

**PROYEK AKHIR MATA KULIAH
JAVA OBJECT ORIENTED PROGRAMMING 2
SEMESTER GENAP 2023-2024**

PERPUSTAKAAN DIGITAL



Disusun oleh :

- 1. Sahrul Maulidi**
- 2. Rendi Mah**
- 3. Mila Adelia**
- 4. Liza Andareta**
- 5. Rifki permana**
- 6. M Fikri Hidayatullah S**

Dosen Pengajar : Anwar Hilar, M.kom.

**FAKULTAS TEKNOLOGI INFORMASI DAN KOMPUTER
UNIVERSITAS HORIZON INDONESIA
KARAWANG
2024**

DAFTAR ISI

DAFTAR ISI	i
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	3
BAB II PEMBAHASAN PROYEK	5
2.1 Perpustakaan Digital	5
2.2 Tim Pengembang Proyek	6
2.3 Database Relational	6
2.4 Usecase Diagram	7
2.5 Activity Diagram	8
2.6 Program	15
BAB III	70
SIMPULAN DAN SARAN	70
3.1 Simpulan	70
3.2 Saran	70
DAFTAR PUSTAKA	72

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin maju, kebutuhan akan akses informasi yang cepat, mudah, dan efisien semakin meningkat. Salah satu sektor yang terpengaruh oleh perkembangan teknologi ini adalah perpustakaan. Perpustakaan tradisional yang berbasis fisik kini bertransformasi menjadi perpustakaan digital yang mampu menyediakan akses informasi dan literatur secara online.

Perpustakaan digital tidak hanya menawarkan kemudahan akses, tetapi juga memperluas jangkauan informasi yang dapat diakses oleh pengguna. Dengan adanya perpustakaan digital, pengguna dapat mengakses berbagai jenis bahan bacaan seperti buku, jurnal, artikel, dan dokumen lainnya dari berbagai belahan dunia tanpa harus terbatas oleh ruang dan waktu. Selain itu, perpustakaan digital memungkinkan pengguna untuk melakukan pencarian informasi dengan lebih cepat dan efisien melalui fitur-fitur pencarian yang canggih.

Mata kuliah Java Object Oriented Programming 2 pada semester genap 2023-2024 menekankan pada penerapan prinsip-prinsip pemrograman berorientasi objek (OOP) dalam pengembangan aplikasi. Sebagai proyek akhir, mahasiswa diharapkan mampu mengimplementasikan konsep-konsep OOP dalam bentuk aplikasi perpustakaan digital. Proyek ini bertujuan untuk memberikan pengalaman praktis kepada mahasiswa dalam membangun sistem yang kompleks dengan menggunakan bahasa pemrograman Java.

Melalui proyek ini, mahasiswa akan belajar bagaimana merancang dan mengembangkan sistem yang dapat mengelola koleksi digital, mengatur akses pengguna, serta menyediakan antarmuka yang intuitif bagi pengguna. Proyek ini juga akan menekankan pada pentingnya keamanan data dan pengelolaan hak akses dalam sistem perpustakaan digital.

Dengan mengembangkan perpustakaan digital, diharapkan mahasiswa tidak hanya memahami konsep-konsep OOP secara teori, tetapi juga mampu menerapkannya dalam konteks dunia nyata. Selain itu, proyek ini juga diharapkan dapat menjadi solusi praktis bagi institusi pendidikan dalam mengelola koleksi perpustakaan mereka secara lebih efektif dan efisien.

1.2 Rumusan Masalah

Dalam pengembangan proyek akhir mata kuliah Java Object Oriented Programming 2, terdapat beberapa masalah yang perlu diidentifikasi dan diselesaikan untuk mewujudkan aplikasi perpustakaan digital yang efektif dan efisien. Adapun rumusan masalah yang dapat diangkat dalam proyek ini adalah sebagai berikut:

1. Bagaimana merancang dan mengembangkan sistem perpustakaan digital yang mampu mengelola koleksi buku dan bahan bacaan lainnya secara efisien.
2. Bagaimana memastikan keamanan data dan pengelolaan hak akses dalam sistem perpustakaan digital.
3. Bagaimana merancang antarmuka pengguna (user interface) yang intuitif dan mudah digunakan.
4. Bagaimana mengintegrasikan berbagai fitur tambahan seperti peminjaman dan pengembalian buku secara online, serta notifikasi pengingat kepada pengguna.
5. Bagaimana memanfaatkan konsep-konsep pemrograman berorientasi objek (OOP) dalam pengembangan aplikasi perpustakaan digital

Dengan menjawab rumusan masalah tersebut, diharapkan proyek perpustakaan digital ini dapat menjadi solusi yang efektif dalam memfasilitasi akses informasi dan literatur bagi pengguna, serta memberikan pengalaman praktis bagi mahasiswa dalam penerapan konsep-konsep OOP menggunakan bahasa pemrograman Java.

1.3 Tujuan

Proyek akhir mata kuliah Java Object Oriented Programming 2 pada semester genap 2023-2024 berfokus pada pengembangan aplikasi perpustakaan digital. Tujuan dari proyek ini adalah sebagai berikut:

1. Meningkatkan Pemahaman dan Keterampilan dalam Pemrograman Berorientasi Objek (OOP).
2. Mengembangkan Sistem Perpustakaan Digital yang Efisien.
3. Meningkatkan Aksesibilitas dan Kemudahan Pencarian Informasi.
4. Memastikan Keamanan dan Privasi Data
5. Mengintegrasikan Fitur Peminjaman dan Pengembalian Buku Secara Online
6. Memberikan Pengalaman Praktis dalam Pengembangan Aplikasi.

Dengan mencapai tujuan-tujuan ini, proyek perpustakaan digital diharapkan dapat memberikan manfaat nyata dalam memfasilitasi akses informasi bagi pengguna serta meningkatkan kompetensi mahasiswa dalam bidang pemrograman berorientasi objek dan pengembangan aplikasi berbasis Java.

1.4 Manfaat

Pengembangan perpustakaan digital sebagai proyek akhir mata kuliah Java Object Oriented Programming 2 pada semester genap 2023-2024 diharapkan dapat memberikan berbagai manfaat sebagai berikut:

1. Manfaat bagi Mahasiswa

- a. Peningkatan Keterampilan Teknis: Mahasiswa dapat meningkatkan keterampilan pemrograman mereka, khususnya dalam penerapan konsep-konsep OOP menggunakan bahasa pemrograman Java.
- b. Pengalaman Praktis: Mahasiswa mendapatkan pengalaman praktis dalam merancang, mengembangkan, dan menguji aplikasi perangkat lunak yang kompleks.
- c. Peningkatan Kemampuan Problem-Solving: Melalui proyek ini, mahasiswa dapat mengasah kemampuan mereka dalam memecahkan masalah yang nyata dan kompleks.
- d. Peningkatan Kemampuan Manajemen Proyek: Mahasiswa belajar bagaimana mengelola proyek perangkat lunak, termasuk pengelolaan waktu, sumber daya, dan kolaborasi tim.

2. Manfaat bagi Pengguna

- a. Akses Informasi yang Lebih Mudah: Pengguna dapat mengakses koleksi buku dan bahan bacaan lainnya dengan mudah dan cepat tanpa terbatas oleh waktu dan lokasi.
- b. Efisiensi dalam Pencarian Informasi: Fitur pencarian yang efisien dan akurat memudahkan pengguna dalam menemukan informasi yang diinginkan.
- c. Kemudahan dalam Peminjaman dan Pengembalian Buku: Fitur peminjaman dan pengembalian buku secara online memudahkan pengguna dalam mengelola transaksi mereka.

- d. Notifikasi Peningkat: Pengguna mendapatkan notifikasi yang membantu mereka mengingat batas waktu pengembalian buku, sehingga mengurangi risiko keterlambatan.
3. Manfaat bagi Horizon University Indonesia
- a. Pengelolaan Koleksi yang Lebih Baik: Sistem perpustakaan digital membantu institusi pendidikan dalam mengelola koleksi mereka secara lebih efisien dan terstruktur.
 - b. Penghematan Biaya dan Ruang: Dengan perpustakaan digital, institusi dapat mengurangi kebutuhan akan ruang fisik untuk menyimpan buku dan menghemat biaya operasional.
 - c. Peningkatan Layanan kepada Mahasiswa dan Staf: Institusi dapat menyediakan layanan perpustakaan yang lebih baik dan modern, meningkatkan kepuasan dan produktivitas mahasiswa serta staf.
4. Manfaat bagi Dunia Pendidikan
- a. Akses Terhadap Sumber Belajar yang Luas: Perpustakaan digital menyediakan akses ke berbagai sumber belajar dari seluruh dunia, meningkatkan kualitas pembelajaran.
 - b. Dukungan terhadap Pembelajaran Jarak Jauh: Dalam situasi pembelajaran jarak jauh, perpustakaan digital menjadi sumber daya yang sangat berharga bagi siswa dan pengajar.
 - c. Kontribusi terhadap Digitalisasi Pendidikan: Proyek ini mendukung tren digitalisasi dalam dunia pendidikan, mempersiapkan institusi untuk masa depan yang lebih terintegrasi secara teknologi.

Dengan manfaat-manfaat ini, proyek perpustakaan digital diharapkan dapat memberikan kontribusi yang signifikan tidak hanya bagi mahasiswa yang terlibat dalam pengembangan, tetapi juga bagi pengguna akhir, institusi pendidikan, dan dunia pendidikan secara umum.

BAB II

PEMBAHASAN PROYEK

2.1 Perpustakaan Digital

Perpustakaan digital adalah sebuah sistem atau platform yang memungkinkan akses, pengelolaan, dan penyebaran koleksi bahan bacaan dan sumber informasi secara elektronik. Berbeda dengan perpustakaan tradisional yang berbasis fisik, perpustakaan digital menyediakan berbagai jenis media digital seperti buku elektronik (e-books), jurnal elektronik, artikel, audio, video, dan format digital lainnya yang dapat diakses melalui perangkat komputer, tablet, atau ponsel pintar. Fitur utama perpustakaan digital meliputi akses informasi yang luas, pencarian dan navigasi yang efisien, manajemen koleksi digital yang baik, serta peminjaman dan pengembalian bahan bacaan secara online. Dengan adanya perpustakaan digital, pengguna dapat mencari dan mengakses informasi kapan saja dan di mana saja, tanpa harus terbatas oleh lokasi dan waktu.

Fitur pencarian yang canggih dalam perpustakaan digital memungkinkan pengguna untuk menemukan informasi secara cepat dan akurat berdasarkan judul, penulis, kata kunci, dan kategori lainnya. Pengguna dapat memanfaatkan filter dan alat navigasi untuk mempersempit hasil pencarian dan menemukan sumber daya yang relevan dengan lebih mudah. Selain itu, sistem manajemen koleksi digital yang efisien mendukung penambahan, penghapusan, dan pembaruan data dengan mudah, serta pengorganisasian koleksi dalam berbagai kategori untuk memudahkan akses dan pemeliharaan. Keamanan dan privasi data juga menjadi fokus utama dalam perpustakaan digital, dengan penerapan sistem autentikasi dan otorisasi yang kuat untuk melindungi data pengguna dan koleksi digital dari ancaman keamanan siber.

