedStatement stat = con.prepareStatement(sql);
            stat.setString(1, getnim);
            stat.setString(2, getnama);
            stat.setString(3, getjurusan);
            stat.setString(4, getkode);
            stat.setString(5, getjudul);
            stat.setString(6, getpinjam);

```

```

        stat.setString(7, getkembali);
        stat.executeUpdate();

        this.setVisible(false);
        JOptionPane.showMessageDialog(null, "Data BERHASIL Diubah", "Informasi",
JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "GAGAL Mengubah Data", "Kesalahan",
JOptionPane.ERROR_MESSAGE);
    }
    } else {
        JOptionPane.showMessageDialog(null, "Silahkan masukkan Tanggal Peminjaman!", "Peringatan!",
JOptionPane.WARNING_MESSAGE);
    }
}

private static boolean empty(final String s) {
    return s == null || s.trim().isEmpty();
}
/**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
*/
    . . .

```