HealthQueue+

LAPORAN PROYEK AKHIR

COMP6364 – Object Oriented Programming

KELAS LB20

Oleh:

260213459 – IQBAL RAFI MAULANA

2602090435 – AUDRIC NAGATA

2602106894 – MADEN TANTA RAMADHANIE

2602134134 – DEWA BAGAS PRATAMA

Semester Ganjil 2023/2024

MALANG

LEMBAR PERSETUJUAN PROYEK AKHIR

Nama Aplikasi

COMP6364 – Object Oriented Programming
KELAS LB20-LEC

Semester Ganjil 2023/2024

Laporan akhir proyek ini adalah benar karya kami :

IQBAL RAFI M.

2602134595

DEWA BAGAS P.

2602134134

Malang, 18 Desember 2023

AUDRIC NAGATA

2602090435

MADEN TANTA R.

2602106894

(Dio Saputra Kudori)

D6910

BAB I

LATAR BELAKANG

Dalam konteks sistem kesehatan di Indonesia, rumah sakit seringkali dihadapkan pada tantangan efisiensi dan pelayanan yang optimal kepada pasien. Dalam upaya meningkatkan kualitas pelayanan, aplikasi Health QueuePlus dirancang khusus untuk mengatasi masalah reservasi konsultasi dengan dokter di rumah sakit.

Rumah sakit di Indonesia sering mengalami antrean panjang dan kesulitan dalam pengelolaan jadwal dokter secara manual. Situasi ini sering menjadi hambatan bagi pasien yang ingin melakukan reservasi dan mencari informasi terkait jadwal dokter. Dengan adanya Health QueuePlus, diharapkan dapat mengoptimalkan proses reservasi, mengurangi waktu tunggu pasien, dan memberikan kemudahan akses bagi masyarakat untuk melakukan reservasi secara mandiri.

Selain itu, aplikasi ini tidak hanya memberikan manfaat bagi pasien, tetapi juga bagi dokter. Dokter dapat dengan mudah mengelola jadwal konsultasi mereka, menghindari bentrok jadwal, dan meningkatkan efisiensi waktu. Peningkatan transparansi antara dokter dan pasien juga menjadi tujuan utama aplikasi ini, di mana pasien dapat melihat dengan jelas jadwal dokter yang tersedia, sementara dokter dapat mengakses informasi tentang pasien yang sudah melakukan reservasi.

Penggunaan Health QueuePlus juga mencerminkan adaptasi rumah sakit terhadap perkembangan teknologi dalam pengelolaan kesehatan. Dengan memanfaatkan teknologi aplikasi, rumah sakit dapat memberikan layanan kesehatan yang lebih modern, terintegrasi, dan responsif terhadap kebutuhan pasien. Keseluruhan, Health QueuePlus diharapkan dapat menciptakan pengalaman pelayanan kesehatan yang lebih baik dan efisien di rumah sakit di Indonesia.

BAB II

LANDASAN TEORI

2.1 JAVA

2.1.1 Java adalah bahasa pemrograman dan platform komputasi pertama kali dirilis oleh Sun Microsystems pada tahun 1995. Java merupakan teknologi yang mendasari kekuatan program untuk utilitas, permainan, dan aplikasi bisnis. Java berjalan pada lebih dari 850 juta komputer pribadi di seluruh dunia, dan pada miliaran perangkat di seluruh dunia, termasuk ponsel dan perangkat TV (Rosihan, 2020).

2.2 CLASS

2.2.1 Class adalah prototype, atau blueprint, atau rancangan yang mendefinisikan variabel dan method-method pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan di jalankan, di dalamnya berisi atribut / type data dan method untuk menjalankan suatu program.

Class merupakan suatu blueprint atau cetakan untuk menciptakan suatu instan dari object. class juga merupakan grup suatu object dengan kemiripan attributes/properties, behaviour dan relasi ke object lain. Contoh : Class Person, Vehicle, Tree, Fruit dan lain-lain (Winardi, 2016).

2.3 CONSTRUCTOR

2.3.1 Constructor adalah method khusus yang akan dieksekusi pada saat pembuatan objek (instance). Biasanya method ini digunakan untuk inisialisasi atau mempersiapkan data untuk objek.

2.4 OVERLOADING OPERATOR

2.4.1 overloading operator adalah salah satu fitur pada java yang memungkinkan kita mendirikan dua atau lebih funcion dengan identitas yang sama selama memiliki keunikan pada funcion parameter

2.5 INHERITANCE

2.5.1 **Inheritance** atau *Pewarisan/Penurunan* adalah konsep pemrograman dimana sebuah *class* dapat '*menurunkan*' ***property*** dan ***method*** yang dimilikinya kepada ***class*** lain.

2.6 DATABASE

2.6.1 **Database** adalah kumpulan data terstruktur yang disimpan secara sistematis dalam suatu sistem komputer. Secara formal, sebuah database terdiri dari satu atau lebih tabel yang terdiri

dari baris dan kolom. Setiap kolom dalam tabel mewakili atribut atau field, sedangkan setiap baris merepresentasikan satu set data atau catatan.

Hubungan antar tabel dapat ditentukan melalui kunci atau kunci asing yang mengaitkan nilai dalam satu tabel dengan nilai dalam tabel lainnya. Database dirancang untuk menyimpan, mengelola, dan mengakses data dengan efisien, serta mendukung operasi seperti penyisipan, pembaruan, dan penghapusan data. Model data yang digunakan untuk mendefinisikan struktur database dapat berupa model relasional, hierarki, jaringan, atau objek, tergantung pada kebutuhan dan kompleksitas aplikasi yang hendak diimplementasikan.

Sistem manajemen basis data (DBMS) berperan dalam memfasilitasi interaksi antara pengguna atau aplikasi dengan database, menyediakan antarmuka untuk mengelola struktur dan isi data, serta menjaga integritas dan keamanan data secara keseluruhan. Database digunakan luas dalam berbagai bidang seperti bisnis, ilmu pengetahuan, pemerintahan, dan sektor lainnya untuk mendukung pengambilan keputusan, analisis, dan penyimpanan informasi secara efisien.

