

PROPOSAL

PROJECT MANAGEMENT APPLICATION

Dosen Mata Kuliah:

Alun Sujjada, S. Kom, M.T (012020037)



Anggota Kelompok:

Julhan Abdul Malik 20220040126

Amelinda Renjani 20220040163

Rayhansyah 20220040283

TEKNIK INFORMATIKA
FAKULTAS TEKNIK, DESAIN DAN KOMPUTER
UNIVERSITAS NUSA PUTRA
2024

DAFTAR ISI

DAFTAR ISI.....	i
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Tujuan	1
1.3. Manfaat.....	2
BAB II DESAIN SISTEM.....	3
2.1. Flowchart	3
2.2. Diagram ERD (Entity-Relationship Diagram).....	4
2.3. Wireframe	4
BAB III IMPLEMENTASI.....	7
3.1. Pemilihan Teknologi.....	7
3.2. Struktur kode	7
BAB IV PENUTUP	15
4.1. Kesimpulan.....	15
4.2. Saran	15

BAB I

PENDAHULUAN

1.1.Latar Belakang

Dalam era globalisasi dan digitalisasi yang semakin maju, pengelolaan proyek menjadi salah satu aspek krusial yang menentukan keberhasilan suatu organisasi atau perusahaan. Banyak organisasi menghadapi tantangan dalam mengelola proyek mereka secara efisien dan efektif. Tantangan-tantangan ini meliputi alokasi sumber daya yang tidak optimal, koordinasi tim yang kurang baik, serta pelaporan progres proyek yang tidak real-time. Kesalahan dan ketidakakuratan dalam manajemen proyek dapat mengakibatkan penundaan, peningkatan biaya, dan kualitas hasil yang menurun.

Selain itu, di tengah kompetisi bisnis yang ketat, kemampuan untuk merespon cepat terhadap perubahan dan kebutuhan pasar menjadi sangat penting. Oleh karena itu, diperlukan sebuah solusi yang dapat membantu organisasi dalam merencanakan, melaksanakan, memantau, dan mengevaluasi proyek-proyek mereka dengan lebih baik.

Aplikasi manajemen proyek berbasis Java Database merupakan solusi yang diharapkan dapat menjawab kebutuhan tersebut. Java sebagai bahasa pemrograman yang populer dan handal, dikombinasikan dengan database yang kuat, dapat menyediakan platform yang stabil dan efisien untuk manajemen proyek. Aplikasi ini dirancang untuk membantu organisasi dalam:

- Menyusun jadwal, menetapkan tugas, dan mengalokasikan sumber daya secara efektif.
- Meningkatkan koordinasi dan kolaborasi antar anggota tim melalui fitur-fitur komunikasi dan manajemen tugas.
- Memantau progres proyek secara real-time dan menyediakan laporan yang akurat dan mudah dipahami.
- Menganalisis kinerja proyek dan memberikan masukan untuk perbaikan di masa mendatang.

1.2.Tujuan

Pengembangan aplikasi manajemen proyek berbasis Java Database bertujuan untuk menyediakan solusi yang komprehensif dan terintegrasi bagi organisasi dalam mengelola proyek mereka. Adapun tujuan spesifik dari pengembangan aplikasi ini adalah sebagai berikut:

- Meningkatkan Efisiensi Pengelolaan Proyek

Membantu organisasi dalam merencanakan dan mengatur tugas-tugas proyek secara lebih efisien, juga meminimalkan waktu dan usaha yang diperlukan untuk melakukan administrasi proyek.

- Meningkatkan kolaborasi tim
- Memastikan Kepatuhan Terhadap Jadwal dan Anggaran

Membantu memastikan bahwa proyek berjalan sesuai dengan jadwal yang telah ditetapkan dan memantau penggunaan anggaran proyek secara ketat untuk mencegah pembengkakan biaya.

1.3.Manfaat

Pengembangan aplikasi manajemen proyek berbasis Java Database menawarkan berbagai manfaat yang signifikan bagi organisasi. Manfaat utama dari penggunaan aplikasi ini yaitu:

- Efisiensi Operasional yang Lebih Tinggi

Aplikasi ini membantu mengurangi pekerjaan manual dan mempercepat proses administrasi melalui otomatisasi tugas-tugas rutin. Selain itu juga membantu tim proyek dalam mengelola waktu secara lebih efektif dengan fitur penjadwalan dan pengingat otomatis.

