

LAPORAN PROYEK AKHIR

PORTAL AKADEMIK SEDERHANA



Dosen Pengampu:
Eka Qadri Nuranti B., S.Kom., M.Kom

Oleh:
Kelompok 4

Andi Magfirah Maqbul	221011048
Dikriani	221011057
Garizah Fauziyah Humaima	221011053
Widya Puspita Sari	221011017

INSTITUT TEKNOLOGI BACHARUDDIN JUSUF HABIBIE
PROGRAM STUDI ILMU KOMPUTER
TAHUN AJARAN 2023/2024

DAFTAR ISI

DAFTAR ISI.....	2
BAB I PENDAHULUAN.....	3
A. Latar Belakang.....	3
B. Tujuan.....	3
C. Deskripsi Proyek.....	3
BAB II RANCANGAN BASIS DATA.....	4
A. Entity Relationship Diagram (ERD).....	4
B. Skema Diagram.....	5
BAB III IMPLEMENTASI.....	6
A. Menghubungkan Java dengan Database.....	6
B. Query untuk Membuat Tabel.....	6
C. Query untuk Menambahkan Data ke Dalam Tabel.....	10
D. Proses Login Pengguna.....	15
E. Akses Database Tiap Pengguna.....	16
F. Penggunaan Fitur.....	29
BAB IV PENUTUP.....	39
A. Kesimpulan.....	39

BAB I

PENDAHULUAN

A. Latar Belakang

Dalam era digital ini, penggunaan teknologi informasi semakin merambah ke berbagai sektor, termasuk di dalamnya pendidikan tinggi. Portal Akademik merupakan sebuah inovasi dalam dunia pendidikan yang bertujuan untuk memberikan kemudahan dan efisiensi dalam manajemen akademik di sebuah institusi pendidikan. Portal Akademik ini muncul dari kebutuhan untuk meningkatkan kualitas layanan pendidikan, memudahkan akses informasi, dan meningkatkan efisiensi proses administratif di lingkungan kampus. Portal Akademik hadir sebagai solusi untuk menyederhanakan dan meningkatkan transparansi dalam manajemen akademik, memungkinkan setiap pengguna untuk mengakses informasi dan melakukan aktivitas tertentu dengan lebih efisien.

B. Tujuan

Tujuan utama dari Portal Akademik ini adalah memberikan pengalaman pengguna yang lebih baik bagi mahasiswa, dosen, dan admin dalam mengakses serta mengelola informasi akademik.