BAB III

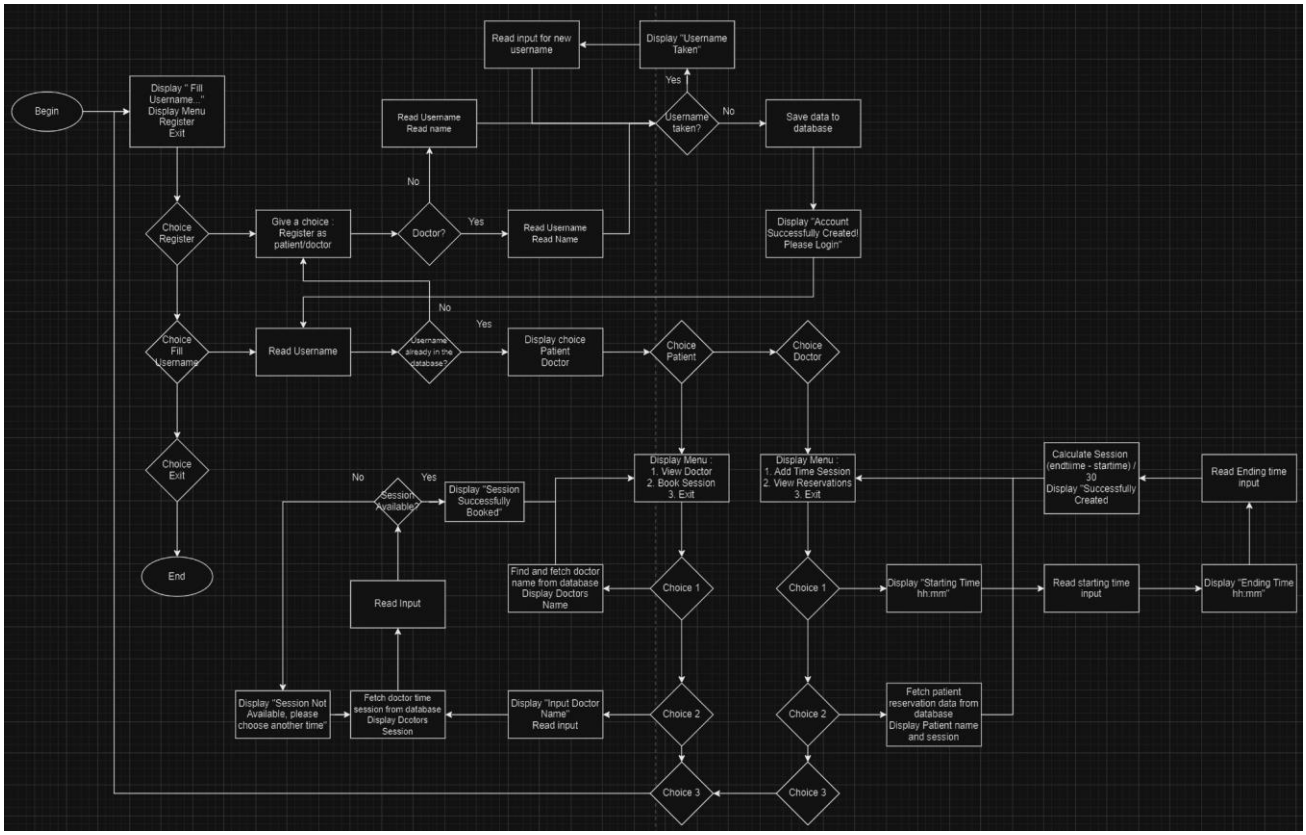
DEFINISI PROGRAM

3.1 Definisi Program

HealthQueue+ merupakan sebuah program yang ditujukan untuk sektor rumah sakit. Aplikasi ini dirancang untuk memperbaiki hubungan antara dokter dan pasien di bagian reservasi untuk konsultasi. Dalam program ini, pasien memiliki kemampuan untuk melakukan registrasi sendiri dengan memasukkan nama dan nickname, sehingga mereka dapat login sebagai pasien. Pasien kemudian diberikan dua pilihan, yaitu melihat nama dan jadwal dokter yang tersedia dalam bentuk sesi per 30 menit, serta melakukan reservasi untuk bertemu dengan dokter pilihan mereka.

Pada sisi dokter, mereka juga dapat login menggunakan nickname mereka dan memiliki dua menu utama. Pertama, mereka dapat mengatur jadwal konsultasi yang tersedia. Kedua, mereka dapat melihat daftar nama pasien yang telah melakukan reservasi, beserta jadwal sesi yang akan digunakan oleh masing-masing pasien yang sudah melakukan reservasi.

3.2 FlowChart



Full Resolution Image :

[Flowchart Finpro OOP FINAL.png]