- Peningkatan Kolaborasi Tim

Aplikasi ini membantu menyediakan platform terpusat untuk berbagi informasi, dokumen, dan update proyek sehingga seluruh tim selalu terinformasi.

- Kepatuhan terhadap Jadwal dan Anggaran

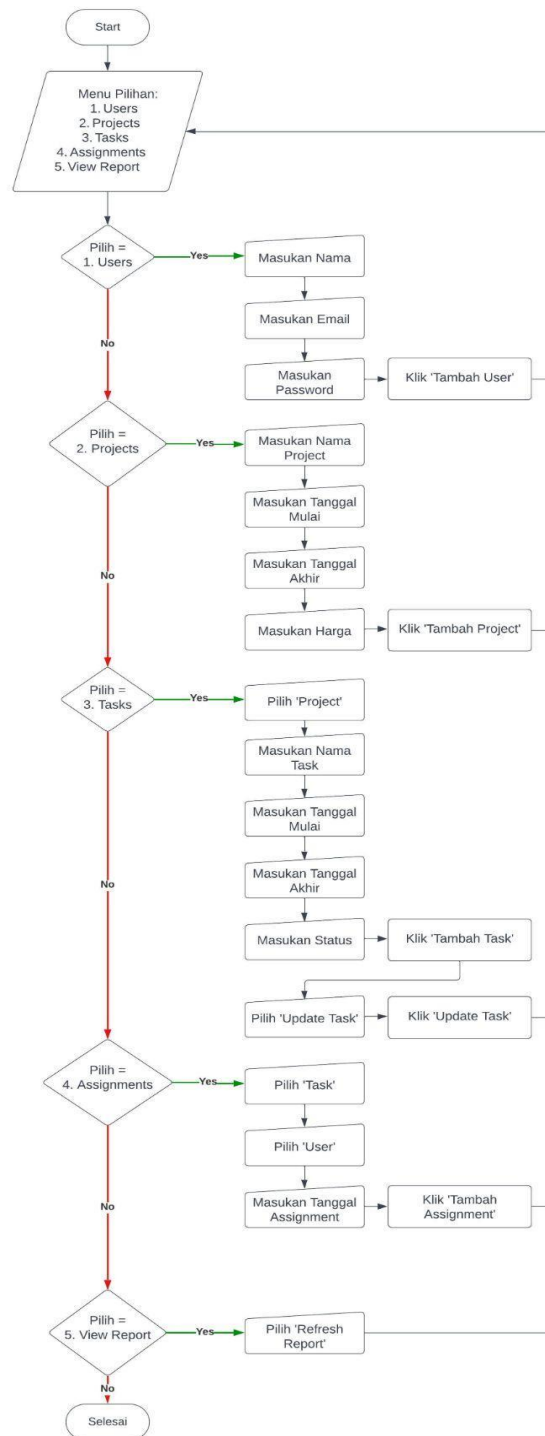
Memantau penggunaan anggaran secara ketat untuk memastikan proyek berjalan sesuai rencana finansial juga membantu memastikan proyek selesai tepat waktu dengan fitur penjadwalan yang efisien dan alat pengingat tugas.