C. Deskripsi Proyek

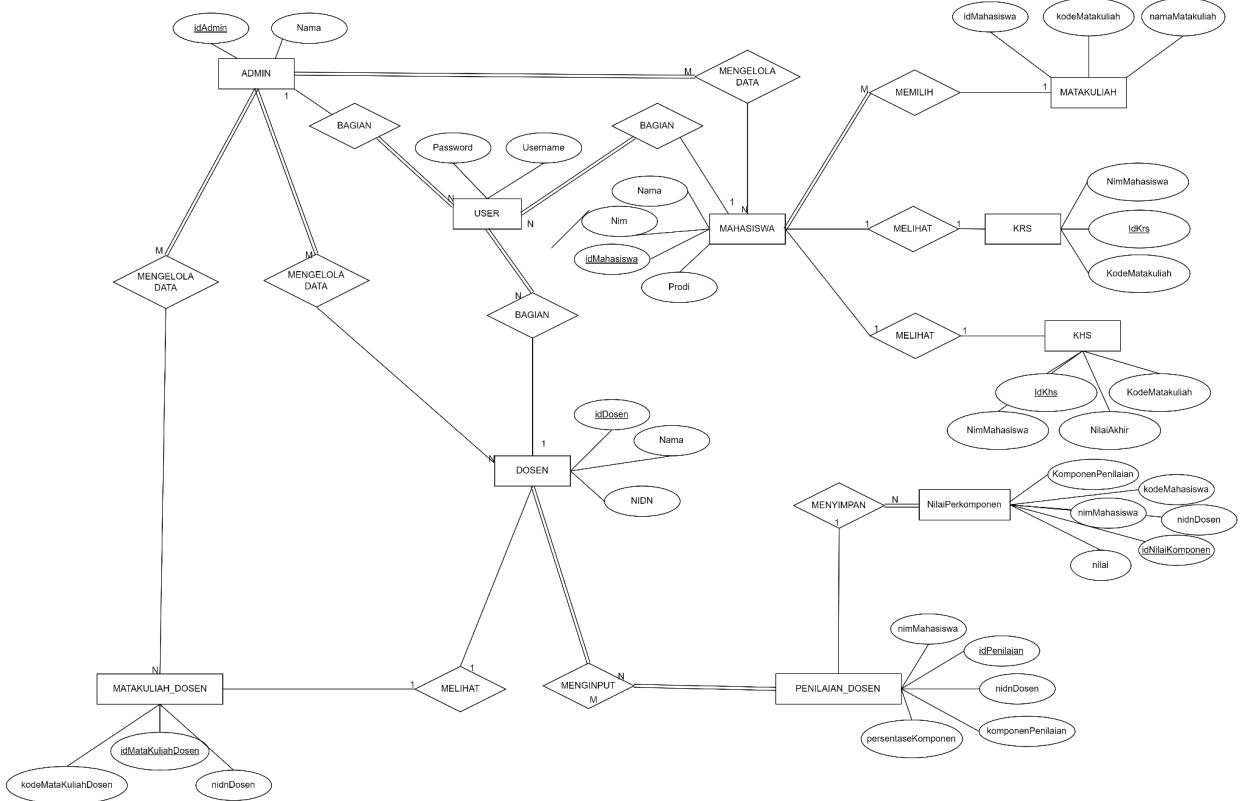
Portal Akademik ini memberikan akses dan fitur yang berbeda kepada setiap jenis pengguna. Mahasiswa dapat menggunakan portal untuk aktivitas seperti pemilihan KRS, melihat KHS, dan meninjau KRS mereka. Dosen memiliki kewenangan untuk melihat jadwal mata kuliah yang diajar dan menginput nilai mahasiswa.

Sementara itu, admin memiliki peran yang lebih luas, memungkinkan mereka untuk menambahkan data mahasiswa, dosen, dan matakuliah ke dalam sistem. Selain itu, admin juga dapat mengelola jadwal matakuliah yang diajarkan oleh dosen, memastikan keakuratan dan ketersediaan informasi jadwal kuliah.