3.3 Source Code Main Program

```
package base;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Scanner;

import java.sql.Connection;


public class HospitalReservationApp {

    private static Connection conn;

    public HospitalReservationApp(){

        conn = SQLConnector.getConnection();

    }


    private static final Map<String, Doctor> doctors = new HashMap<>();

    private static final Map<String, Patient> patients = new HashMap<>();

    private static final Scanner scanner = new Scanner(System.in);


    public static void main(String[] args) {

        initializeSampleData();

        while (true) {

            System.out.println("\n--- Main Menu ---");

            System.out.println("1. Login");
```

```
        System.out.println("2. Register");
        System.out.println("3. Exit");
        System.out.print("Choose an option: ");
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                login();
                break;
            case 2:
                register();
                break;
            case 3:
                System.out.println("Exiting the application. Goodbye!");
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}

private static void login() {
    System.out.println("\n--- Login ---");
    System.out.print("Enter your role (doctor/patient): ");
    String role = scanner.next();
    if ("doctor".equalsIgnoreCase(role)) {
```

```
            doctorLogin();
        } else if ("patient".equalsIgnoreCase(role)) {
            patientLogin();
        } else {
            System.out.println("Invalid role. Please try again.");
        }
    }

    private static void doctorLogin() {
        System.out.print("Enter your username: ");
        String username = scanner.next();
        Doctor doctor = doctors.get(username);
        if (doctor != null) {
            System.out.println("Login successful, Dr. " + doctor.getName());
            doctorMenu(doctor);
        } else {
            System.out.println("Please register yourself.");
        }
    }

    private static void patientLogin() {
        System.out.print("Enter your username: ");
        String username = scanner.next();
        Patient patient = patients.get(username);
        if (patient != null) {
```

```
            System.out.println("Login successful, " + patient.getName());
            patientMenu(patient);
        } else {
            System.out.println("Please register yourself.");
        }
    }

    private static void register() {
        System.out.println("\n--- Registration ---");
        System.out.print("Enter your role (doctor/patient): ");
        String role = scanner.next();
        System.out.print("Enter your username: ");
        String username = scanner.next();
        if ("doctor".equalsIgnoreCase(role)) {
            if (!doctors.containsKey(username)) {
                System.out.print("Enter your name: ");
                String name = scanner.next();
                doctors.put(username, new Doctor(name));
                System.out.println("Registration successful, Dr. " + name + "!");
            } else {
                System.out.println("Username already exists. Please choose a different username.");
            }
        } else if ("patient".equalsIgnoreCase(role)) {
            if (!patients.containsKey(username)) {
                System.out.print("Enter your name: ");
```

```
            String name = scanner.next();
            patients.put(username, new Patient(name));
            System.out.println("Registration successful, " + name + "!");
        } else {
            System.out.println("Username already exists. Please choose a different username.");
        }
    } else {
        System.out.println("Invalid role. Please try again.");
    }
}

private static void doctorMenu(Doctor doctor) {
    while (true) {
        System.out.println("\n--- Doctor Menu ---");
        System.out.println("1. Set Availability");
        System.out.println("2. View Reservations");
        System.out.println("3. Exit");
        System.out.print("Choose an option: ");
        int choice = scanner.nextInt();

        switch (choice) {
            case 1:
                DoctorMenu.setAvailability(doctor);
                break;
            case 2:
```

```
                DoctorMenu.viewReservations(doctor);
                break;
            case 3:
                System.out.println("Exiting Doctor Menu.");
                return;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}

private static void patientMenu(Patient patient) {
    while (true) {
        System.out.println("\n--- Patient Menu ---");
        System.out.println("1. View Doctors");
        System.out.println("2. Make Reservation");
        System.out.println("3. Exit");
        System.out.print("Choose an option: ");
        int choice = scanner.nextInt();
       switch (choice) {
            case 1:
                PatientMenu.viewDoctors(doctors);
                break;
            case 2:
                PatientMenu.makeReservation(patient, doctors);
```

```
                break;
            case 3:
                System.out.println("Exiting Patient Menu.");
                return;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}

// Sample Data Initialization
private static void initializeSampleData() {
    doctors.put("DrSmith", new Doctor("Dr. Smith"));
    doctors.put("DrJohnson", new Doctor("Dr. Johnson"));
    patients.put("JohnDoe", new Patient("John Doe"));
    patients.put("JaneDoe", new Patient("Jane Doe"));
}
}

class Human {
    public String name;
}

class Doctor extends Human{
    private List<String> availability;
```

```
    private List<String> reservations;
    public Doctor(String name) {
        this.name = name;
        this.availability = new ArrayList<>();
        this.reservations = new ArrayList<>();
    }

    public String getName() {
        return name;
    }
    public List<String> getAvailability() {
        return availability;
    }
    public List<String> getReservations() {
        return reservations;
    }

    public void setAvailability(List<String> availability) {
        this.availability = availability;
    }
}

class Patient extends Human{
```

```
    public Patient(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
}

class DoctorMenu {
    public static void setAvailability(Doctor doctor) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter available start time (hh:mm): ");
        String startTime = scanner.next();
        System.out.print("Enter available end time (hh:mm): ");
        String endTime = scanner.next();
        // Assume each session is 30 minutes
        int sessions = calculateSessions(startTime, endTime);
        List<String> availability = new ArrayList<>();
        for (int i = 0; i < sessions; i++) {
            availability.add(addMinutes(startTime, i * 30));
        }
        doctor.setAvailability(availability);
        System.out.println("Availability set successfully.");
    }
```

```
public static void viewReservations(Doctor doctor) {
    System.out.println("\n--- View Reservations ---");
    System.out.printf("%-20s %-20s\n", "Patient Name", "Reserved Session");
    for (String reservation : doctor.getReservations()) {
        System.out.println(reservation);
    }
    if (doctor.getReservations().isEmpty()) {
        System.out.println("No reservations yet.");
    }
}

private static int calculateSessions(String startTime, String endTime) {
    int startMinutes = getMinutes(startTime);
    int endMinutes = getMinutes(endTime);
    return (endMinutes - startMinutes) / 30;
}
private static int getMinutes(String time) {
    String[] parts = time.split(":");
    int hours = Integer.parseInt(parts[0]);
    int minutes = Integer.parseInt(parts[1]);
    return hours * 60 + minutes;
}
private static String addMinutes(String time, int minutesToAdd) {
    String[] parts = time.split(":");
    int hours = Integer.parseInt(parts[0]);
```

```
        int minutes = Integer.parseInt(parts[1]);
        int totalMinutes = hours * 60 + minutes + minutesToAdd;
        return String.format("%02d:%02d", totalMinutes / 60, totalMinutes % 60);
    }
}

class PatientMenu {
    public static void viewDoctors(Map<String, Doctor> doctors) {
        System.out.println("\n--- View Doctors ---");
        for (String doctorName : doctors.keySet()) {
            System.out.println(doctorName);
        }
    }

    public static void makeReservation(Patient patient, Map<String, Doctor> doctors) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter doctor's name: ");
        String doctorName = scanner.nextLine();
        Doctor doctor = doctors.get(doctorName);
        if (doctor != null) {
            System.out.println("Doctor: " + doctor.getName());
            System.out.println("Availability: " + doctor.getAvailability());
            System.out.print("Choose a session: ");
            String session = scanner.nextLine();
            // Update doctor's reservations or handle reservation as needed
```

```
            doctor.getReservations().add(patient.getName() + " - " + session);



            System.out.println("Reservation made successfully!");
        } else {
            System.out.println("Doctor not found.");
        }
    }
}
```

3.3 Login Page

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package base;


import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;


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


    private final Connection conn;
```

```
/**
 * Creates new form LOGIN
 */
public login() {
    initComponents();
    conn = SQLConnector.getConnection();
    showDoctor();
    showPatient();
}

 public void showDoctor() {
    try {
        String sql = "SELECT username, name FROM doctors WHERE username = ?";
        PreparedStatement st = conn.prepareStatement(sql);
        st.setString(1, "Input your name here");
        ResultSet rs = st.executeQuery();

        if (rs.next()) {
            String name = rs.getString("name");
            String username = rs.getString("username");
            USERNAME.setText(name);
        }
    } catch (SQLException ex) {
        Logger.getLogger(HospitalReservationApp.class.getName()).log(Level.SEVERE, null, ex);
    }
```

```
}

public void showPatient() {
    try {
        String sql = "SELECT username, name FROM patients WHERE username = ?";
        PreparedStatement st = conn.prepareStatement(sql);
        st.setString(1, "Input your name here");
        ResultSet rs = st.executeQuery();

        if (rs.next()) {
            String name = rs.getString("name");
            String username = rs.getString("username");
            USERNAME.setText(name);
        }
    } catch (SQLException ex) {
        Logger.getLogger(HospitalReservationApp.class.getName()).log(Level.SEVERE, null, ex);
    }
}

/**
 * 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.
 */
@SuppressWarnings("unchecked")
```

```
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        USERNAME = new javax.swing.JTextField();
        PATIENT = new javax.swing.JButton();
        DOCTOR = new javax.swing.JButton();
        REGISTER = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        USERNAME.setText("Username");
        USERNAME.setBorder(null);
        USERNAME.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                USERNAMEActionPerformed(evt);
            }
        });
        getContentPane().add(USERNAME, new org.netbeans.lib.awtextra.AbsoluteConstraints(110,
430, 330, 40));

        PATIENT.setBackground(new java.awt.Color(80, 39, 121));
        PATIENT.setForeground(new java.awt.Color(255, 255, 255));
        PATIENT.setText("Patient");
```

```
        PATIENT.setBorder(null);
        PATIENT.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                PATIENTActionPerformed(evt);
            }
        });
        getContentPane().add(PATIENT, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 600,
330, 40));