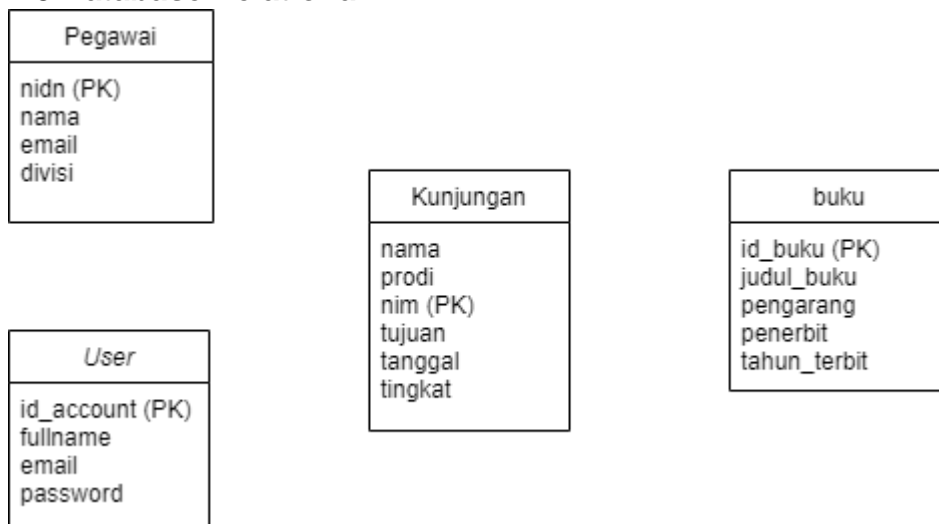
Keuntungan dari perpustakaan digital sangatlah banyak, mulai dari aksesibilitas yang lebih baik hingga efisiensi dan penghematan biaya. Perpustakaan digital memungkinkan akses 24/7 ke sumber daya tanpa terbatas oleh waktu dan lokasi, mendukung pembelajaran jarak jauh dan fleksibilitas dalam mengakses informasi. Selain itu, perpustakaan digital mengurangi kebutuhan ruang fisik untuk menyimpan buku dan bahan bacaan, serta memudahkan dalam pengelolaan koleksi besar dengan fitur otomatisasi dan digitalisasi. Penghematan biaya operasional juga menjadi salah satu keuntungan signifikan, karena perpustakaan digital mengurangi biaya pemeliharaan ruang perpustakaan fisik dan pembelian buku fisik. Dengan

ketersediaan sumber daya terbaru yang dapat diakses dengan cepat, perpustakaan digital menjadi solusi yang efektif dan efisien dalam memenuhi kebutuhan akses informasi di era digital, mendukung proses pembelajaran dan penelitian, serta mempermudah masyarakat umum dalam memperoleh informasi dan pengetahuan secara lebih luas dan mudah.

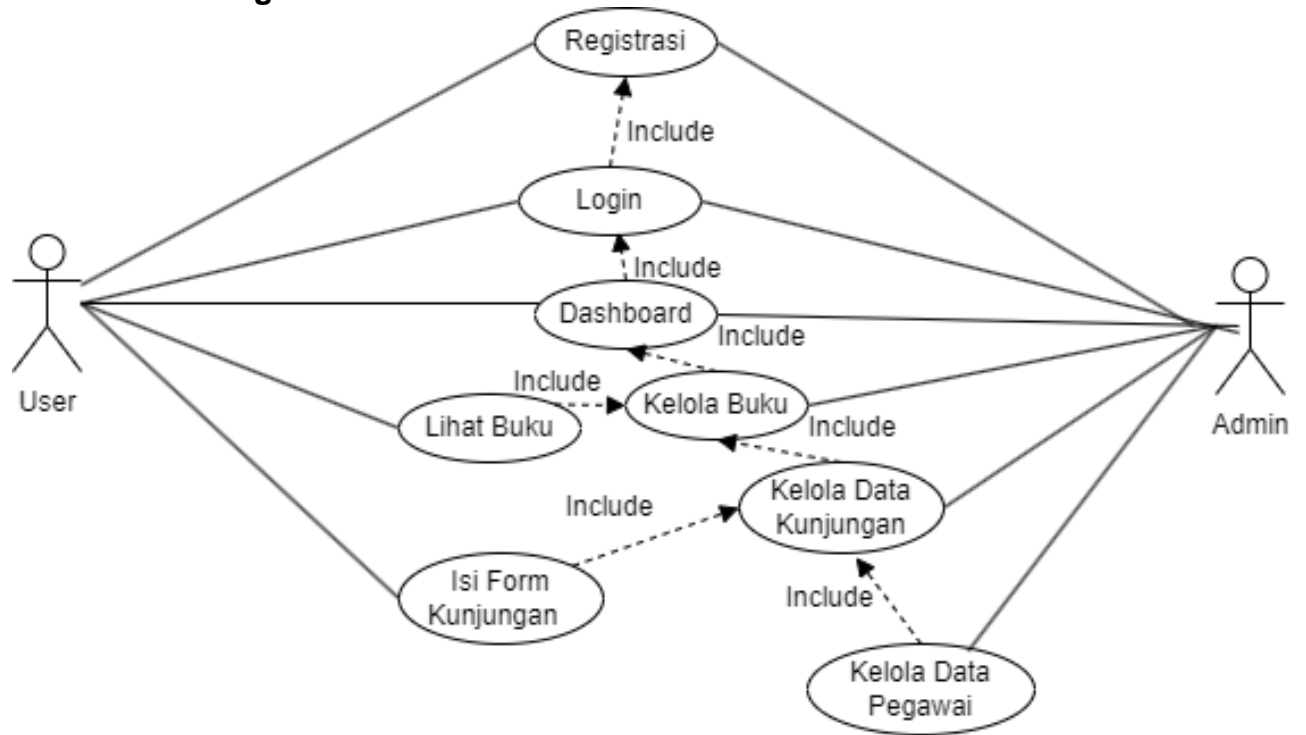
2.2 Tim Pengembang Proyek

1. Project Manager : Liza Andareta, Mila Adelia
2. Frontend Developer : Rifky Permana
3. Backend Developer : Sahrul Maulidi
4. UI/UX Design : Rendi Mah, Mila Adelia, Rifky Permana
5. Technical Writer : M.Fikri Hidayatulloh,S