BAB II

DESAIN SISTEM

2.1. Flowchart

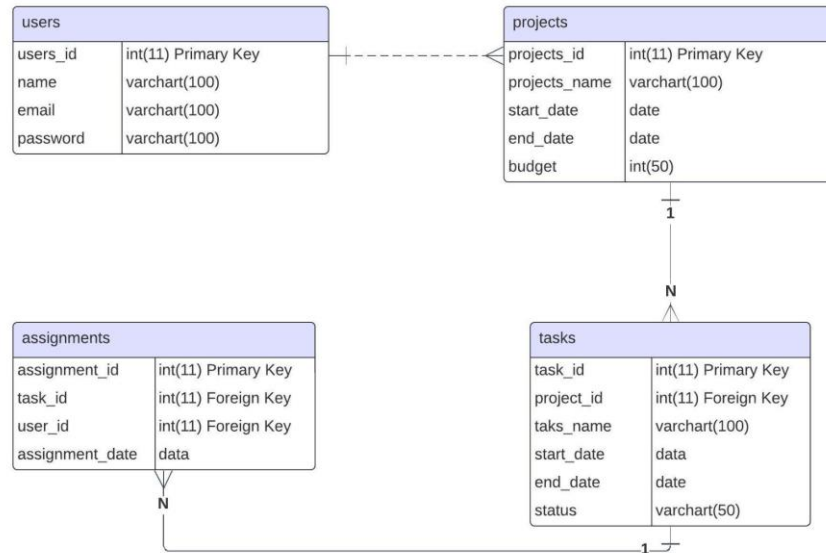
Flowchart



Gambar 2.1. Flowchart Project Management Application

2.2. Diagram ERD (Entity-Relationship Diagram)

Entity Relationship Diagram (ERD)



Gambar 2.2. Diagram ERD Project Management Application

2.3. Wireframe



Gambar 2.3. Tampilan Halaman Utama Project Managament Application



Manage Users

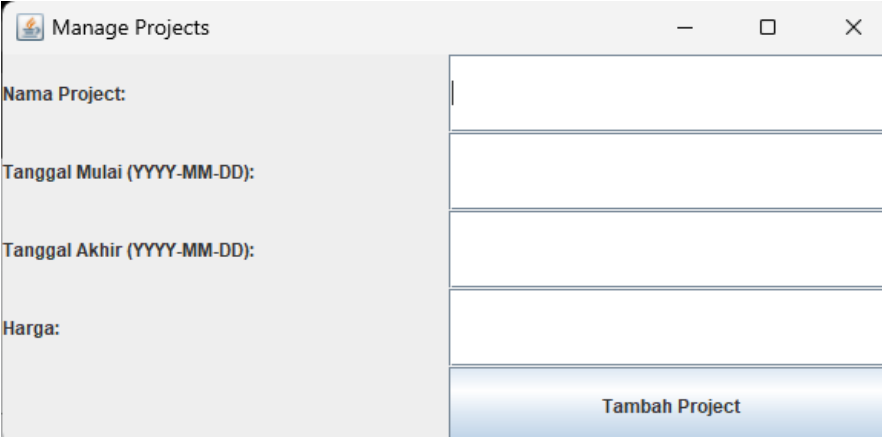
Nama:

Email:

Password:

Tambah User

Gambar 2.4. Tampilan Project Managament Application Bagian **Users**



Manage Projects

Nama Project:

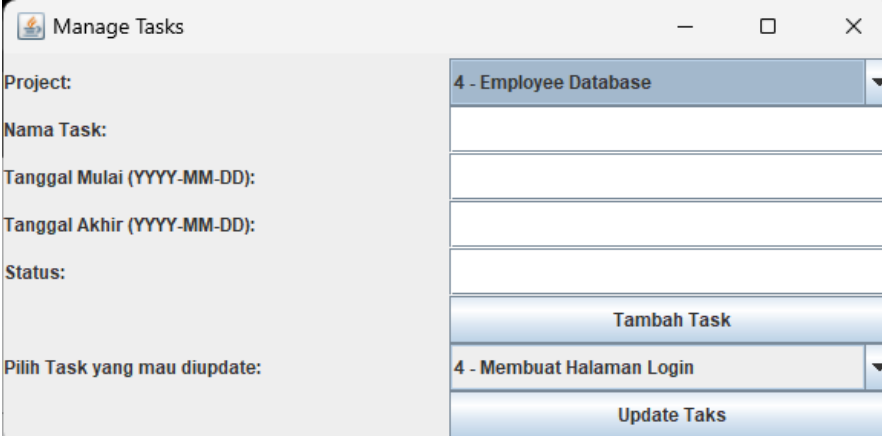
Tanggal Mulai (YYYY-MM-DD):

Tanggal Akhir (YYYY-MM-DD):

Harga:

Tambah Project

Gambar 2.5. Tampilan Project Managament Application Bagian **Projects**



Manage Tasks

Project: 4 - Employee Database

Nama Task:

Tanggal Mulai (YYYY-MM-DD):

Tanggal Akhir (YYYY-MM-DD):

Status:

Tambah Task

Pilih Task yang mau diupdate: 4 - Membuat Halaman Login

Update Taks

Gambar 2.6. Tampilan Project Managament Application Bagian **Tasks**

Manage Assignments

Task: 4 - Membuat Halaman Login

User: 5 - Julhan

Tanggal Assignment (YYYY-MM-DD):

Tambah Assignment

Gambar 2.7. Tampilan Project Management Application Bagian **Assignments**

Project Report

Nama Project	Tanggal Mulai Project	Tanggal Akhir Project	Harga	Nama Task	Tanggal Mulai Task	Tanggal Akhir Task	Status	Assigned User
Employee Database	2024-06-25	2024-06-25	500000	Membuat Halaman Login	2024-06-25	2024-06-25	Finished	Julhan
Employee Database	2024-06-25	2024-06-25	500000	Membuat Halaman Dashboard	2024-06-25	2024-06-25	Coba	Abdul
Employee Database	2024-06-25	2024-06-25	500000	Membuat Halaman Laporan	2024-06-25	2024-06-25	In Progress	Amelinda
Keuangan Apps	2024-07-01	2024-07-30	25000000	Membuat Halaman Neraca	2024-07-01	2024-07-10	In Progress	Renjani
Keuangan Apps	2024-07-01	2024-07-30	25000000	Membuat Halaman Piutang	2024-07-10	2024-07-15	In Progress	Malik
SIPP	2024-06-01	2024-07-01	10000000	Membuat SIPP Login	2024-06-01	2024-06-25	In Progressss	Malik
Employee Database	2024-06-25	2024-06-25	500000	Coba2	2023-07-01	2023-07-02	Finished	Amelinda
Employee Database	2024-06-25	2024-06-25	500000	Membuat SIPP Login	2024-06-01	2024-06-25	In Progressss	
Keuangan Apps	2024-07-01	2024-07-30	25000000	COBa	2023-07-01	2023-07-02	Coba	
Keuangan Apps	2024-07-01	2024-07-30	25000000	Coba	2024-06-22	2024-06-21	Finishedd	

Refresh Report

Gambar 2.8. Tampilan Project Management Application Bagian **View Report**

BAB III

IMPLEMENTASI

3.1. Pemilihan Teknologi

Pemilihan teknologi yang tepat sangat penting untuk memastikan aplikasi manajemen proyek dapat berfungsi dengan optimal, aman, dan dapat dikembangkan lebih lanjut. Berikut adalah teknologi yang dipilih untuk pengembangan aplikasi "Management Project Application".

a) Bahasa Pemrograman Java

Java adalah bahasa pemrograman yang kuat, aman, dan platform-independen, yang memungkinkan aplikasi berjalan di berbagai sistem operasi tanpa perubahan kode. Java mendukung pemrograman berorientasi objek (OOP), yang membantu dalam pengorganisasian kode dan memungkinkan pengembangan perangkat lunak yang scalable dan mudah dikelola. Selain itu, ekosistem yang kaya dengan banyak library dan framework yang dapat mempercepat pengembangan.

b) Basis Data MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang handal, aman, dan efisien. MySQL mendukung skala besar dengan performa tinggi, menjadikannya pilihan yang tepat untuk aplikasi yang membutuhkan penyimpanan data yang besar. Komunitas yang besar dan dokumentasi yang luas memudahkan pengembang dalam mencari solusi dan dukungan teknis. Disamping itu, MySQL mendukung integrasi yang baik dengan Java melalui JDBC (Java Database Connectivity).

c) Apache

Apache adalah server aplikasi open-source yang ringan dan mendukung Java Servlet dan JavaServer Pages (JSP). Apache menyediakan performa tinggi dan dapat diandalkan untuk menjalankan aplikasi web berbasis Java, juga mudah dikonfigurasi dan diintegrasikan dengan berbagai aplikasi Java lainnya.