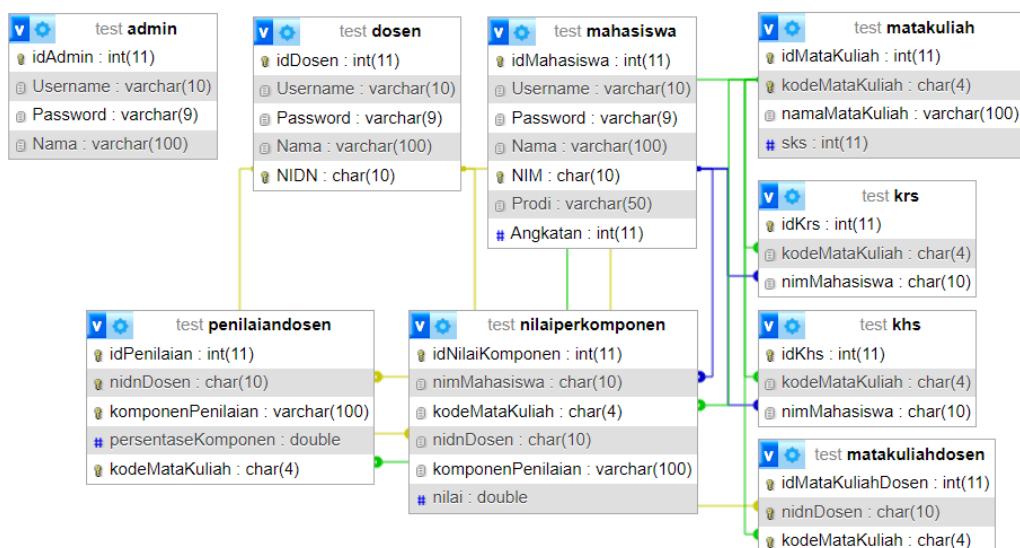
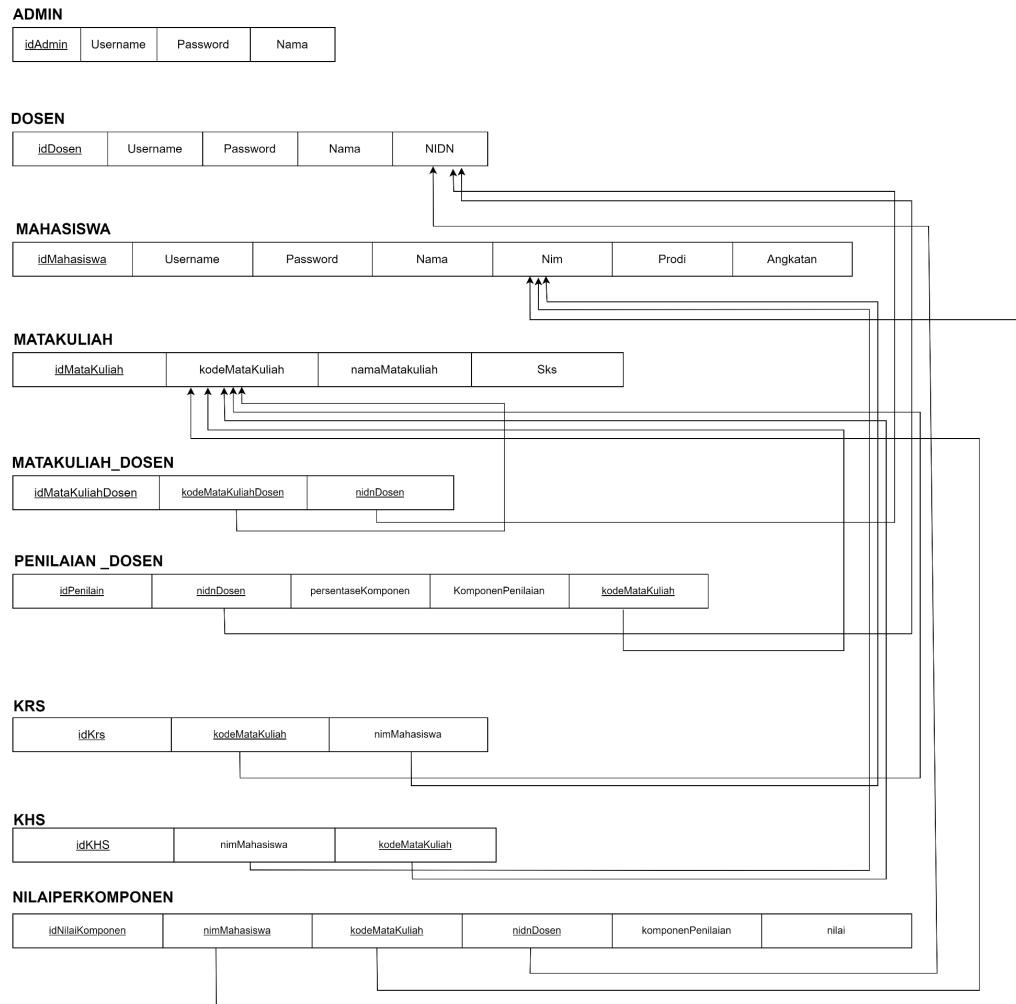
BAB II

RANCANGAN BASIS DATA

A. Entity Relationship Diagram (ERD)



B. Skema Diagram



BAB III

IMPLEMENTASI

A. Menghubungkan Java dengan Database

```
src > ConnectionDB.java > ...
1  import java.sql.Connection;
2  import java.sql.DriverManager;
3  import java.sql.SQLException;
4
5  public class ConnectionDB {
6      private static final String URL = "jdbc:mysql://localhost/portalAkademik";
7      private static final String USERNAME = "root";
8      private static final String PASSWORD = "";
9
10     public static Connection getConnection() throws SQLException {
11         return DriverManager.getConnection(URL, USERNAME, PASSWORD);
12     }
13 }
```

B. Query untuk Membuat Tabel

1. Memanggil Semua metode buat table dengan satu konstruktor

```
7   public static void createTables(Connection connection) {
8       AdminTable(connection);
9       DosenTable(connection);
10      MahasiswaTable(connection);
11      MataKuliahTable(connection);
12      KrsTable(connection);
13      KhsTable(connection);
14      NilaiPerKomponenTable(connection);
15      PenilaianDosenTable(connection);
16      MataKuliahDosenTable(connection);
17  }
```