        DOCTOR.setBackground(new java.awt.Color(80, 39, 121));
        DOCTOR.setForeground(new java.awt.Color(255, 255, 255));
        DOCTOR.setText("Doctor");
        DOCTOR.setBorder(null);
        DOCTOR.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                DOCTORActionPerformed(evt);
            }
        });
        getContentPane().add(DOCTOR, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 670,
330, 40));

        REGISTER.setBackground(new java.awt.Color(173, 83, 166));
        REGISTER.setForeground(new java.awt.Color(255, 255, 255));
        REGISTER.setText("Register");
        REGISTER.setBorder(null);
        REGISTER.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            REGISTERActionPerformed(evt);
        }
    });
    getContentPane().add(REGISTER, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 760, 140, 30));

    jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/uiuxfix/2.png"))); // NOI18N
    jLabel1.setText("jLabel1");
    getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 80, 1392, -1));

    pack();
}// </editor-fold>

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

private void PATIENTActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String sql = "SELECT COUNT(*) FROM patients WHERE username = ?";
        PreparedStatement st = conn.prepareStatement(sql);
        String username = USERNAME.getText();
        st.setString(1, username);
```

```
        ResultSet resultSet = st.executeQuery();


        resultSet.next();
        int count = resultSet.getInt(1);


        if (count > 0) {
            JOptionPane.showMessageDialog(null, "Login Successful!");
            pagepatient pagepat = new pagepatient();
            pagepat.setVisible(true);
        } else{
            JOptionPane.showMessageDialog(null, "Username doesn't exists.");
            login Log = new login();
            Log.setVisible(true);
        }
    } catch (SQLException ex) {
        Logger.getLogger(HospitalReservationApp.class.getName()).log(Level.SEVERE, null, ex);
    }
}


private void REGISTERActionPerformed(java.awt.event.ActionEvent evt) {
    Register register = new Register();
    register.setVisible(true);
}


private void DOCTORActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    try {
        String sql = "SELECT COUNT(*) FROM doctors WHERE username = ?";
        PreparedStatement st = conn.prepareStatement(sql);
        String username = USERNAME.getText();
        st.setString(1, username);
        ResultSet resultSet = st.executeQuery();

        resultSet.next();
        int count = resultSet.getInt(1);

        if (count > 0) {
            JOptionPane.showMessageDialog(null, "Login Successful!");
            doctorpage Doc = new doctorpage();
            Doc.setVisible(true);;
        } else{
            JOptionPane.showMessageDialog(null, "Username doesn't exists.");
            login Log = new login();
            Log.setVisible(true);
        }
    } catch (SQLException ex) {
        Logger.getLogger(HospitalReservationApp.class.getName()).log(Level.SEVERE, null, ex);
    }
}

/**
```

```
     * @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(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
```

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

java.util.logging.Logger.getLogger(login.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 login().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton DOCTOR;
    private javax.swing.JButton PATIENT;
    private javax.swing.JButton REGISTER;
    private javax.swing.JTextField USERNAME;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

3.4 Register Page

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package base;


import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;


/**
 *
 * @author ASUS TUF
 */
public class Register extends javax.swing.JFrame {
    private static Connection conn;
    /**
     * Creates new form Register
     */
    public Register() {
```

```
    initComponents();
    conn = SQLConnector.getConnection();
    showDoctor();
    showPatient();
}

 public void showDoctor() {
    try {
        String sql = "SELECT username, name FROM doctors WHERE username = ?";
        PreparedStatement st = conn.prepareStatement(sql);
        st.setString(1, "Input your name here");
        ResultSet rs = st.executeQuery();

        if (rs.next()) {
            String name = rs.getString("name");
            String username = rs.getString("username");
            USERNAME.setText(name);
            NAME.setText(username);
        }
    } catch (SQLException ex) {
        Logger.getLogger(HospitalReservationApp.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void showPatient() {
```

```
    try {
        String sql = "SELECT username, name FROM patients WHERE username = ?";
        PreparedStatement st = conn.prepareStatement(sql);
        st.setString(1, "Input your name here");
        ResultSet rs = st.executeQuery();


        if (rs.next()) {
            String name = rs.getString("name");
            String username = rs.getString("username");
            USERNAME.setText(name);
            NAME.setText(username);
        }
    } catch (SQLException ex) {
        Logger.getLogger(HospitalReservationApp.class.getName()).log(Level.SEVERE, null, ex);
    }
}


/**
 * 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.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {
```

```
USERNAME = new javax.swing.JTextField();
NAME = new javax.swing.JTextField();
PATIENT = new javax.swing.JButton();
DOCTOR = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

USERNAME.setText("Username");
USERNAME.setBorder(null);

NAME.setText("Name");
NAME.setBorder(null);
NAME.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        NAMEActionPerformed(evt);
    }
});

PATIENT.setBackground(new java.awt.Color(80, 39, 121));
PATIENT.setForeground(new java.awt.Color(255, 255, 255));
PATIENT.setText("Patient");
PATIENT.setBorder(null);
PATIENT.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            PATIENTActionPerformed(evt);
        }
    });

    DOCTOR.setBackground(new java.awt.Color(80, 39, 121));
    DOCTOR.setForeground(new java.awt.Color(255, 255, 255));
    DOCTOR.setText("Doctor");
    DOCTOR.setBorder(null);
    DOCTOR.setBorderPainted(false);
    DOCTOR.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            DOCTORActionPerformed(evt);
        }
    });

    jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/uiuxfix/1.png"))); //
NOI18N
    jLabel1.setText("jLabel1");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
```

```
            .addGap(180, 180, 180)
            .addComponent(USERNAME, javax.swing.GroupLayout.PREFERRED_SIZE, 330, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(180, 180, 180)
            .addComponent(NAME, javax.swing.GroupLayout.PREFERRED_SIZE, 330, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(180, 180, 180)
            .addComponent(PATIENT, javax.swing.GroupLayout.PREFERRED_SIZE, 340, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(190, 190, 190)
            .addComponent(DOCTOR, javax.swing.GroupLayout.PREFERRED_SIZE, 320, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(77, 77, 77)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 1406, javax.swing.GroupLayout.PREFERRED_SIZE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(310, 310, 310)
            .addComponent(USERNAME, javax.swing.GroupLayout.PREFERRED_SIZE, 30, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(50, 50, 50)
```

```
            .addComponent(NAME, javax.swing.GroupLayout.PREFERRED_SIZE, 30, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(250, 250, 250)
            .addComponent(PATIENT, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(740, 740, 740)
            .addComponent(DOCTOR, javax.swing.GroupLayout.PREFERRED_SIZE, 50, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(120, 120, 120)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 680, javax.swing.GroupLayout.PREFERRED_SIZE))
    );

    pack();
}// </editor-fold>

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

private void PATIENTActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String sql = "INSERT INTO patients(name, username) VALUES (?, ?)";
        PreparedStatement st = conn.prepareStatement(sql);
```

```
        String name = NAME.getText();
        String username = USERNAME.getText();
        st.setString(1, name);
        st.setString(2, username);
        st.executeUpdate();
    } catch (SQLException ex) {
        Logger.getLogger(HospitalReservationApp.class.getName()).log(Level.SEVERE, null, ex);
    }
    login Log = new login();
    Log.setVisible(true);
}

private void DOCTORActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String sql = "INSERT INTO doctors(name, username) VALUES (?, ?)";
        PreparedStatement st = conn.prepareStatement(sql);
        String name = NAME.getText();
        String username = USERNAME.getText();
        st.setString(1, name);
        st.setString(2, username);
        st.executeUpdate();
    } catch (SQLException ex) {
        Logger.getLogger(HospitalReservationApp.class.getName()).log(Level.SEVERE, null, ex);
    }
    login Log = new login();
```

```
        Log.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(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

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

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

java.util.logging.Logger.getLogger(Register.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 Register().setVisible(true);
            }
        });
    }


    // Variables declaration - do not modify
    private javax.swing.JButton DOCTOR;
    private javax.swing.JTextField NAME;
    private javax.swing.JButton PATIENT;
    private javax.swing.JTextField USERNAME;
```

```
    private javax.swing.JLabel jLabel1;

    // End of variables declaration

}
```