3.2. Struktur kode

a) File ProjectManagementApp.java

```

ProjectManagementApp.java

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.SQLException;

public class ProjectManagementApp {
    private JFrame frame;
    private JPanel panel;

    public ProjectManagementApp() {
        frame = new JFrame("Project UAS PBO Semester 4 - Julhan, Amel");
        panel = new JPanel();
        panel.setLayout(new BorderLayout());

        JLabel titleLabel = new JLabel("- Project Management Application -", JLabel.CENTER);
        titleLabel.setFont(new Font("Lato", Font.BOLD, 22));
        panel.add(titleLabel, BorderLayout.NORTH);

        JButton manageUsersButton = new JButton("1. Users");
        JButton manageProjectsButton = new JButton("2. Projects");
        JButton manageTasksButton = new JButton("3. Tasks");
        JButton manageAssignmentsButton = new JButton("4. Assignments");
        JButton viewReportButton = new JButton("5. View Report");

        JPanel buttonPanel = new JPanel();
        buttonPanel.setLayout(new GridLayout(3, 2));
        buttonPanel.add(manageUsersButton);
        buttonPanel.add(manageProjectsButton);
        buttonPanel.add(manageTasksButton);
        buttonPanel.add(manageAssignmentsButton);
        buttonPanel.add(viewReportButton);

        panel.add(buttonPanel, BorderLayout.CENTER);

        frame.add(panel);
        frame.setSize(600, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

        manageUsersButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new UserManagement();
            }
        });

        manageProjectsButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new ProjectManagement();
            }
        });

        manageTasksButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new TaskManagement();
            }
        });

        manageAssignmentsButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new AssignmentManagement();
            }
        });

        viewReportButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new ReportManagement();
            }
        });
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new ProjectManagementApp();
            }
        });
    }
}

```

Gambar 3.1. Kode Program file ProjectManagementApp.java

b) File UserManagement.java

```

UserManagement.java

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class UserManagement extends JFrame {
    private JTextField nameField;
    private JTextField emailField;
    private JPasswordField passwordField;
    private JButton addButton;

    public UserManagement() {
        setTitle("Manage Users");
        setSize(600, 300);
        setLayout(new GridLayout(4, 2));

        JLabel nameLabel = new JLabel("Nama:");
        nameField = new JTextField();
        JLabel emailLabel = new JLabel("Email:");
        emailField = new JTextField();
        JLabel passwordLabel = new JLabel("Password:");
        passwordField = new JPasswordField();
        addButton = new JButton("Tambah User");

        add(nameLabel);
        add(nameField);
        add(emailLabel);
        add(emailField);
        add(passwordLabel);
        add(passwordField);
        add(new JLabel()); // Empty cell
        add(addButton);

        addButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                addUser();
            }
        });

        setVisible(true);
    }

    private void addUser() {
        String name = nameField.getText();
        String email = emailField.getText();
        String password = new String(passwordField.getPassword());

        try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "INSERT INTO users (name, email, password) VALUES (?, ?, ?)";
            PreparedStatement statement = connection.prepareStatement(query);
            statement.setString(1, name);
            statement.setString(2, email);
            statement.setString(3, password);
            statement.executeUpdate();

            JOptionPane.showMessageDialog(this, "User berhasil ditambahkan!");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "Error menambahkan User: " + ex.getMessage());
        }
    }
}