2. Create Table Admin

```
19  private static void AdminTable(Connection connection) {
20      String query = "CREATE TABLE IF NOT EXISTS Admin (" +
21          "idAdmin INT(10) PRIMARY KEY," +
22          "Username VARCHAR(10) COLLATE utf8mb4_bin," +
23          "Password VARCHAR(9) COLLATE utf8mb4_bin," +
24          "Nama VARCHAR(100) COLLATE utf8mb4_bin" +
25          ")";
26      executeQuery(connection, query);
27  }
```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	idAdmin	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	Username	varchar(10)	utf8mb4_bin		Ya	NULL		
3	Password	varchar(9)	utf8mb4_bin		Ya	NULL		
4	Nama	varchar(100)	utf8mb4_bin		Ya	NULL		

3. Create Table Dosen

```
29     private static void DosenTable(Connection connection) {  
30         String query = "CREATE TABLE IF NOT EXISTS Dosen (" +  
31             "idDosen INT AUTO_INCREMENT PRIMARY KEY," +  
32             "Username VARCHAR(10) COLLATE utf8mb4_bin," +  
33             "Password VARCHAR(9) COLLATE utf8mb4_bin," +  
34             "Nama VARCHAR(100) COLLATE utf8mb4_bin," +  
35             "NIDN CHAR(10) UNIQUE COLLATE utf8mb4_bin" +  
36             ")";  
37         executeQuery(connection, query);  
38     }
```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/>	1 idDosen 🛡	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/>	2 Username	varchar(10)	utf8mb4_bin		Ya	NULL		
<input type="checkbox"/>	3 Password	varchar(9)	utf8mb4_bin		Ya	NULL		
<input type="checkbox"/>	4 Nama	varchar(100)	utf8mb4_bin		Ya	NULL		
<input type="checkbox"/>	5 NIDN 🛡	char(10)	utf8mb4_bin		Ya	NULL		

4. Create Table Mahasiswa

```
40     static void MahasiswaTable(Connection connection) {  
41         String query = "CREATE TABLE IF NOT EXISTS Mahasiswa (" +  
42             "idMahasiswa INT AUTO_INCREMENT PRIMARY KEY," +  
43             "Username VARCHAR(10) COLLATE utf8mb4_bin," +  
44             "Password VARCHAR(9) COLLATE utf8mb4_bin," +  
45             "Nama VARCHAR(100) COLLATE utf8mb4_bin," +  
46             "NIM CHAR(10) UNIQUE COLLATE utf8mb4_bin," +  
47             "Prodi VARCHAR(50) COLLATE utf8mb4_bin," +  
48             "Angkatan INT" +  
49             ")";  
50         executeQuery(connection, query);  
51     }
```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/>	1 idMahasiswa 🛡	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/>	2 Username	varchar(10)	utf8mb4_bin		Ya	NULL		
<input type="checkbox"/>	3 Password	varchar(9)	utf8mb4_bin		Ya	NULL		
<input type="checkbox"/>	4 Nama	varchar(100)	utf8mb4_bin		Ya	NULL		
<input type="checkbox"/>	5 NIM 🛡	char(10)	utf8mb4_bin		Ya	NULL		
<input type="checkbox"/>	6 Prodi	varchar(50)	utf8mb4_bin		Ya	NULL		
<input type="checkbox"/>	7 Angkatan	int(11)			Ya	NULL		

5. Create Table Mata Kuliah

```

53     private static void MataKuliahTable(Connection connection) {
54         String query = "CREATE TABLE IF NOT EXISTS mataKuliah (" +
55             "idMataKuliah INT AUTO_INCREMENT PRIMARY KEY," +
56             "kodeMataKuliah CHAR(4) UNIQUE COLLATE utf8mb4_bin," +
57             "namaMataKuliah VARCHAR(100) COLLATE utf8mb4_bin," +
58             "sks INT" +
59             ");";
60         executeQuery(connection, query);
61     }