3.5 Laman Pasien

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package base;


/**
 *
 * @author ASUS TUF
 */
public class pagepatient extends javax.swing.JFrame {


    /**
     * Creates new form pagepatient
     */
    public pagepatient() {
        initComponents();
    }


    /**
     * 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.
```

```
    */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        BOOKING = new javax.swing.JButton();
        JADWAL = new javax.swing.JButton();
        KEMBALI = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        BOOKING.setBackground(new java.awt.Color(69, 143, 246));
        BOOKING.setForeground(new java.awt.Color(255, 255, 255));
        BOOKING.setText("Booking Sekarang");
        BOOKING.setBorder(null);
        BOOKING.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BOOKINGActionPerformed(evt);
            }
        });
        getContentPane().add(BOOKING, new org.netbeans.lib.awtextra.AbsoluteConstraints(170,
750, 120, 30));
```

```
        JADWAL.setBackground(new java.awt.Color(69, 143, 246));
        JADWAL.setForeground(new java.awt.Color(255, 255, 255));
        JADWAL.setText("Lihat Jadwal");
        JADWAL.setBorder(null);
        JADWAL.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                JADWALActionPerformed(evt);
            }
        });
        getContentPane().add(JADWAL, new org.netbeans.lib.awtextra.AbsoluteConstraints(730, 740,
140, 30));


        KEMBALI.setText("Kembali");
        KEMBALI.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                KEMBALIActionPerformed(evt);
            }
        });
        getContentPane().add(KEMBALI, new org.netbeans.lib.awtextra.AbsoluteConstraints(110,
140, -1, -1));


        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/uiuxfix/Frame
3.png"))); // NOI18N
        jLabel1.setText("jLabel1");
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(81, 119,
1233, 680));
```

```
    pack();
}// </editor-fold>

private void BOOKINGActionPerformed(java.awt.event.ActionEvent evt) {
    bookingform Book = new bookingform();
    Book.setVisible(true);
}

private void JADWALActionPerformed(java.awt.event.ActionEvent evt) {
    listdoctor List = new listdoctor();
    List.setVisible(true);
}

private void KEMBALIActionPerformed(java.awt.event.ActionEvent evt) {
    login Log = new login();
    Log.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(pagepatient.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(pagepatient.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(pagepatient.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(pagepatient.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 pagepatient().setVisible(true);
            }
        });
    }


    // Variables declaration - do not modify
    private javax.swing.JButton BOOKING;
    private javax.swing.JButton JADWAL;
    private javax.swing.JButton KEMBALI;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

3.6 Laman Booking

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package base;


/**
 *
 * @author ASUS TUF
 */
public class bookingform extends javax.swing.JFrame {


    /**
     * Creates new form bookingform
     */
    public bookingform() {
        initComponents();
    }


    /**
     * 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.
```

```
    */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton2 = new javax.swing.JButton();
        jTextField1 = new javax.swing.JTextField();
        jComboBox1 = new javax.swing.JComboBox<>();
        BOOKING = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jButton2.setBackground(new java.awt.Color(217, 217, 217));
        jButton2.setText("Doctor Name");
        jButton2.setBorder(null);
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 170,
620, 60));
```

```
        jTextField1.setBackground(new java.awt.Color(217, 217, 217));
        jTextField1.setBorder(null);
        getContentPane().add(jTextField1, new org.netbeans.lib.awtextra.AbsoluteConstraints(1380, 280, 80, 30));

        jComboBox1.setBackground(new java.awt.Color(217, 217, 217));
        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
        jComboBox1.setBorder(null);
        jComboBox1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jComboBox1ActionPerformed(evt);
            }
        });
        getContentPane().add(jComboBox1, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 270, 620, 50));

        BOOKING.setBackground(new java.awt.Color(0, 0, 0));
        BOOKING.setForeground(new java.awt.Color(255, 255, 255));
        BOOKING.setText("Booking Sekarang");
        BOOKING.setBorder(null);
        BOOKING.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BOOKINGActionPerformed(evt);
            }
        });
```

```
        getContentPane().add(BOOKING, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 400, 270, 50));


        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/uiuxfix/bookingformfixx.png"))); // NOI18N
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(-121, 0, 2660, -1));


        pack();
    }// </editor-fold>

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void BOOKINGActionPerformed(java.awt.event.ActionEvent evt) {
        pagepatient pagepat = new pagepatient();
        pagepat.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(bookingform.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(bookingform.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(bookingform.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

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

    // Variables declaration - do not modify
    private javax.swing.JButton BOOKING;
    private javax.swing.JButton jButton2;
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

3.7 Laman List Dokter

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package base;


/**
 *
 * @author ASUS TUF
 */
public class listdoctor extends javax.swing.JFrame {


    /**
     * Creates new form listdoctor
     */
    public listdoctor() {
        initComponents();
    }


    /**
     * 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.
```

```
    */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        KEMBALI = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList<>();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        KEMBALI.setText("Kembali");
        KEMBALI.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                KEMBALIActionPerformed(evt);
            }
        });
        getContentPane().add(KEMBALI, new org.netbeans.lib.awtextra.AbsoluteConstraints(300,
740, 120, 40));

        jList1.setModel(new javax.swing.AbstractListModel<String>() {
            String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };
            public int getSize() { return strings.length; }
```