```

Gambar 3.2. Kode Program file UserManagement.java

d) File ProjectManagement.java

```

ProjectManagement.java

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class ProjectManagement extends JFrame {
    private JTextField projectNameField;
    private JTextField startDateField;
    private JTextField endDateField;
    private JTextField budgetField;
    private JButton addButton;

    public ProjectManagement() {
        setTitle("Manage Projects");
        setSize(600, 300);
        setLayout(new GridLayout(5, 2));

        JLabel projectNameLabel = new JLabel("Nama Project:");
        projectNameField = new JTextField();
        JLabel startDateLabel = new JLabel("Tanggal Mulai (YYYY-MM-DD):");
        startDateField = new JTextField();
        JLabel endDateLabel = new JLabel("Tanggal Akhir (YYYY-MM-DD):");
        endDateField = new JTextField();
        JLabel budgetLabel = new JLabel("Harga:");
        budgetField = new JTextField();
        addButton = new JButton("Tambah Project");

        add(projectNameLabel);
        add(projectNameField);
        add(startDateLabel);
        add(startDateField);
        add(endDateLabel);
        add(endDateField);
        add(budgetLabel);
        add(budgetField);
        add(new JLabel());
        add(addButton);

        addButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                addProject();
            }
        });

        setVisible(true);
    }

    private void addProject() {
        String projectName = projectNameField.getText();
        String startDate = startDateField.getText();
        String endDate = endDateField.getText();
        int budget = Integer.parseInt(budgetField.getText());

        try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "INSERT INTO projects (project_name, start_date, end_date, budget) VALUES (?, ?, ?, ?)";
            PreparedStatement statement = connection.prepareStatement(query);
            statement.setString(1, projectName);
            statement.setString(2, startDate);
            statement.setString(3, endDate);
            statement.setInt(4, budget);
            statement.executeUpdate();

            JOptionPane.showMessageDialog(this, "Project berhasil ditambahkan!!");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "Error menambahkan project: " + ex.getMessage());
        }
    }
}

```

Gambar 3.4. Kode Program file ProjectManagement.java

e) File AssignmentManagement.java

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class AssignmentManagement extends JFrame {
    private JComboBox<String> taskComboBox;
    private JComboBox<String> userComboBox;
    private JTextField assignedDateField;
    private JButton addButton;

    public AssignmentManagement() {
        setTitle("Manage Assignments");
        setSize(600, 300);
        setLayout(new GridLayout(4, 2));

        JLabel taskLabel = new JLabel("Task:");
        taskComboBox = new JComboBox<>();
        loadTasks();

        JLabel userLabel = new JLabel("User:");
        userComboBox = new JComboBox<>();
        loadUsers();

        JLabel assignedDateLabel = new JLabel("Tanggal Assignment (YYYY-MM-DD):");
        assignedDateField = new JTextField();
        addButton = new JButton("Tambah Assignment");

        add(taskLabel);
        add(taskComboBox);
        add(userLabel);
        add(userComboBox);
        add(assignedDateLabel);
        add(assignedDateField);
        add(new JLabel()); // Empty cell
        add(addButton);

        addButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                addAssignment();
            }
        });

        setVisible(true);
    }

    private void loadTasks() {
        try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "SELECT task_id, task_name FROM tasks";
            PreparedStatement statement = connection.prepareStatement(query);
            ResultSet resultSet = statement.executeQuery();

            while (resultSet.next()) {
                int taskId = resultSet.getInt("task_id");
                String taskName = resultSet.getString("task_name");
                taskComboBox.addItem(taskId + " - " + taskName);
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "Error loading tasks: " + ex.getMessage());
        }
    }

    private void loadUsers() {
        try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "SELECT user_id, name FROM users";
            PreparedStatement statement = connection.prepareStatement(query);
            ResultSet resultSet = statement.executeQuery();

            while (resultSet.next()) {
                int userId = resultSet.getInt("user_id");
                String name = resultSet.getString("name");
                userComboBox.addItem(userId + " - " + name);
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "Error loading users: " + ex.getMessage());
        }
    }

    private void addAssignment() {
        String selectedTask = (String) taskComboBox.getSelectedItem();
        int taskId = Integer.parseInt(selectedTask.split(" - ")[0]);
        String selectedUser = (String) userComboBox.getSelectedItem();
        int userId = Integer.parseInt(selectedUser.split(" - ")[0]);
        String assignedDate = assignedDateField.getText();

        try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "INSERT INTO assignments (task_id, user_id, assigned_date) VALUES (?, ?, ?)";
            PreparedStatement statement = connection.prepareStatement(query);
            statement.setInt(1, taskId);
            statement.setInt(2, userId);
            statement.setString(3, assignedDate);
            statement.executeUpdate();

            JOptionPane.showMessageDialog(this, "Assignment berhasil ditambahkan!!");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "Error menambahkan assignment: " + ex.getMessage());
        }
    }
}

```

Gambar 3.5. Kode Program file AssignmentManagement.java

f) File ReportManagement.java