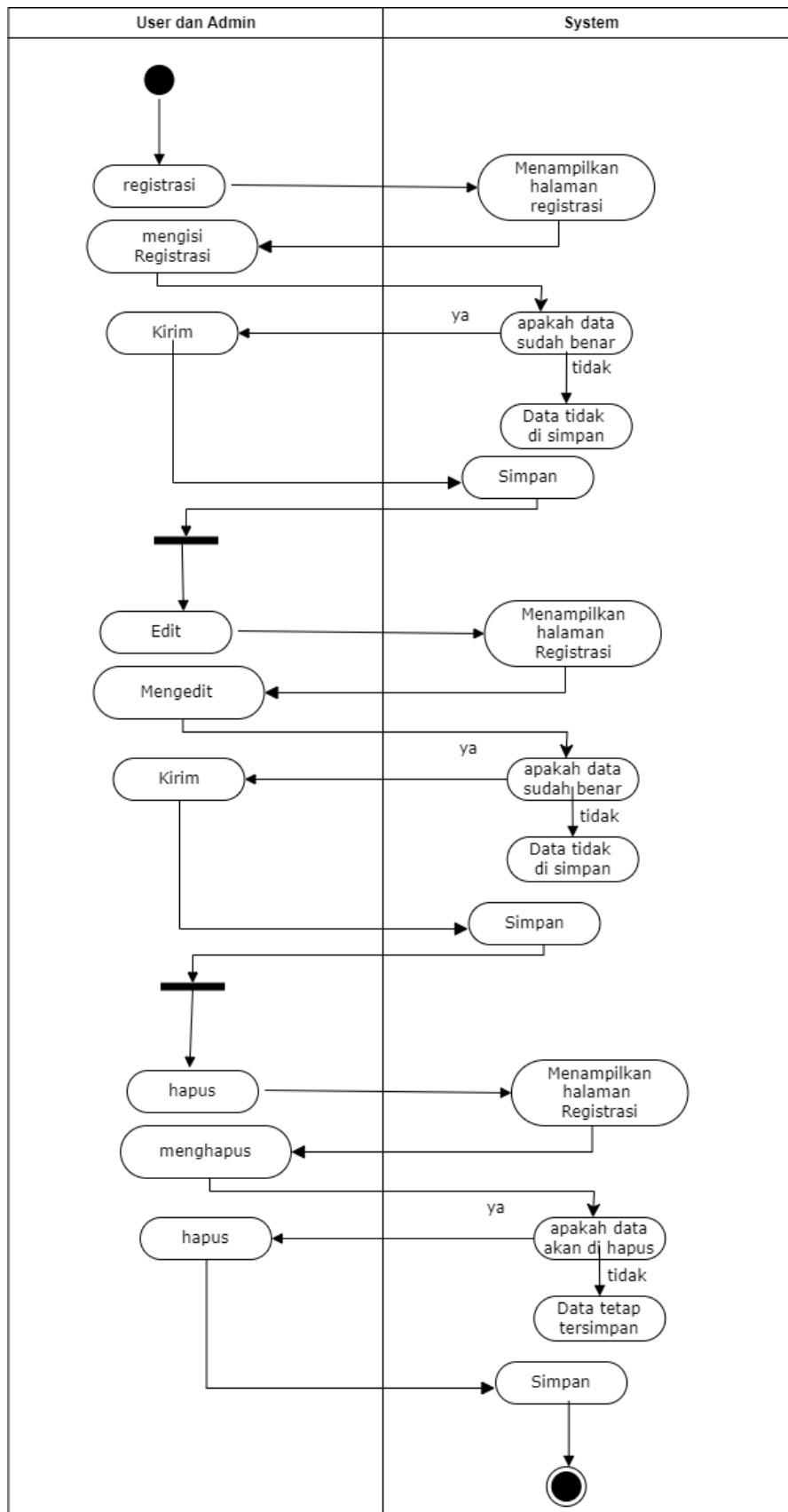
2.3 Database Relational



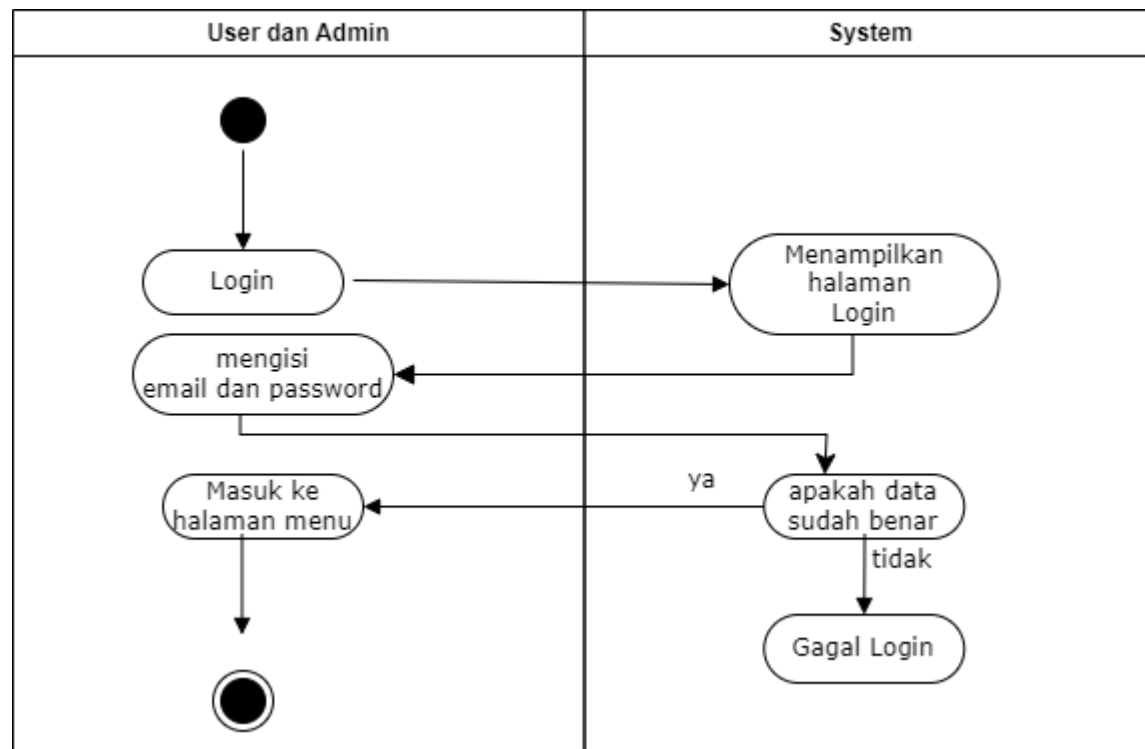
2.4 Usecase Diagram



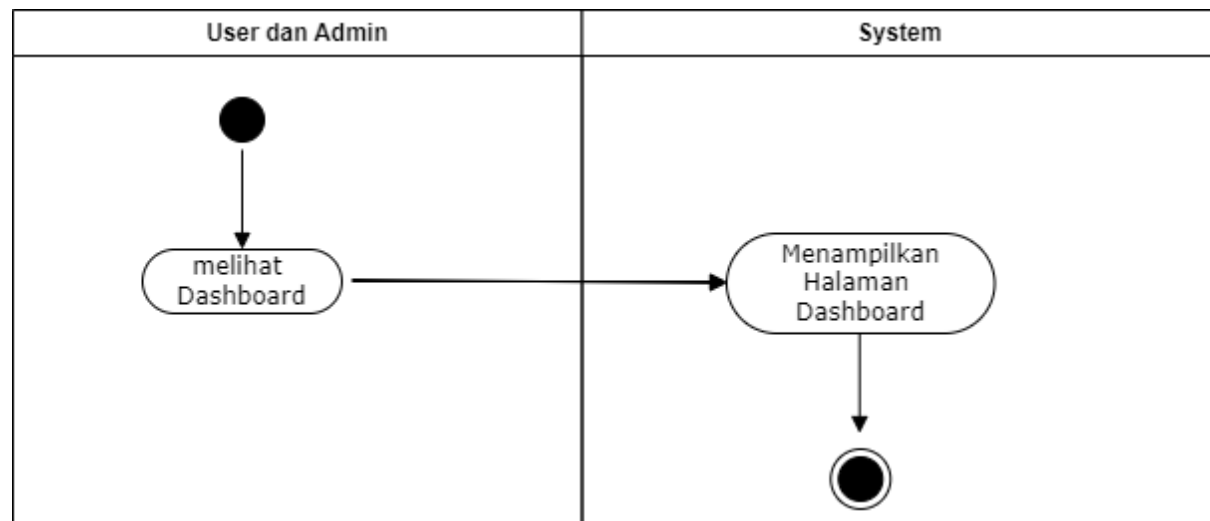
2.5 Activity Diagram Registrasi



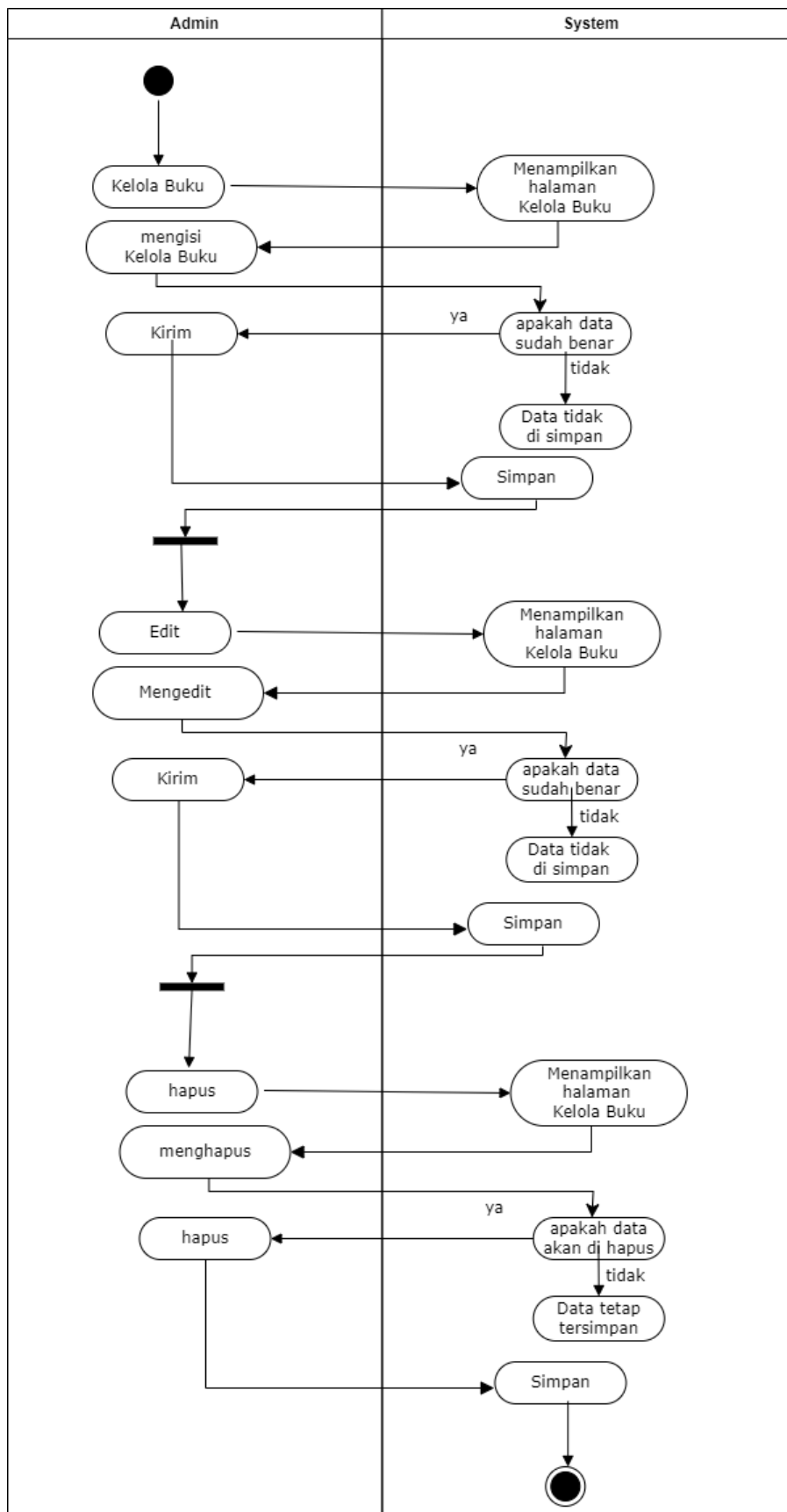
Login



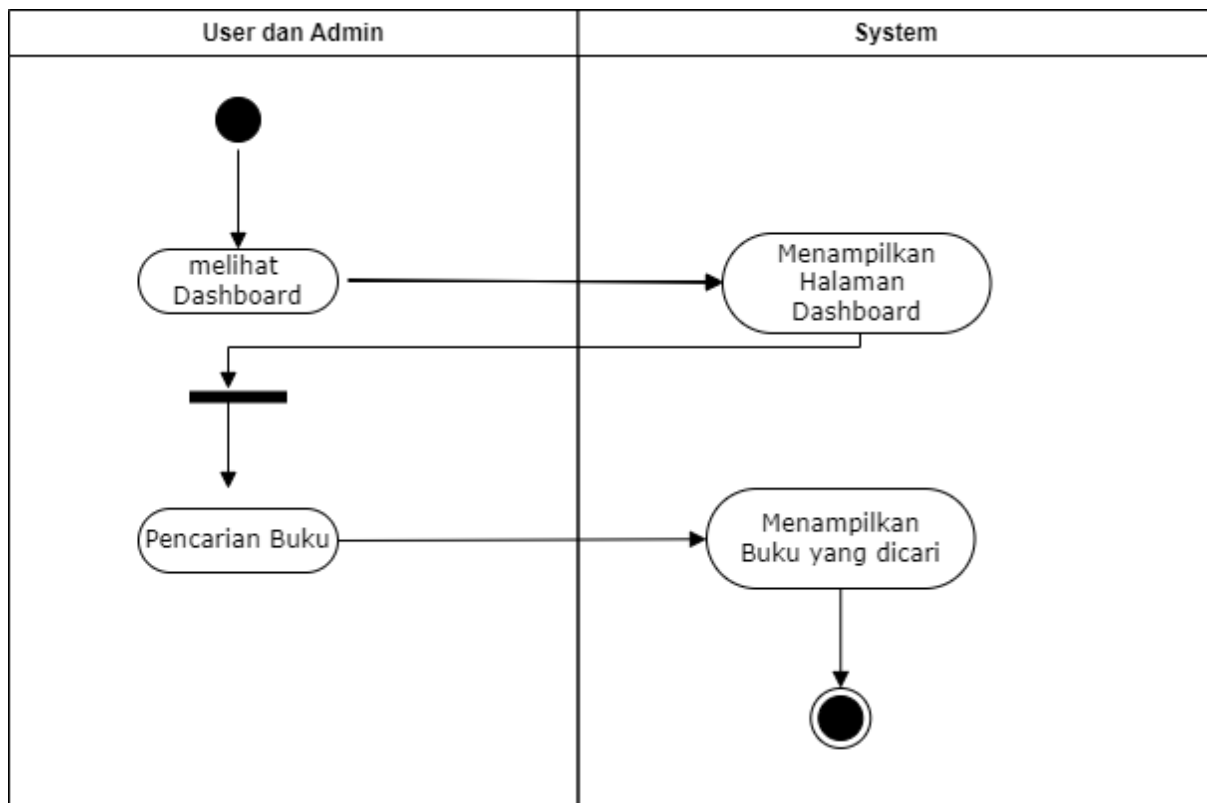
Dashboard



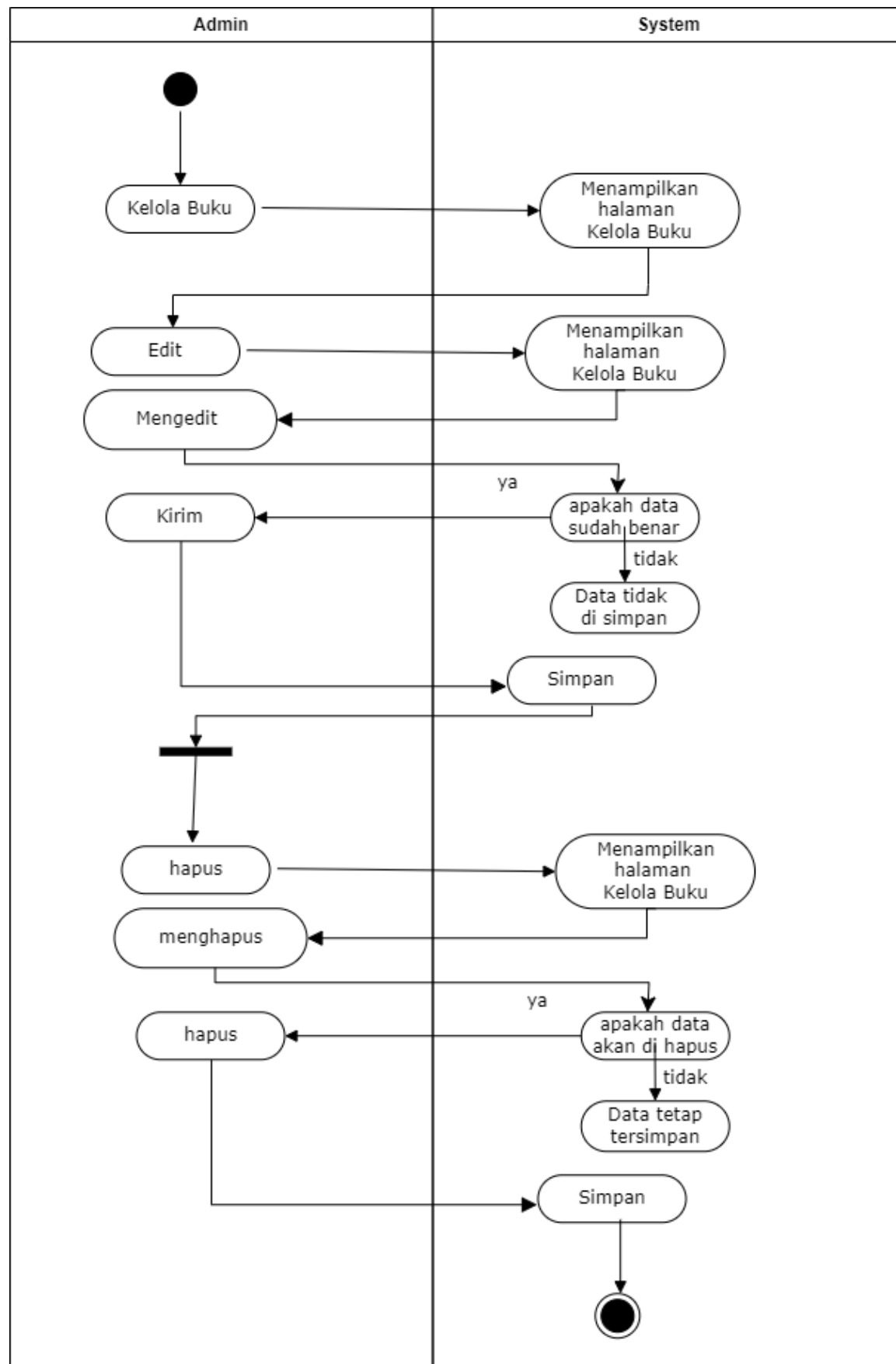
Kelola Buku



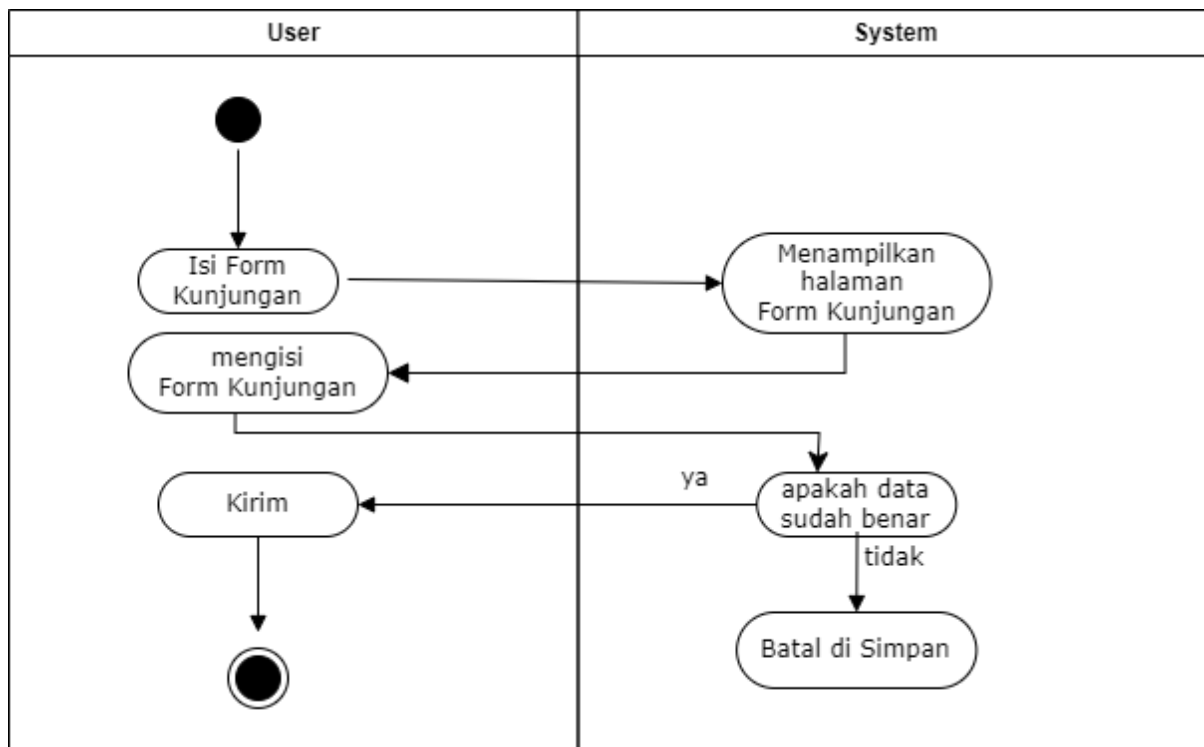
Lihat Buku



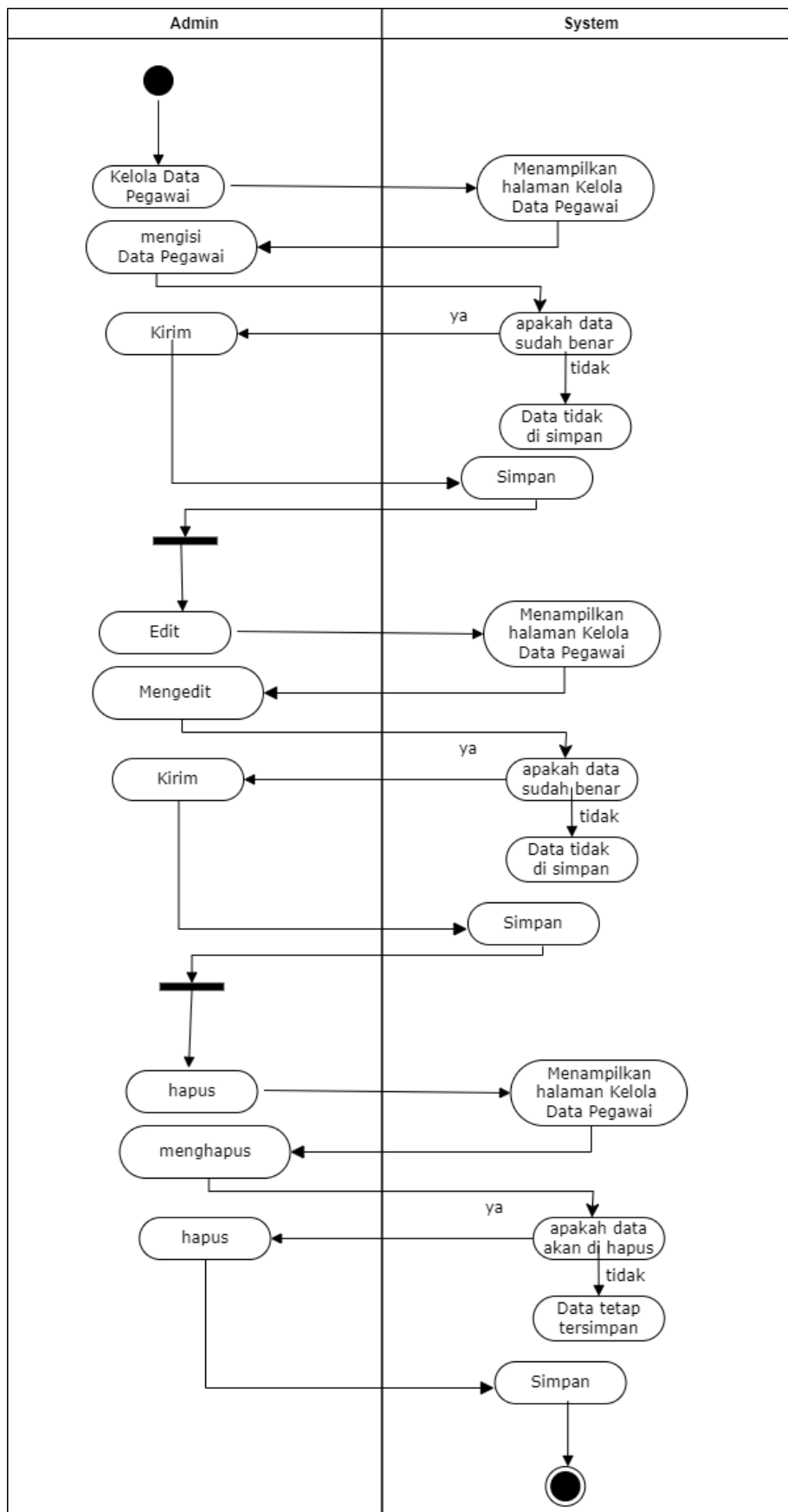
Kelola Data Kunjungan



Isi Form Kunjungan



Kelola Data Pegawai



2.6 Program

Tampilan User

Masuk



Login

A screenshot of a web application window titled "UILogin". The background is a solid dark red. In the center is a white rectangular box containing the login form. At the top of the box, the text "PERDI" is displayed in a large, bold, dark red font, with "Perpustakaan Digital Horizon University Indonesia" below it in a smaller, dark red font. Below this, there are two input fields for "Email/Username :". The first field has a cursor inside. Below the second field, there are two dark red buttons with white text: "LOGIN" and "REGISTER". Between these buttons, the text "Belum mempunyai akun" (Don't have an account) is displayed in a small, dark red font.