```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	idMataKuliah	int(10)			Tidak	Tidak ada		AUTO_INCREMENT
2	kodeMataKuliah	char(4)	utf8mb4_general_ci		Ya	NULL		
3	namaMataKuliah	varchar(100)	utf8mb4_general_ci		Ya	NULL		
4	sks	int(3)			Ya	NULL		

6. Create Table Krs

```

63     private static void KrsTable(Connection connection) {
64         String query = "CREATE TABLE IF NOT EXISTS Krs (" +
65             "idKrs INT AUTO_INCREMENT PRIMARY KEY," +
66             "kodeMataKuliah CHAR(4) COLLATE utf8mb4_bin," +
67             "nimMahasiswa CHAR(10) COLLATE utf8mb4_bin," +
68             "FOREIGN KEY (kodeMataKuliah) REFERENCES mataKuliah(kodeMataKuliah)," +
69             "FOREIGN KEY (nimMahasiswa) REFERENCES mahasiswa(NIM)" +
70             ") ENGINE=InnoDB;";
71         executeQuery(connection, query);
72     }

```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	idKrs	int(10)			Tidak	Tidak ada		AUTO_INCREMENT
2	kodeMataKuliah	char(4)	utf8mb4_bin		Ya	NULL		
3	nimMahasiswa	char(10)	utf8mb4_bin		Ya	NULL		

7. Create Table Khs

```

74     private static void KhsTable(Connection connection) {
75         String query = "CREATE TABLE IF NOT EXISTS KHS (" +
76             "idKhs INT AUTO_INCREMENT PRIMARY KEY," +
77             "kodeMataKuliah CHAR(4) COLLATE utf8mb4_bin," +
78             "nimMahasiswa CHAR(10) COLLATE utf8mb4_bin," +
79             "nilaiAkhir DOUBLE," +
80             "FOREIGN KEY (kodeMataKuliah) REFERENCES mataKuliah(kodeMataKuliah)," +
81             "FOREIGN KEY (nimMahasiswa) REFERENCES mahasiswa(NIM)" +
82             ") ENGINE=InnoDB;";
83         executeQuery(connection, query);
84     }

```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	idKHS	int(10)			Tidak	Tidak ada		AUTO_INCREMENT
2	nimMahasiswa	char(9)	utf8mb4_bin		Ya	NULL		
3	kodeMataKuliah	char(4)	utf8mb4_bin		Ya	NULL		
4	nilaiAkhir	double			Ya	NULL		

8. Create Table nilaiPerKomponen

```

85     private static void NilaiPerKomponenTable(Connection connection) {
86         String query = "CREATE TABLE IF NOT EXISTS nilaiPerKomponen (" +
87             "idNilaiKomponen INT AUTO_INCREMENT PRIMARY KEY," +
88             "nimMahasiswa CHAR(10) COLLATE utf8mb4_bin," +
89             "kodeMataKuliah CHAR(4) COLLATE utf8mb4_bin," +
90             "nidnDosen CHAR(10) COLLATE utf8mb4_bin," +
91             "komponenPenilaian VARCHAR(100) COLLATE utf8mb4_bin," +
92             "nilai DOUBLE," +
93             "FOREIGN KEY (nimMahasiswa) REFERENCES mahasiswa(NIM)," +
94             "FOREIGN KEY (kodeMataKuliah) REFERENCES mataKuliah(kodeMataKuliah)," +
95             "FOREIGN KEY (nidnDosen) REFERENCES dosen(NIDN)" +
96         ") ENGINE=InnoDB;";
97         executeQuery(connection, query);
98     }

```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	idNilaiKomponen 🔑	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	nimMahasiswa 🔑	char(10)	utf8mb4_bin		Ya	NULL		
3	kodeMataKuliah 🔑	char(4)	utf8mb4_bin		Ya	NULL		
4	nidnDosen 🔑	char(10)	utf8mb4_bin		Ya	NULL		
5	komponenPenilaian	varchar(100)	utf8mb4_bin		Ya	NULL		
6	nilai	double			Ya	NULL		