```

ReportManagement.java

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class ReportManagement extends JFrame {
    private JTable reportTable;
    private DefaultTableModel tableModel;
    private JButton refreshButton;

    public ReportManagement() {
        setTitle("Project Report");
        setSize(800, 400);
        setLayout(new BorderLayout());

        // Kolom untuk tabel laporan
        String[] columnNames = {"Nama Project", "Tanggal Mulai Project", "Tanggal Akhir Project", "Harga",
                                "Nama Task", "Tanggal Mulai Task", "Tanggal Akhir Task",
                                "Status", "Assigned User"};

        // Model tabel
        tableModel = new DefaultTableModel(columnNames, 0);
        reportTable = new JTable(tableModel);
        JScrollPane scrollPane = new JScrollPane(reportTable);
        add(scrollPane, BorderLayout.CENTER);

        // Tombol refresh
        refreshButton = new JButton("Refresh Report");
        add(refreshButton, BorderLayout.SOUTH);

        refreshButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                loadReport();
            }
        });

        loadReport();
        setVisible(true);
    }

    private void loadReport() {
        // Bersihkan data tabel sebelumnya
        tableModel.setRowCount(0);

        try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "SELECT p.project_name, p.start_date, p.end_date, p.budget, " +
                           "t.task_name, t.start_date AS task_start, t.end_date AS task_end, " +
                           "t.status, " +
                           "u.name AS user_name " +
                           "FROM projects p " +
                           "LEFT JOIN tasks t ON p.project_id = t.project_id " +
                           "LEFT JOIN assignments a ON t.task_id = a.task_id " +
                           "LEFT JOIN users u ON a.user_id = u.user_id";

            PreparedStatement statement = connection.prepareStatement(query);
            ResultSet resultSet = statement.executeQuery();

            while (resultSet.next()) {
                String projectName = resultSet.getString("project_name");
                String projectStartDate = resultSet.getString("start_date");
                String projectEndDate = resultSet.getString("end_date");
                int projectBudget = resultSet.getInt("budget");
                String taskName = resultSet.getString("task_name");
                String taskStartDate = resultSet.getString("task_start");
                String taskEndDate = resultSet.getString("task_end");
                String taskStatus = resultSet.getString("status");
                String userName = resultSet.getString("user_name");

                // Tambahkan baris data ke tabel
                tableModel.addRow(new Object[]{projectName, projectStartDate, projectEndDate,
                    projectBudget, taskName, taskStartDate, taskEndDate,
                    taskStatus, userName});
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "Error loading report: " + ex.getMessage());
        }
    }
}

```

Gambar 3.6. Kode Program file ReportManagement.java

Link GitHub Full Source Code Project:

-> https://github.com/JulhanAbdulMalik-TI22A/Project_UAS_PBO.git

BAB IV

PENUTUP

4.1.Kesimpulan

Aplikasi "Project Management Application" berbasis Java dan MySQL yang dikembangkan dalam proyek ini berhasil memenuhi tujuan utamanya yaitu menyediakan solusi manajemen proyek yang komprehensif dan terintegrasi. Dengan menggunakan bahasa pemrograman Java dan basis data MySQL yang efisien, aplikasi ini mampu membantu organisasi dalam merencanakan, melaksanakan, dan memantau proyek secara lebih efektif. Secara keseluruhan, aplikasi ini memberikan kontribusi signifikan dalam meningkatkan efisiensi operasional dan kinerja proyek.

4.2.Saran

Untuk pengembangan lebih lanjut, aplikasi ini dapat ditingkatkan dengan integrasi alat kolaborasi eksternal dan kemampuan analisis data yang lebih mendalam. Penting juga untuk meningkatkan antarmuka pengguna agar lebih intuitif dan meningkatkan pengalaman pengguna secara keseluruhan. Keamanan data harus diperkuat dengan implementasi protokol keamanan yang lebih kuat sebagai prioritas utama. Pengujian dan evaluasi berkala terhadap aplikasi ini penting untuk memastikan kesesuaian dan kehandalan dalam memenuhi kebutuhan pengguna secara efektif.