Register

Design Preview [register]

PERDI

Perpustakaan Digital Horizon University Indonesia

Full Name :

Email :

Password :

REGISTERSudah punya akunLOGIN

Kelola Buku

PERDI

Hii, Dadan satria

Dashboard

MASTER

Daftar Buku

Kunjungan

Logout

Menu Dashboard

Buku

999

Master > Dashboard

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Daftar Buku

PERDI

Hii, Dadan satria

Dashboard

Daftar Buku

Kunjungan

Logout

Kelola Buku

Master > Kelola Buku

Cari

ID Buku	Judul Buku	Pengarang	Penerbit	Tahun Terbit
KD004	Algoritma	sh	ss	d
KD007	Inggris	gd	ss	33

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Kunjungan

PERDI

Hii, Dadan satria

Dashboard

Daftar Buku

Kunjungan

Logout

Data Kunjungan

Master > Data Kunjungan

TAMBAH

Cari

Nama	Prodi/Dosen	NIM/NIDN	Tujuan	Tanggal	Tingkat
rendi	D3 Kebidanan	12	Mengerjakan Tugas/Skripsi	2024-06-03	2
sgdh	S1 Keperawatan	23	Membaca	2024-06-04	2
rify per	D3 Sistem Informasi	321	Mengerjakan Tugas/Skripsi	2024-06-07	4
hgfg	S1 Keperawatan	43	Membaca	2024-06-06	2

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Tampilan Admin

Dashboard

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Menu Dashboard

Master > Dashboard

Buku
999

Pegawai
999

Kunjungan
999

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Kelola Buku

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Kelola Buku

Master > Kelola Buku

TAMBAH

Cari

ID Buku	Judul Buku	Pengarang	Penerbit	Tahun Terbit
KD004	Algoritma	sh	ss	d
KD007	Inggris	gd	ss	33

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Form Ubah Buku

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Perbarui Data Buku Perpustakaan

Master > Kelola Buku

ID Buku

KD004

Judul Buku

Algoritma

Pengarang

sh

Penerbit

ss

Tahun Terbit

d

PERBARUI

BATAL

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Ubah Buku

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Kelola Buku

Master > Kelola Buku

UBAH

HAPUS

BATAL

Cari

ID Buku	Judul Buku	Pengarang	Penerbit	Tahun Terbit
KD004	Algoritma	sh	ss	d
KD007	inggris	gd	ss	33

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Data Pegawai

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Data Pegawai

Master > Data Pegawai

TAMBAH

Cari

NIDN	Nama Lengkap	Email	Divisi
653653	ahmadss	ahmad@gmail.com	ssss2

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Form Ubah Pegawai

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Perbarui Pegawai Perpustakaan

Master > Data Pegawai

NIDN

653653

Nama Lengkap

ahmadss

Email

ahmad@gmail.com

Divisi

ssss2

PERBARUI

BATAL

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Ubah Pegawai

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Data Pegawai

Master > Data Pegawai

UBAH

HAPUS

BATAL

Cari

NIDN	Nama Lengkap	Email	Divisi
653653	ahmadss	ahmad@gmail.com	ssss2

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Data Kunjungan

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Data Kunjungan

Master > Data Kunjungan

TAMBAH

Cari

Nama	Prodi/Dosen	NIM/NIDN	Tujuan	Tanggal	Tingkat
rendi	D3 Kebidanan	12	Mengerjakan Tugas/Skripsi	2024-06-03	2
rfly per	D3 Sistem Informasi	321	Mengerjakan Tugas/Skripsi	2024-06-07	4

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Form Ubah Kunjungan

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Perbarui Data Buku Perpustakaan

Master > Data Kunjungan

Nama Lengkap

rendi

Prodi/Dosen

D3 Kebidanan

NIM/NIDN

12

Tujuan Kunjungan

Mengerjakan Tugas/Skripsi

Tanggal

Jun 3, 2024

Tingkat

2

PERBARUI

BATAL

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

Ubah Kunjungan

PERDI

Hii, Admin

Dashboard

MASTER

Kelola Buku

Data Pegawai

Data Kunjungan

Logout

Data Kunjungan

Master > Data Kunjungan

UBAH

HAPUS

BATAL

Cari

Nama	Prodi/Dosen	NIM/NIDN	Tujuan	Tanggal	Tingkat
rendi	D3 Kebidanan	12	Mengerjakan Tugas/Skripsi	2024-06-03	2
rtiky per	D3 Sistem Informasi	321	Mengerjakan Tugas/Skripsi	2024-06-07	4

Copyright © 2024 | PERDI (Perpustakaan Digital) . All Rights Reserved.

2.6 Code/Script

Login

```
String Email, Password, query, fname = null, passDb = null;

String SUrl, SUser, SPass;

SUrl = "jdbc:mysql://localhost:3306/db_perpus";

SUser = "root";

SPass = "";

int Found = 0;

try {

    Class.forName("com.mysql.cj.jdbc.Driver");

    Connection con = DriverManager.getConnection(SUrl, SUser, SPass);

    Statement st = con.createStatement();

    if ("".equals(email.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "Email harus diisi",
"Error", JOptionPane.ERROR_MESSAGE);

    } else if ("".equals(new String(password.getPassword()))) {

        JOptionPane.showMessageDialog(new JFrame(), "Password harus diisi",
"Error", JOptionPane.ERROR_MESSAGE);

    } else {

        Email = email.getText();

        Password = new String(password.getPassword());

        query = "SELECT * FROM user WHERE email =" + Email + "";

        ResultSet rs = st.executeQuery(query);

        while (rs.next()) {

            passDb = rs.getString("password");
```

```

        fname = rs.getString("fullname");

        Found = 1;
    }

    if (Found == 1 && Password.equals(passDb)) {

        if (Email.equals("admin1@gmail.com")) {

            if (Password.equals("admin123")) {

                dasboard1 HomeFrame = new dasboard1();

                HomeFrame.setUser(fname);

                HomeFrame.setVisible(true);

                HomeFrame.pack();

                HomeFrame.setLocationRelativeTo(null);

                this.dispose();

            } else {

                JOptionPane.showMessageDialog(new JFrame(), "Email dan
                Password Salah", "Error", JOptionPane.ERROR_MESSAGE);

            }

        } else {

            dasboarduser UserFrame = new dasboarduser();

            UserFrame.setUser(fname);

            UserFrame.setVisible(true);

            UserFrame.pack();

            UserFrame.setLocationRelativeTo(null);

            this.dispose();

        }

    } else {

```

```

        JOptionPane.showMessageDialog(new JFrame(), "Email dan Password
Salah", "Error", JOptionPane.ERROR_MESSAGE);

    }

}

} catch (Exception e) {

    System.out.println("Error! " + e.getMessage());

}

```

Registrasi

String fullname, email, password, query;

```

String SUrl, SUser, SPass;

SUrl = "jdbc:MySQL://localhost:3306/db_perpus";

SUser = "root";

SPass = "";

try {

    Class.forName("com.mysql.cj.jdbc.Driver");

    Connection con = DriverManager.getConnection(SUrl, SUser, SPass);

    Statement st = con.createStatement();

    if ("".equals(fname.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "Full Name harus di isi",
"Error", JOptionPane.ERROR_MESSAGE);

    } else if ("".equals(mail.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "Email harus diisi", "Error",
JOptionPane.ERROR_MESSAGE);

    } else if ("".equals(jPasswordField1.getText())) {

```

```

        JOptionPane.showMessageDialog(new JFrame(), "Password harus diisi",
"Error", JOptionPane.ERROR_MESSAGE);

    } else {

        fullname = fname.getText();

        email = mail.getText();

        password = jPasswordField1.getText();


        query = "INSERT INTO user(fullname, email, password) VALUES('" +
fullname + "', '" + email + "', '" + password + "')";

        st.execute(query);

        fname.setText("");

        mail.setText("");

        jPasswordField1.setText("");

        JOptionPane.showMessageDialog(null, "New Account has been Created");


        // Dispose the current window and open the login window

        JFrame currentFrame = (JFrame)
SwingUtilities.getWindowAncestor(jButton2);

        currentFrame.dispose();

        login loginFrame = new login();

        loginFrame.setVisible(true);

    }

} catch (Exception e) {

    System.out.println("Error! " + e.getMessage());

}

```

Kelola Buku

```
import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import static javax.swing.JOptionPane.showMessageDialog;

import javax.swing.table.DefaultTableModel;
```

```
public class kelbuku extends javax.swing.JPanel {
```

```
    public final void loadData() {

        btnHapus.setVisible(false);

        btnCancel.setVisible(false);

        String SUrl, SUser, SPass;

        SUrl = "jdbc:MySQL://localhost:3306/db_perpus";

        SUser = "root";

        SPass = "";

        try {
```

```

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection(SUrl, SUser, SPass);

Statement st = con.createStatement();

DefaultTableModel model = new DefaultTableModel(new String[]{"ID Buku",
"Judul Buku", "Pengarang", "Penerbit", "Tahun Terbit"}, 0);

tabelbuku.setModel(model);

String sql = "SELECT * FROM buku";

java.sql.ResultSet rs = st.executeQuery(sql);

String idBuku, judulBuku, pengarang, penerbit, tahunTerbit;

while (rs.next()) {

    idBuku = rs.getString("id_buku");

    judulBuku = rs.getString("judul_buku");

    pengarang = rs.getString("pengarang");

    penerbit = rs.getString("penerbit");

    tahunTerbit = rs.getString("tahun_terbit");

    model.addRow(new Object[]{idBuku, judulBuku, pengarang, penerbit,
tahunTerbit});

}

} catch (Exception e) {

    System.out.println("Error " + e.getMessage());

}

}

public kelbuku() {

    initComponents();

```

```
loadData();  
}
```

```
private void showPanel() {  
    panelmain.removeAll();  
    panelmain.add(new kelbuku());  
    panelmain.repaint();  
    panelmain.revalidate();  
}
```

```
private void addBukuActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    if (addBuku.getText().equals("TAMBAH")) {  
        addBuku.setText("SIMPAN");  
    } else if (addBuku.getText().equals("SIMPAN")) {  
        tambahdata();  
    } else if (addBuku.getText().equals("PERBARUI")) {  
        updateData();  
    }  
}
```

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    panelmain.removeAll();
```

```
    panelmain.add(panelview);  
    panelmain.repaint();  
    panelmain.revalidate();  
}
```

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    panelmain.removeAll();  
    panelmain.add(paneladd);  
    panelmain.repaint();  
    panelmain.revalidate();  
  
    if (btnAdd.getText().equals("UBAH")) {  
        dataTabel();  
        addBuku.setText("PERBARUI");  
    }  
}
```

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

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



```
hapusData();  
  
showPanel();  
  
loadData();  
}
```

```
private void btnCancelActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    showPanel();  
  
    loadData();  
}
```

```
private void tabelbukuMouseClicked(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    if (btnAdd.getText().equals("TAMBAH")) {  
  
        btnAdd.setText("UBAH");  
  
        btnHapus.setVisible(true);  
  
        btnCancel.setVisible(true);  
  
    }  
}
```