9. Create Table penilaianDosen

```

100    private static void PenilaianDosenTable(Connection connection) {
101        String query = "CREATE TABLE IF NOT EXISTS penilaianDosen (" +
102            "idPenilaian INT AUTO_INCREMENT PRIMARY KEY," +
103            "nidnDosen CHAR(10) COLLATE utf8mb4_bin," +
104            "komponenPenilaian VARCHAR(100) COLLATE utf8mb4_bin," +
105            "persentaseKomponen DOUBLE," +
106            "kodeMataKuliah CHAR(4) COLLATE utf8mb4_bin," +
107            "FOREIGN KEY (nidnDosen) REFERENCES dosen(NIDN)," +
108            "FOREIGN KEY (kodeMataKuliah) REFERENCES mataKuliah(kodeMataKuliah)," +
109            "UNIQUE KEY (nidnDosen, kodeMataKuliah, komponenPenilaian)" +
110        ") ENGINE=InnoDB;";
111        executeQuery(connection, query);
112    }

```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	idPenilaian 🔑	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	nidnDosen 🔑	varchar(10)	utf8mb4_general_ci		Tidak	Tidak ada		
3	komponenPenilaian	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada		
4	persentaseKomponen	char(5)	utf8mb4_general_ci		Ya	NULL		
5	kodeMataKuliah	char(4)	utf8mb4_general_ci		Ya	NULL		

10. Create Table mataKuliahDosen

```

113
114    private static void MataKuliahDosenTable(Connection connection) {
115        String query = "CREATE TABLE IF NOT EXISTS mataKuliahDosen (" +
116            "idMataKuliahDosen INT AUTO_INCREMENT PRIMARY KEY," +
117            "nidnDosen CHAR(10) COLLATE utf8mb4_bin," +
118            "kodeMataKuliah CHAR(4) COLLATE utf8mb4_bin," +
119            "FOREIGN KEY (nidnDosen) REFERENCES dosen(NIDN)," +
120            "FOREIGN KEY (kodeMataKuliah) REFERENCES mataKuliah(kodeMataKuliah)," +
121            "UNIQUE KEY (nidnDosen, kodeMataKuliah)" +
122        ") ENGINE=InnoDB;";
123        executeQuery(connection, query);
124    }

```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	idMKDosen 🛡	int(10)			Tidak	Tidak ada		AUTO_INCREMENT
2	kodeMataKuliah 🛡	varchar(4)	utf8mb4_bin		Tidak	Tidak ada		
3	nidnDosen 🛡	char(9)	utf8mb4_bin		Ya	NULL		

11. Membuat konstruktor `executeQuery` untuk menjalankan pernyataan SQL yang dapat mengubah data dan di dalam konstruktor ini ditambahkan `SQLException` untuk memeriksa pesan error

```

126     private static void executeQuery(Connection connection, String query) {
127         try (Statement statement = connection.createStatement()) {
128             statement.executeUpdate(query);
129         } catch (SQLException e) {
130             e.printStackTrace();
131         }
132     }

```

C. Query untuk Menambahkan Data ke Dalam Tabel

Menggunakan metode `INSERT INTO` untuk menambahkan data Admin, Dosen, Mahasiswa dan Mata Kuliah. Setiap fungsi `insertDosen`, `insertMahasiswa`, `insertAdmin` dan `insertMataKuliah` mengambil parameter yang mewakili data yang akan dimasukkan ke dalam tabel dan memastikan bahwa data yang ingin ditambahkan belum ada sebelumnya dalam tabel.

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class InsertTable {

    public static void insertTables(Connection connection) {
        try {
            // Memasukkan data dosen
            insertDosen(connection, "Dosen", "NIDN123", "Dosen 1",
"NIDN123");
            insertDosen(connection, "Dosen", "NIDN456", "Dosen 2",
"NIDN456");
            insertDosen(connection, "Dosen", "NIDN789", "Dosen 3",
"NIDN789");
            insertDosen(connection, "Dosen", "NIDN101", "Dosen 4",
"NIDN101");
            insertDosen(connection, "Dosen", "NIDN202", "Dosen 5",
"NIDN202");