```
            public String getElementAt(int i) { return strings[i]; }
        });
        jScrollPane1.setViewportView(jList1);

        getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 230, 770, 410));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/uiuxfix/Contact Form 5.png"))); // NOI18N
        jLabel1.setText("Kembali");
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(196, 124, 1169, 680));

        pack();
    }// </editor-fold>

    private void KEMBALIActionPerformed(java.awt.event.ActionEvent evt) {
        pagepatient pagepat = new pagepatient();
        pagepat.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(listdoctor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(listdoctor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(listdoctor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(listdoctor.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 listdoctor().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton KEMBALI;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JList<String> jList1;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration
}
```

3.8 Laman Dokter

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package base;


/**
 *
 * @author ASUS TUF
 */
public class doctorpage extends javax.swing.JFrame {


    /**
     * Creates new form doctorpage
     */
    public doctorpage() {
        initComponents();
    }


    /**
     * 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.
```

```
    */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        JADWAL = new javax.swing.JButton();
        DAFTARPASIEN = new javax.swing.JButton();
        KEMBALI = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        JADWAL.setBackground(new java.awt.Color(69, 143, 246));
        JADWAL.setForeground(new java.awt.Color(255, 255, 255));
        JADWAL.setText("Atur Jadwal");
        JADWAL.setBorder(null);
        JADWAL.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                JADWALActionPerformed(evt);
            }
        });
        getContentPane().add(JADWAL, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 670,
120, 40));
```

```
        DAFTARPASIEN.setBackground(new java.awt.Color(69, 143, 246));
        DAFTARPASIEN.setForeground(new java.awt.Color(255, 255, 255));
        DAFTARPASIEN.setText("Lihat Sekarang");
        DAFTARPASIEN.setBorder(null);
        DAFTARPASIEN.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                DAFTARPASIENActionPerformed(evt);
            }
        });
        getContentPane().add(DAFTARPASIEN, new org.netbeans.lib.awtextra.AbsoluteConstraints(830, 660, 120, 40));

        KEMBALI.setText("Kembali");
        getContentPane().add(KEMBALI, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 50, -1, -1));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/uiuxfix/doctor page.png"))); // NOI18N
        jLabel1.setText("jLabel1");
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(172, 0, 1144, -1));

        pack();
    }// </editor-fold>

    private void JADWALActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        sceduledoktor sch = new sceduledoktor();
        sch.setVisible(true);
    }

    private void DAFTARPASIENActionPerformed(java.awt.event.ActionEvent evt) {
        ViewPatient sch = new ViewPatient();
        sch.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(doctorpage.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (InstantiationException ex) {

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

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

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


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

```
    // Variables declaration - do not modify
    private javax.swing.JButton DAFTARPASIEN;
    private javax.swing.JButton JADWAL;
    private javax.swing.JButton KEMBALI;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

3.9 Laman Set Schedule Dokter

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package base;


/**
 *
 * @author ASUS TUF
 */
public class sceduledoktor extends javax.swing.JFrame {


    /**
     * Creates new form sceduledoktor
     */
    public sceduledoktor() {
        initComponents();
    }


    /**
     * 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.
```

```
    */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        SUBMIT = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jTextField1.setText("Start");
        getContentPane().add(jTextField1, new org.netbeans.lib.awtextra.AbsoluteConstraints(300,
130, 460, 40));

        jTextField2.setText("End");
        jTextField2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField2ActionPerformed(evt);
            }
        });
        getContentPane().add(jTextField2, new org.netbeans.lib.awtextra.AbsoluteConstraints(300,
200, 460, 40));
```

```
        SUBMIT.setBackground(new java.awt.Color(0, 0, 0));
        SUBMIT.setForeground(new java.awt.Color(255, 255, 255));
        SUBMIT.setText("SUBMIT");
        SUBMIT.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                SUBMITActionPerformed(evt);
            }
        });
        getContentPane().add(SUBMIT, new org.netbeans.lib.awtextra.AbsoluteConstraints(430, 300, 200, 40));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/uiuxfix/scedule.png"))); // NOI18N
        jLabel1.setText("jLabel1");
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1194, -1));

        pack();
    }// </editor-fold>

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void SUBMITActionPerformed(java.awt.event.ActionEvent evt) {
        doctorpage Doc = new doctorpage();
```

```
        Doc.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(sceduledoktor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(sceduledoktor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(sceduledoktor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(sceduledoktor.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 sceduledoktor().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton SUBMIT;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
```

```
    // End of variables declaration

}
```

3.10 Laman List Reservasi Pasien

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package base;


/**
 *
 * @author ASUS TUF
 */
public class ViewPatient extends javax.swing.JFrame {


    /**
     * Creates new form ViewPatient
     */
    public ViewPatient() {
        initComponents();
    }


    /**
     * 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.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList<>();
        KEMBALI = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jList1.setBorder(null);
        jList1.setModel(new javax.swing.AbstractListModel<String>() {
            String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };
            public int getSize() { return strings.length; }
            public String getElementAt(int i) { return strings[i]; }
        });
        jScrollPane1.setViewportView(jList1);

        getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(110,
140, 790, 430));
```

```
        KEMBALI.setText("Kembali");
        KEMBALI.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                KEMBALIActionPerformed(evt);
            }
        });
        getContentPane().add(KEMBALI, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 660, 110, 40));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/uiuxfix/viewpatient.png"))); // NOI18N
        jLabel1.setText("jLabel1");
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, -2, -1, 770));

        pack();
    }// </editor-fold>
    private doctorpage Doc;
    private void KEMBALIActionPerformed(java.awt.event.ActionEvent evt) {
        if (Doc == null || !Doc.isVisible()) {
        // If secondFrame is not instantiated or not visible, create and show it
        Doc = new doctorpage();
        Doc.setVisible(true);
    } else {
        // If secondFrame is already visible, bring it to the front
        Doc.toFront();
```

```
    }
    }

    /**
     * @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(ViewPatient.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(ViewPatient.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(ViewPatient.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(ViewPatient.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 ViewPatient().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton KEMBALI;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JList<String> jList1;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration
```

```
}
```

BAB IV

HASIL

4.1 Database

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	doctors	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	doctor_availability	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	patients	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	reservations	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
	4 tables	Sum	2	InnoDB	utf8mb4_general_ci	96.0 KiB	0 B

- Table Doctors

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	username	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

- Table Jadwal Doctors

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	doctor_username	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	2	session_start_time	time			No	None			Change Drop More
☐	3	session_end_time	time			Yes	NULL			Change Drop More