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

```
private void dispose() {

    throw new UnsupportedOperationException("Not supported yet."); // Generated
from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}
```

```
void pack() {

    throw new UnsupportedOperationException("Not supported yet."); // Generated
from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}
```

```
void setLocationRelativeTo(Object object) {

    throw new UnsupportedOperationException("Not supported yet."); // Generated
from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}
```

```
private void tambahdata() {

    String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";

    String dbUser = "root";

    String dbPass = "";

    String sql = "INSERT INTO buku (id_buku, judul_buku, pengarang, penerbit,
tahun_terbit) VALUES (?, ?, ?, ?, ?)";
```

```

try {

    Class.forName("com.mysql.cj.jdbc.Driver");

    Connection conn = DriverManager.getConnection(dbUrl, dbUser, dbPass);

    if ("".equals(idBuku.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "ID Buku harus diisi",
        "Error", JOptionPane.WARNING_MESSAGE);

    } else if ("".equals(judulBuku.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "Judul Buku harus diisi",
        "Error", JOptionPane.WARNING_MESSAGE);

    } else if ("".equals(pengarang.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "Pengarang harus diisi",
        "Error", JOptionPane.WARNING_MESSAGE);

    } else if ("".equals(penerbit.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "Penerbit harus diisi",
        "Error", JOptionPane.WARNING_MESSAGE);

    } else if ("".equals(thnTerbit.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "Tahun Terbit harus
        diisi", "Error", JOptionPane.WARNING_MESSAGE);

    } else {

        try (PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, idBuku.getText());

            stmt.setString(2, judulBuku.getText());

            stmt.setString(3, pengarang.getText());

            stmt.setString(4, penerbit.getText());

```

```

        stmt.setString(5, thnTerbit.getText());

        stmt.executeUpdate();

        JOptionPane.showMessageDialog(this, "Data berhasil disimpan!");

        idBuku.setText("");
        judulBuku.setText("");
        pengarang.setText("");
        penerbit.setText("");
        thnTerbit.setText("");

        showPanel();

        loadData();
    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

    } finally {

        conn.close();

    }

}

} catch (Exception e) {

    JOptionPane.showMessageDialog(new JFrame(), "Error! " +
e.getMessage());

}

}

private void dataTabel() {

    panelview.setVisible(false);

```

```
paneladd.setVisible(true);
```

```
int row = tabelbuku.getSelectedRow();
```

```
labelbuku.setText("Perbarui Data Buku Perpustakaan");
```

```
idBuku.setEnabled(false);
```

```
idBuku.setText(tabelbuku.getValueAt(row, 0).toString());
```

```
judulBuku.setText(tabelbuku.getValueAt(row, 1).toString());
```

```
pengarang.setText(tabelbuku.getValueAt(row, 2).toString());
```

```
penerbit.setText(tabelbuku.getValueAt(row, 3).toString());
```

```
thnTerbit.setText(tabelbuku.getValueAt(row, 4).toString());
```

```
}
```

```
private void updateData() {
```

```
    String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";
```

```
    String dbUser = "root";
```

```
    String dbPass = "";
```

```
    String sql = "UPDATE buku SET judul_buku=?, pengarang=?, penerbit=?,  
tahun_terbit=? WHERE id_buku=?";
```

```
try {
```

```
    Class.forName("com.mysql.cj.jdbc.Driver");
```

```
    Connection conn = DriverManager.getConnection(dbUrl, dbUser, dbPass);
```

```

        if ("".equals(idBuku.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "ID Buku harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if ("".equals(judulBuku.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "Judul Buku harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if ("".equals(pengarang.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "Pengarang harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if ("".equals(penerbit.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "Penerbit harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if ("".equals(thnTerbit.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "Tahun Terbit harus
diisi", "Error", JOptionPane.WARNING_MESSAGE);

        } else {

            try (PreparedStatement stmt = conn.prepareStatement(sql)) {

                stmt.setString(1, judulBuku.getText());

                stmt.setString(2, pengarang.getText());

                stmt.setString(3, penerbit.getText());

                stmt.setString(4, thnTerbit.getText());

                stmt.setString(5, idBuku.getText());

                stmt.executeUpdate();

```

```

        JOptionPane.showMessageDialog(this, "Data berhasil diperbarui!");

        idBuku.setText("");
        judulBuku.setText("");
        pengarang.setText("");
        penerbit.setText("");
        thnTerbit.setText("");

        showPanel();

        loadData();
    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

    } finally {

        conn.close();

    }

}

} catch (Exception e) {

    JOptionPane.showMessageDialog(new JFrame(), "Error! " +
e.getMessage());

}

}

private void hapusData() {

    int selectedRow = tabelbuku.getSelectedRow();

    if (selectedRow == -1) {

```

```

        JOptionPane.showMessageDialog(this, "Pilih baris yang akan dihapus",
        "Error", JOptionPane.WARNING_MESSAGE);

        return;
    }

```

```

int confirm = JOptionPane.showConfirmDialog(this,

        "Apakah yakin ingin menghapus data?",

        "Konfirmasi Hapus Data", JOptionPane.YES_NO_OPTION);

if (confirm == JOptionPane.YES_OPTION) {

    String idBuku = (String) tabelbuku.getValueAt(selectedRow, 0);

    String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";

    String dbUser = "root";

    String dbPass = "";

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection con = DriverManager.getConnection(dbUrl, dbUser, dbPass);

        String sql = "DELETE FROM buku WHERE id_buku = ?";

        PreparedStatement pst = con.prepareStatement(sql);

        pst.setString(1, idBuku);

        int rowsAffected = pst.executeUpdate();

        if (rowsAffected > 0) {

            ((DefaultTableModel) tabelbuku.getModel()).removeRow(selectedRow);

            JOptionPane.showMessageDialog(this, "Data berhasil dihapus");

        } else {

```



```

        JOptionPane.showMessageDialog(this, "Gagal menghapus data",
"Error", JOptionPane.ERROR_MESSAGE);

    }

    pst.close();

    con.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(this, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);

}

}

}

```

```

private void cariData() {

    String SUrl, SUser, SPass;

    SUrl = "jdbc:MySQL://localhost:3306/db_perpus";

    SUser = "root";

    SPass = "";

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection con = DriverManager.getConnection(SUrl, SUser, SPass);

        Statement st = con.createStatement();

        DefaultTableModel model = new DefaultTableModel(new String[]{"ID Buku",
"Judul Buku", "Pengarang", "Penerbit", "Tahun Terbit"}, 0);

        tabelbuku.setModel(model);
    }
}

```

```

String searchKeyword = txtCari.getText();

String sql = "SELECT * FROM buku WHERE judul_buku LIKE " +
searchKeyword + "%";

java.sql.ResultSet rs = st.executeQuery(sql);

String idBuku, judulBuku, pengarang, penerbit, tahunTerbit;

while (rs.next()) {

    idBuku = rs.getString("id_buku");

    judulBuku = rs.getString("judul_buku");

    pengarang = rs.getString("pengarang");

    penerbit = rs.getString("penerbit");

    tahunTerbit = rs.getString("tahun_terbit");

    model.addRow(new Object[]{idBuku, judulBuku, pengarang, penerbit,
tahunTerbit});

}

} catch (Exception e) {

    System.out.println("Error " + e.getMessage());

}

}

}

```

Kelola Kunjungan

```

import java.sql.Connection;

import java.sql.DriverManager;

```

```

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;


public class kelolakunjungan extends javax.swing.JPanel {


    public final void loadData() {

        btnHapus.setVisible(false);

        btnCancel.setVisible(false);

        String SUrl = "jdbc:MySQL://localhost:3306/db_perpus";

        String SUser = "root";

        String SPass = "";

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection con = DriverManager.getConnection(SUrl, SUser, SPass);

            Statement st = con.createStatement();

            DefaultTableModel model = new DefaultTableModel(new String[]{"Nama",
"Prodi/Dosen", "NIM/NIDN", "Tujuan", "Tanggal", "Tingkat"}, 0);

            tabelkunjungan.setModel(model);

```

```

String sql = "SELECT * FROM kunjungan";

java.sql.ResultSet rs = st.executeQuery(sql);

String nama, prodi, nim, tujuan, tanggal, tingkat;

while (rs.next()) {

    nama = rs.getString("nama");

    prodi = rs.getString("prodi");

    nim = rs.getString("nim");

    tujuan = rs.getString("tujuan");

    tanggal = rs.getString("tanggal");

    tingkat = rs.getString("tingkat");

    model.addRow(new Object[]{nama, prodi, nim, tujuan, tanggal, tingkat});

}

} catch (Exception e) {

    System.out.println("Error " + e.getMessage());

}

}

/**

* Creates new form contoh

*/

public kelolakunjungan() {

    initComponents();

    loadData();

}

```

```

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

    // TODO add your handling code here:

    if (addkunjungan.getText().equals("TAMBAH")) {

        addkunjungan.setText("SIMPAN");

    } else if (addkunjungan.getText().equals("SIMPAN")) {

        tambahdata();

    } else if (addkunjungan.getText().equals("PERBARUI")) {

        updateData();

    }

}

```

```

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    panelmain.removeAll();

    panelmain.add(panelview);

    panelmain.repaint();

    panelmain.revalidate();

}

```

```

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

    // TODO add your handling code here:

}

```

```

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

    // TODO add your handling code here:

    panelmain.removeAll();

    panelmain.add(paneladd);

    panelmain.repaint();

    panelmain.revalidate();


    if (btnAdd.getText().equals("UBAH")) {

        dataTabel();

        addkunjungan.setText("PERBARUI");

    }

}

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

    // TODO add your handling code here:

    hapusData();

    showPanel();

    loadData();

}

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

    // TODO add your handling code here:

    showPanel();

}

```

```
    loadData();  
}
```

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

```
private void tabelkunjunganMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    if (btnAdd.getText().equals("TAMBAH")) {  
        btnAdd.setText("UBAH");  
        btnHapus.setVisible(true);  
        btnCancel.setVisible(true);  
    }  
}
```

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

```
private void dispose() {
```

```
        throw new UnsupportedOperationException("Not supported yet."); // Generated
from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

    }
```

```
private void showPanel() {

    panelmain.removeAll();

    panelmain.add(new kelolakunjungan());

    panelmain.repaint();

    panelmain.revalidate();

}
```

```
private void tambahdata() {

    String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";

    String dbUser = "root";

    String dbPass = "";
```

```
    String sqlInsert = "INSERT INTO kunjungan (nama, prodi, nim, tujuan, tanggal,
tingkat) VALUES (?, ?, ?, ?, ?, ?)";
```

```
    String sqlUpdate = "UPDATE kunjungan SET nama = ?, prodi = ?, tujuan = ?,
tanggal = ?, tingkat = ? WHERE nim = ?";
```

```
try {

    Class.forName("com.mysql.cj.jdbc.Driver");

    Connection conn = DriverManager.getConnection(dbUrl, dbUser, dbPass);
```



```

// Check if fields are empty

if (nama.getText().trim().isEmpty()) {

    JOptionPane.showMessageDialog(new JFrame(), "Nama harus diisi",
    "Error", JOptionPane.WARNING_MESSAGE);

    } else if (prodi.getSelectedItemAt() == null ||
prodi.getSelectedItemAt().toString().trim().isEmpty()) {

        JOptionPane.showMessageDialog(new JFrame(), "Prodi harus diisi",
        "Error", JOptionPane.WARNING_MESSAGE);

        } else if (nim.getText().trim().isEmpty()) {

            JOptionPane.showMessageDialog(new JFrame(), "NIM harus diisi",
            "Error", JOptionPane.WARNING_MESSAGE);

            } else if (tujuan.getSelectedItemAt() == null ||
tujuan.getSelectedItemAt().toString().trim().isEmpty()) {

                JOptionPane.showMessageDialog(new JFrame(), "Tujuan harus diisi",
                "Error", JOptionPane.WARNING_MESSAGE);

                } else if (tanggal.getDate() == null) {

                    JOptionPane.showMessageDialog(new JFrame(), "Tanggal harus diisi",
                    "Error", JOptionPane.WARNING_MESSAGE);

                    } else if (tingkat.getText().trim().isEmpty()) {

                        JOptionPane.showMessageDialog(new JFrame(), "Tingkat harus diisi",
                        "Error", JOptionPane.WARNING_MESSAGE);

                        } else {

                            String sql = addkunjungan.getText().equals("PERBARUI") ? sqlUpdate :
sqlInsert;

                            try (PreparedStatement stmt = conn.prepareStatement(sql)) {

```

```

stmt.setString(1, nama.getText());

stmt.setString(2, prodi.getSelectedItems().toString());

stmt.setString(3, nim.getText());

stmt.setString(4, tujuan.getSelectedItems().toString());

stmt.setDate(5, new java.sql.Date(tanggal.getDate().getTime())); //

```

Convert to java.sql.Date

```

stmt.setString(6, tingkat.getText());

if (addkunjungan.getText().equals("PERBARUI")) {
    stmt.setString(7, nim.getText()); // Set NIM for WHERE clause
}

stmt.executeUpdate();

OptionPane.showMessageDialog(this, "Data berhasil disimpan!");

nama.setText("");

prodi.setSelectedIndex(-1);

nim.setText("");

tujuan.setSelectedIndex(-1);

tanggal.setDate(null);

tingkat.setText("");

showPanel();

loadData();

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

} finally {

```

```

        conn.close();
    }
}

} catch (Exception e) {

    JOptionPane.showMessageDialog(new JFrame(), "Error! " +
e.getMessage());
}
}

private void dataTabel() {

    panelview.setVisible(false);
    paneladd.setVisible(true);

    int row = tabelkunjungan.getSelectedRow();

    if (row == -1) {

        JOptionPane.showMessageDialog(this, "Please select a row first.");

        return;
    }

    labelkunjungan.setText("Perbarui Data Buku Perpustakaan");

    nim.setEnabled(false);

    // Fetch and set the values from the table to the form fields

    try {

```

```

String namaVal = tabelkunjungan.getValueAt(row, 0).toString();
String prodiVal = tabelkunjungan.getValueAt(row, 1).toString();
String nimVal = tabelkunjungan.getValueAt(row, 2).toString();
String tujuanVal = tabelkunjungan.getValueAt(row, 3).toString();
String tanggalVal = tabelkunjungan.getValueAt(row, 4).toString();
String tingkatVal = tabelkunjungan.getValueAt(row, 5).toString();

// Set the values to the form fields
nama.setText(namaVal);
prodi.setSelectedItem(prodiVal);
nim.setText(nimVal);
tujuan.setSelectedItem(tujuanVal);

// Convert string to date and set it to the date picker
java.util.Date date = new SimpleDateFormat("yyyy-MM-
dd").parse(tanggalVal);
tanggal.setDate(date);

tingkat.setText(tingkatVal);

} catch (Exception e) {
    e.printStackTrace();

    JOptionPane.showMessageDialog(this, "Error setting data: " +
e.getMessage());
}

```

```

}

private void updateData() {

    String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";

    String dbUser = "root";

    String dbPass = "";

    String sql = "UPDATE kunjungan SET nama=?, prodi=?, tujuan=?, tanggal=?,
tingkat=? WHERE nim=?";

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection conn = DriverManager.getConnection(dbUrl, dbUser, dbPass);

        // Validate form fields

        if (nama.getText().trim().isEmpty()) {

            JOptionPane.showMessageDialog(new JFrame(), "Nama harus diisi",
            "Error", JOptionPane.WARNING_MESSAGE);

        } else if (prodi.getSelectedIndex() == null ||
prodi.getSelectedIndex().toString().trim().isEmpty()) {

            JOptionPane.showMessageDialog(new JFrame(), "Prodi harus diisi",
            "Error", JOptionPane.WARNING_MESSAGE);

        } else if (nim.getText().trim().isEmpty()) {

            JOptionPane.showMessageDialog(new JFrame(), "NIM harus diisi",
            "Error", JOptionPane.WARNING_MESSAGE);

```

```

        } else if (tujuan.getSelectedItemId() == null ||
tujuan.getSelectedItemId().toString().trim().isEmpty()) {

            JOptionPane.showMessageDialog(new JFrame(), "Tujuan harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if (tanggal.getDate() == null) {

            JOptionPane.showMessageDialog(new JFrame(), "Tanggal harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if (tingkat.getText().trim().isEmpty()) {

            JOptionPane.showMessageDialog(new JFrame(), "Tingkat harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else {

            // Update the database record

            try (PreparedStatement stmt = conn.prepareStatement(sql)) {

                stmt.setString(1, nama.getText());

                stmt.setString(2, prodi.getSelectedItemId().toString());

                stmt.setString(3, tujuan.getSelectedItemId().toString());

                stmt.setDate(4, new java.sql.Date(tanggal.getDate().getTime())); //
Convert to java.sql.Date

                stmt.setString(5, tingkat.getText());

                stmt.setString(6, nim.getText());

                stmt.executeUpdate();

                JOptionPane.showMessageDialog(this, "Data berhasil disimpan!");

                // Reset form fields after successful update

                nama.setText("");

```

```

        prodi.setSelectedIndex(-1);

        nim.setText("");

        tujuan.setSelectedIndex(-1);

        tanggal.setDate(null);

        tingkat.setText("");

        showPanel();

        loadData(); // Reload the table data to reflect the updates
    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

    } finally {

        conn.close();

    }

}

} catch (Exception e) {

    JOptionPane.showMessageDialog(new JFrame(), "Error! " +
e.getMessage());

}

}

```

```

private void hapusData() {

    int selectedRow = tabelkunjungan.getSelectedRow();

    if (selectedRow == -1) {

        JOptionPane.showMessageDialog(this, "Pilih baris yang akan dihapus",
"Error", JOptionPane.WARNING_MESSAGE);

        return;
    }
}

```

```

    }

    int confirm = JOptionPane.showConfirmDialog(this,
        "Apakah yakin ingin menghapus data?",
        "Konfirmasi Hapus Data", JOptionPane.YES_NO_OPTION);
    if (confirm == JOptionPane.YES_OPTION) {
        String nama = (String) tabelkunjungan.getValueAt(selectedRow, 0);
        String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";
        String dbUser = "root";
        String dbPass = "";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(dbUrl, dbUser, dbPass);
            String sql = "DELETE FROM kunjungan WHERE nama = ?";
            PreparedStatement pst = con.prepareStatement(sql);
            pst.setString(1, nama);
            int rowsAffected = pst.executeUpdate();
            if (rowsAffected > 0) {
                ((DefaultTableModel)
tabelkunjungan.getModel()).removeRow(selectedRow);
                JOptionPane.showMessageDialog(this, "Data berhasil dihapus");
            } else {
                JOptionPane.showMessageDialog(this, "Gagal menghapus data",
"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}

```



```

    }

    pst.close();

    con.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(this, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);

}

}

}

```

```

private void cariData() {

    String SUrl = "jdbc:mysql://localhost:3306/db_perpus";

    String SUser = "root";

    String SPass = "";

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection con = DriverManager.getConnection(SUrl, SUser, SPass);

        DefaultTableModel model = new DefaultTableModel(new String[]{"Nama",
"Prodi/Dosen", "NIM/NIDN", "Tujuan", "Tanggal", "Tingkat"}, 0);

        tabelkunjungan.setModel(model);

        String searchKeyword = txtCari.getText();

        String sql = "SELECT * FROM kunjungan WHERE nama LIKE ?";

        PreparedStatement stmt = con.prepareStatement(sql);
    }
}

```

```

stmt.setString(1, "%" + searchKeyword + "%");

java.sql.ResultSet rs = stmt.executeQuery();

String nama, prodi, nim, tujuan, tanggal, tingkat;
while (rs.next()) {

    nama = rs.getString("nama");

    prodi = rs.getString("prodi");

    nim = rs.getString("nim");

    tujuan = rs.getString("tujuan");

    tanggal = rs.getString("tanggal");

    tingkat = rs.getString("tingkat");

    model.addRow(new Object[]{nama, prodi, nim, tujuan, tanggal, tingkat});

}
} catch (Exception e) {

    System.out.println("Error " + e.getMessage());

}

}
}

```

Kelola Pegawai

```

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

```

```

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;


public class kelolapegawai extends javax.swing.JPanel {

    public final void loadData() {

        btnHapus.setVisible(false);

        btnCancel.setVisible(false);

        String SUrl, SUser, SPass;

        SUrl = "jdbc:MySQL://localhost:3306/db_perpus";

        SUser = "root";

        SPass = "";

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection con = DriverManager.getConnection(SUrl, SUser, SPass);

            Statement st = con.createStatement();

            DefaultTableModel model = new DefaultTableModel(new String[]{"NIDN",
"Nama Lengkap", "Email", "Divisi"}, 0);

            tabelpegawai.setModel(model);

            String sql = "SELECT * FROM pegawai";

            java.sql.ResultSet rs = st.executeQuery(sql);

            String nidn,nama,email,divisi;

```

```

        while (rs.next()) {

            nidn = rs.getString("nidn");

            nama = rs.getString("nama");

            email = rs.getString("email");

            divisi = rs.getString("divisi");

            model.addRow(new Object[]{nidn, nama, email, divisi});

        }

    } catch (Exception e) {

        System.out.println("Error " + e.getMessage());

    }

}

/**
 * Creates new form contoh
 */

public kelolapegawai() {

    initComponents();

    loadData();

}

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

    // TODO add your handling code here:

}

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

```

```

// TODO add your handling code here:

if (addPegawai.getText().equals("TAMBAH")) {
    addPegawai.setText("SIMPAN");
} else if (addPegawai.getText().equals("SIMPAN")) {
    tambahdata();
} else if (addPegawai.getText().equals("PERBARUI")) {
    updateData();
}
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    panelmain.removeAll();
    panelmain.add(panelview);
    panelmain.repaint();
    panelmain.revalidate();
}

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

    showPanel();
    loadData();
}

```

```

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

    // TODO add your handling code here:

    hapusData();

    showPanel();

    loadData();

}

```

```

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

    // TODO add your handling code here:

    panelmain.removeAll();

    panelmain.add(paneladd);

    panelmain.repaint();

    panelmain.revalidate();

    if (btnAdd.getText().equals("UBAH")) {

        dataTabel();

        addPegawai.setText("PERBARUI");

    }

}

```

```

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

    // TODO add your handling code here:

    cariData();

}

```

```

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

    if (btnAdd.getText().equals("TAMBAH")) {
        btnAdd.setText("UBAH");
        btnHapus.setVisible(true);
        btnCancel.setVisible(true);
    }
}