- Table Patient

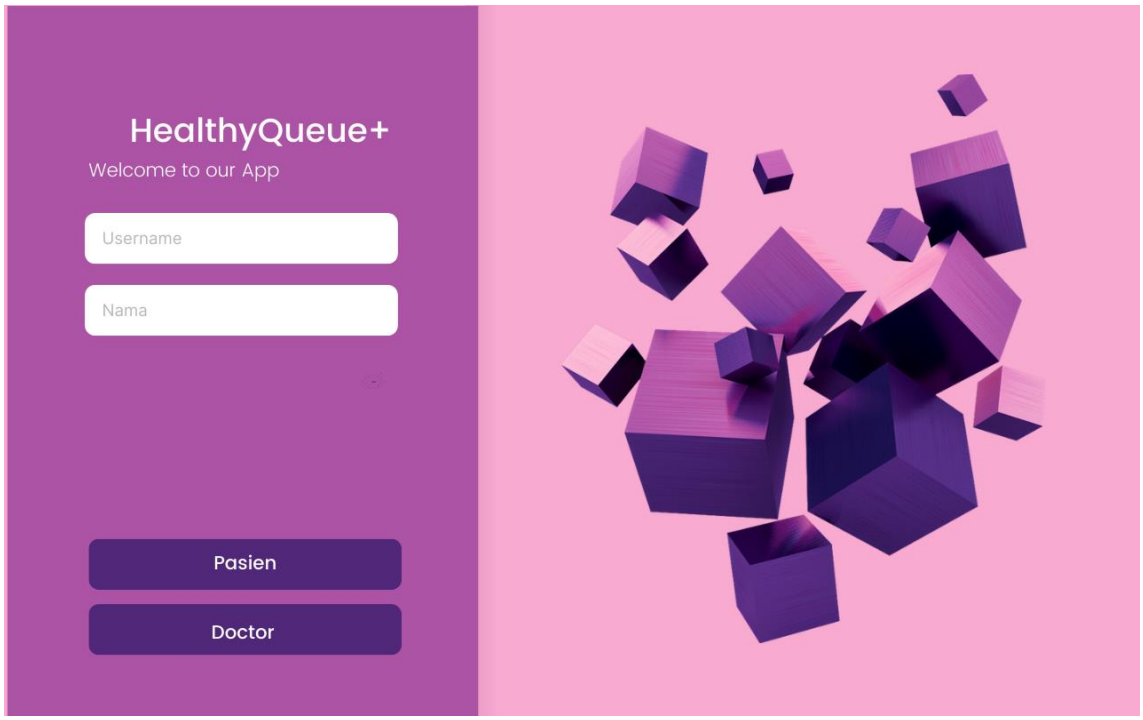
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	username	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

- Table Reservasi Doctor

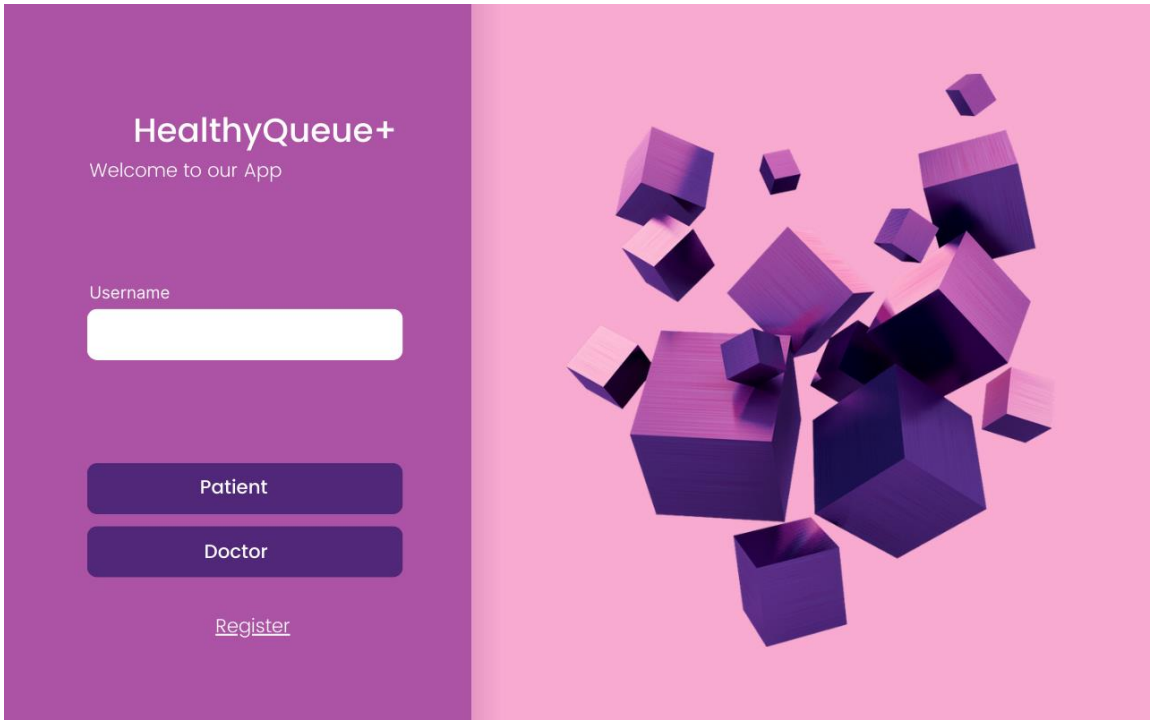
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	patient_username	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	3	doctor_username	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	4	session_start_time	time			Yes	NULL			Change Drop More