```

```

private void showPanel() {
    panelmain.removeAll();
    panelmain.add(new kelolapegawai());
    panelmain.repaint();
    panelmain.revalidate();
}

```

```

private void dispose() {
    throw new UnsupportedOperationException("Not supported yet."); // Generated
from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}

```

```

private void tambahdata() {

    String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";

    String dbUser = "root";

    String dbPass = "";

    String sql = "INSERT INTO pegawai (nidn, nama, email, divisi) VALUES (?, ?,
?, ?)";

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection conn = DriverManager.getConnection(dbUrl, dbUser, dbPass);

        if ("".equals(nidn.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "NIDN harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if ("".equals(nama.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "Nama harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if ("".equals(email.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "Email harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        } else if ("".equals(divisi.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "Divisi harus diisi",
"Error", JOptionPane.WARNING_MESSAGE);

        }

        else {

```



```

try (PreparedStatement stmt = conn.prepareStatement(sql)) {

    stmt.setString(1, nidn.getText());

    stmt.setString(2, nama.getText());

    stmt.setString(3, email.getText());

    stmt.setString(4, divisi.getText());


    stmt.executeUpdate();

    JOptionPane.showMessageDialog(this, "Data berhasil disimpan!");

    nidn.setText("");

    nama.setText("");

    email.setText("");

    divisi.setText("");

    showPanel();

    loadData();

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

} finally {

    conn.close();

}

} catch (Exception e) {

    JOptionPane.showMessageDialog(new JFrame(), "Error! " +
e.getMessage());

}

}

```

```

private void dataTabel() {

    panelview.setVisible(false);

    paneladd.setVisible(true);


    int row = tabelpegawai.getSelectedRow();

    labelpegawai.setText("Perbarui Pegawai Perpustakaan");


    nidn.setEnabled(false);


    nidn.setText(tabelpegawai.getValueAt(row, 0).toString());
    nama.setText(tabelpegawai.getValueAt(row, 1).toString());
    email.setText(tabelpegawai.getValueAt(row, 2).toString());
    divisi.setText(tabelpegawai.getValueAt(row, 3).toString());
}

```

```

private void updateData() {

    String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";

    String dbUser = "root";

    String dbPass = "";


    String sql = "UPDATE pegawai SET nama=?, email=?, divisi=? WHERE
nidn=?";


    try {

```

```

Class.forName("com.mysql.cj.jdbc.Driver");

Connection conn = DriverManager.getConnection(dbUrl, dbUser, dbPass);

if ("".equals(nidn.getText())) {

    JOptionPane.showMessageDialog(new JFrame(), "NIDN harus diisi",
    "Error", JOptionPane.WARNING_MESSAGE);

    } else if ("".equals(nama.getText())) {

        JOptionPane.showMessageDialog(new JFrame(), "Nama harus diisi",
        "Error", JOptionPane.WARNING_MESSAGE);

        } else if ("".equals(email.getText())) {

            JOptionPane.showMessageDialog(new JFrame(), "Email harus diisi",
            "Error", JOptionPane.WARNING_MESSAGE);

            } else if ("".equals(divisi.getText())) {

                JOptionPane.showMessageDialog(new JFrame(), "Divisi harus diisi",
                "Error", JOptionPane.WARNING_MESSAGE);

                } else {

                    try (PreparedStatement stmt = conn.prepareStatement(sql)) {

                        stmt.setString(1, nama.getText());

                        stmt.setString(2, email.getText());

                        stmt.setString(3, divisi.getText());

                        stmt.setString(4, nidn.getText());

                        stmt.executeUpdate();

```

```

        JOptionPane.showMessageDialog(this, "Data berhasil diperbarui!");

        nidn.setText("");

        nama.setText("");

        email.setText("");

        divisi.setText("");

        showPanel();

        loadData();

    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

    } finally {

        conn.close();

    }

}

} catch (Exception e) {

    JOptionPane.showMessageDialog(new JFrame(), "Error! " +
e.getMessage());

}

}

private void hapusData() {

    int selectedRow = tabelpegawai.getSelectedRow();

    if (selectedRow == -1) {

        JOptionPane.showMessageDialog(this, "Pilih baris yang akan dihapus",
"Error", JOptionPane.WARNING_MESSAGE);

```

```

        return;
    }

    int confirm = JOptionPane.showConfirmDialog(this,
        "Apakah yakin ingin menghapus data?",
        "Konfirmasi Hapus Data", JOptionPane.YES_NO_OPTION);
    if (confirm == JOptionPane.YES_OPTION) {
        String idBuku = (String) tabelpegawai.getValueAt(selectedRow, 0);
        String dbUrl = "jdbc:mysql://localhost:3306/db_perpus";
        String dbUser = "root";
        String dbPass = "";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(dbUrl, dbUser, dbPass);
            String sql = "DELETE FROM pegawai WHERE nidn = ?";
            PreparedStatement pst = con.prepareStatement(sql);
            pst.setString(1, idBuku);
            int rowsAffected = pst.executeUpdate();
            if (rowsAffected > 0) {
                ((DefaultTableModel)
tabelpegawai.getModel()).removeRow(selectedRow);
                JOptionPane.showMessageDialog(this, "Data berhasil dihapus");
            } else {

```

```

        JOptionPane.showMessageDialog(this, "Gagal menghapus data",
"Error", JOptionPane.ERROR_MESSAGE);

    }

    pst.close();

    con.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(this, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);

}

}

}

```

```

private void cariData() {

    String SUrl, SUser, SPass;

    SUrl = "jdbc:MySQL://localhost:3306/db_perpus";

    SUser = "root";

    SPass = "";

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection con = DriverManager.getConnection(SUrl, SUser, SPass);

        Statement st = con.createStatement();

        DefaultTableModel model = new DefaultTableModel(new String[]{"NIDN",
"Nama Lengkap", "Email", "Divisi"}, 0);

        tabelpegawai.setModel(model);
    }
}

```

```

String searchKeyword = txtCari.getText();

String sql = "SELECT * FROM pegawai WHERE nama LIKE '" +
searchKeyword + "%'";

java.sql.ResultSet rs = st.executeQuery(sql);

String nidn, nama, email, divisi;

while (rs.next()) {

    nidn = rs.getString("nidn");

    nama = rs.getString("nama");

    email = rs.getString("email");

    divisi = rs.getString("divisi");

    model.addRow(new Object[]{nidn, nama, email, divisi});

}

} catch (Exception e) {

    System.out.println("Error " + e.getMessage());

}

}

}

```

BAB III

SIMPULAN DAN SARAN

3.1 Simpulan

Perpustakaan digital merupakan solusi modern yang mengubah cara kita mengakses, mengelola, dan menyebarkan informasi. Dengan menyediakan berbagai jenis media digital seperti buku elektronik, jurnal, artikel, audio, dan video, perpustakaan digital memungkinkan pengguna untuk mengakses informasi kapan saja dan di mana saja. Fitur-fitur canggih seperti pencarian yang efisien, manajemen koleksi yang terorganisir, serta peminjaman dan pengembalian bahan bacaan secara online membuat perpustakaan digital menjadi alat yang sangat berguna dan efisien dalam mendukung pembelajaran, penelitian, dan kebutuhan informasi sehari-hari.

Keuntungan dari perpustakaan digital mencakup aksesibilitas yang lebih baik, efisiensi dalam pengelolaan koleksi, dan penghematan biaya operasional. Dengan perpustakaan digital, kebutuhan akan ruang fisik dan biaya pemeliharaan perpustakaan tradisional dapat dikurangi secara signifikan. Selain itu, sistem keamanan dan privasi yang kuat memastikan bahwa data pengguna dan koleksi digital terlindungi dengan baik, memberikan rasa aman kepada pengguna dalam mengakses dan memanfaatkan sumber daya yang tersedia.

Secara keseluruhan, perpustakaan digital menawarkan berbagai manfaat yang tidak hanya meningkatkan efisiensi dan kemudahan akses informasi, tetapi juga mendukung upaya digitalisasi dalam dunia pendidikan dan penelitian. Dengan adopsi perpustakaan digital, institusi pendidikan dan masyarakat umum dapat menikmati akses yang lebih luas dan mudah terhadap sumber-sumber informasi yang penting, mendukung perkembangan ilmu pengetahuan dan pengetahuan di era digital.

3.2 Saran

Berikut adalah beberapa saran untuk pengembangan lebih lanjut dari aplikasi Perpustakaan Digital:

1. Pengembangan Fitur Tambahan:
 - a. Implementasi fitur ulasan dan rating buku yang memungkinkan anggota untuk memberikan feedback dan rekomendasi.
 - b. Pengembangan fitur rekomendasi buku berdasarkan riwayat peminjaman dan ulasan pengguna lainnya.

2. Integrasi Teknologi Baru:
 - a. Integrasi teknologi RFID untuk peminjaman dan pengembalian buku yang lebih cepat dan akurat.
 - b. Penggunaan teknologi cloud untuk penyimpanan data yang lebih aman dan scalable.
3. Peningkatan Keamanan:
 - a. Penambahan lapisan keamanan seperti two-factor authentication untuk login anggota dan admin.
 - b. Peningkatan mekanisme enkripsi data untuk melindungi informasi sensitif.
4. Optimasi Kinerja:
 - a. Optimasi algoritma pencarian dan pengelolaan database untuk menangani volume data yang lebih besar.
 - b. Penggunaan cache untuk mempercepat akses data yang sering digunakan.
5. Pengujian dan Pemeliharaan:
 - a. Melakukan pengujian lebih lanjut dengan skenario penggunaan yang lebih luas untuk memastikan stabilitas dan keandalan sistem.
 - b. Menyediakan dokumentasi lengkap dan pelatihan untuk admin perpustakaan agar dapat memanfaatkan semua fitur aplikasi dengan maksimal.

Dengan mengikuti saran-saran ini, diharapkan aplikasi Perpustakaan Digital dapat terus berkembang dan memberikan manfaat yang lebih besar bagi pengelola perpustakaan dan anggota. Pengembangan berkelanjutan dan adaptasi terhadap kebutuhan pengguna yang berubah akan memastikan bahwa aplikasi ini tetap relevan dan berguna di masa depan.

DAFTAR PUSTAKA

- Casey, Anne-Marie, dan Fritts, Jeanne R. (2003).** *Managing Digital Collections: Collaborative Initiative of Five Academic Libraries*. Library Administration and Management Association.
- Covey, Denise Troll. (2002).** *Usage and Usability Assessment: Library Practices and Concerns*. Digital Library Federation, Council on Library and Information Resources.
- Cox, Andrew M., dan Pinfield, Stephen. (2014).** *"Research Data Management and Libraries: Current Activities and Future Priorities"*. Journal of Librarianship and Information Science, 46(4), 299-316.
- Harinarayana, N. S., dan Raju, N. V. (2010).** *"Web 2.0 Features in University Library Web Sites"*. The Electronic Library, 28(1), 69-88.
- Yakel, Elizabeth, dan Kim, Jihyun. (2005).** *"Adopting Web-based Interactivity in Library and Information Science Education"*. Journal of Education for Library and Information Science, 46(3), 201-217.
- Tedd, Lucy A. (2005).** *"Digital Libraries: A Conceptual Framework"*. Library Review, 54(4), 218-225.