4.2 Screenshot Aplikasi

- Register page

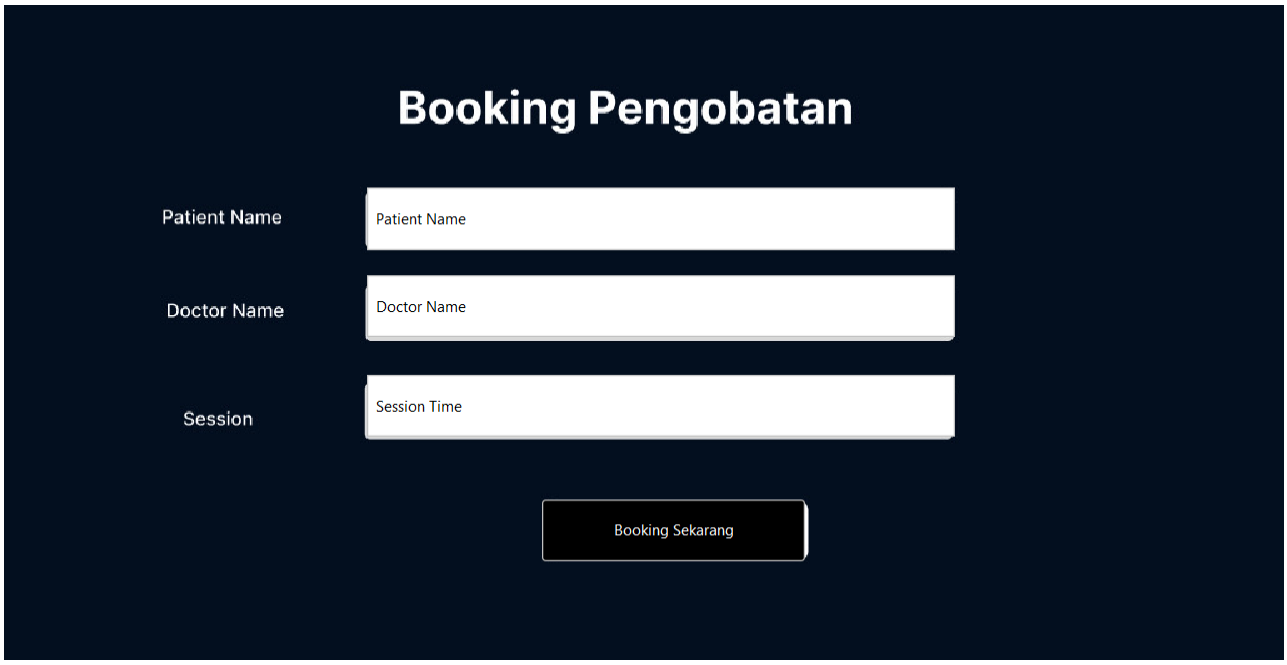


- Login page

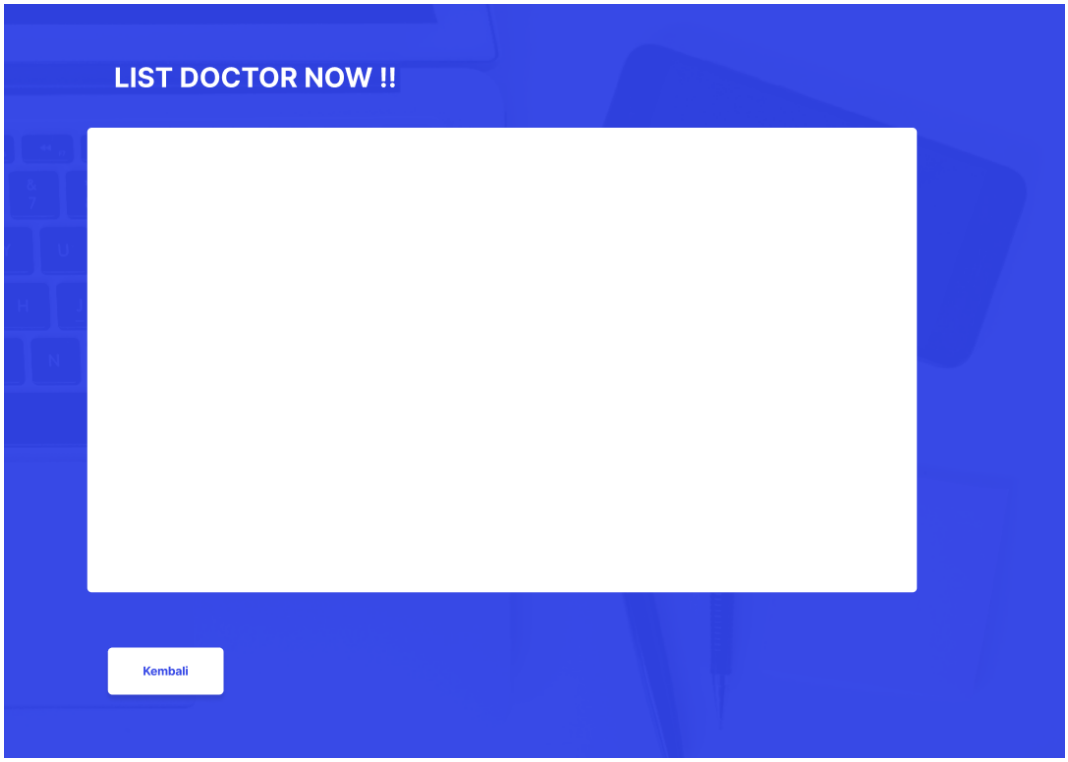


- Patient page

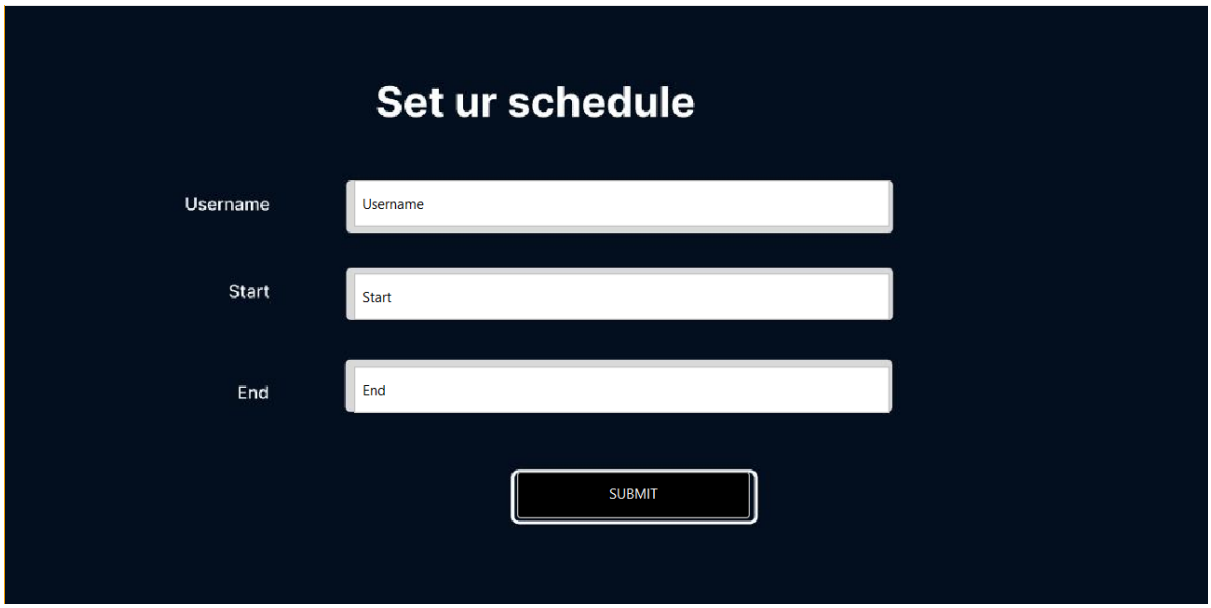
- Booking form (patient page)



- List Doctor (patient page)



- Schedule page (doctor page)



- List patient (doctor page)

DAFTAR PUSTAKA

https://d1wqtxts1xzle7.cloudfront.net/48848145/document1-libre.pdf?1473920301=&response-content-disposition=inline%3B+filename%3DPemrograman_Java.pdf&Expires=1702864989&Signature=GUAkrPspkFKz2iEcv3uHCNchT5qFKaWDqKcIVTlVGL7UB10oU1BwhdHBejvXFarkKGo9DZXnvUjdb9asrIYOreVwD9z6kvWuY7Y9RwTF0ZvJiqhyhmW8cFIkxt2zIchhwZHwN9xqheICdlVo9lNz9kZvQaoY23uz7us7SpTZk1xR5rn3L6adCbefZiJ4ODXszr~jrTEaew0mNAlP6cnge1nboBwxPMJtoGnISFJG7ACGmE4MtaPfV7K2xFL~x8W7Z55jY~I8bYjl8y6ZHC~vmfSQxSvfHeDbJGvIWuf7RRVq6J6yl5rtN6qAzwEJv3o9Y~RK9sRMdotuM0Fjjyhp0Q__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

https://student-activity.binus.ac.id/himsisfo/2016/07/pengertian-methode-class-dan-objek-dalam-oop/