            // Memasukkan data mahasiswa

```

```
        insertMahasiswa(connection, "Mahasiswa", "NIM123",
"Mahasiswa 1", "NIM123", "Teknik Informatika", 2023);
        insertMahasiswa(connection, "Mahasiswa", "NIM456",
"Mahasiswa 2", "NIM456", "Sistem Informasi", 2022);
        insertMahasiswa(connection, "Mahasiswa", "NIM789",
"Mahasiswa 3", "NIM789", "Ilmu Komputer", 2022);
        insertMahasiswa(connection, "Mahasiswa", "NIM101",
"Mahasiswa 4", "NIM101", "Teknik Elektro", 2022);
        insertMahasiswa(connection, "Mahasiswa", "NIM202",
"Mahasiswa 5", "NIM202", "Teknik Mesin", 2023);
        insertMahasiswa(connection, "Mahasiswa", "NIM303",
"Mahasiswa 6", "NIM303", "Matematika", 2022);

        // Memasukkan data admin
        insertAdmin(connection, "12345", "Admin", "pass1",
"Nama admin1");
        insertAdmin(connection, "67891", "Admin", "pass2",
"Nama admin2");

        // Memasukkan data mata kuliah
        insertMataKuliah(connection, "MK1", "Aljabar Linear",
3);
        insertMataKuliah(connection, "MK2", "Interaksi Manusia
dan Komputer", 3);
        insertMataKuliah(connection, "MK3", "Pemrograman
Berorientasi Objek", 3);
        insertMataKuliah(connection, "MK4", "Sistem Basis
Data", 3);
        insertMataKuliah(connection, "MK5", "Sistem Operasi",
3);
        insertMataKuliah(connection, "MK6", "Jaringan
Komputer", 3);
        insertMataKuliah(connection, "MK7",
"Technopreneurship", 2);
        insertMataKuliah(connection, "MK8", "Riset teknologi
informasi", 2);
        insertMataKuliah(connection, "MK9", "Transformasi
Digital", 3);
        insertMataKuliah(connection, "MK10", "Supply Chain
Management", 2);
        insertMataKuliah(connection, "MK11", "Teori Peluang",
3);
```

```
        insertMataKuliah(connection, "MK12", "Matematika Diskrit", 3);
        insertMataKuliah(connection, "MK13", "Persamaan Diferensial Biasa", 3);
        insertMataKuliah(connection, "MK14", "Analisis Suku Bunga Berbasis Web", 3);
        insertMataKuliah(connection, "MK15", "Matematika Bisnis dan Teknologi", 3);
        insertMataKuliah(connection, "MK16", "Kalkulus lanjutan", 3);

        // Memasukkan data Mata Kuliah yang Diajarkan Dosen
        insertMataKuliahDosen(connection, "NIDN123", "MK1");
        insertMataKuliahDosen(connection, "NIDN456", "MK2");
        insertMataKuliahDosen(connection, "NIDN789", "MK3");
        insertMataKuliahDosen(connection, "NIDN101", "MK4");
        insertMataKuliahDosen(connection, "NIDN202", "MK5");
        insertMataKuliahDosen(connection, "NIDN123", "MK6");
        insertMataKuliahDosen(connection, "NIDN456", "MK7");
        insertMataKuliahDosen(connection, "NIDN789", "MK8");
        insertMataKuliahDosen(connection, "NIDN101", "MK9");
        insertMataKuliahDosen(connection, "NIDN202", "MK10");
        insertMataKuliahDosen(connection, "NIDN123", "MK12");
        insertMataKuliahDosen(connection, "NIDN456", "MK11");
        insertMataKuliahDosen(connection, "NIDN789", "MK13");
        insertMataKuliahDosen(connection, "NIDN101", "MK14");
        insertMataKuliahDosen(connection, "NIDN202", "MK15");
        insertMataKuliahDosen(connection, "NIDN123", "MK16");

        System.out.println("Data berhasil dimasukkan ke dalam tabel.");
    }

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

private static void insertDosen(Connection connection, String username, String password, String nama, String nidn)
    throws SQLException {
    if (!isDataExist(connection, "dosen", "NIDN", nidn)) {
        String query = "INSERT INTO dosen (Username, Password, Nama, NIDN ) VALUES (?, ?, ?, ?, ?)";

```

```
        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setString(1, username);
            pstmt.setString(2, password);
            pstmt.setString(3, nama);
            pstmt.setString(4, nidn);
            pstmt.executeUpdate();
        }
    }
}

private static void insertMahasiswa(Connection connection,
String Username, String Password, String nama,
        String nim, String prodi, int angkatan) throws
SQLException {
    if (!isDataExist(connection, "mahasiswa", "NIM", nim)) {
        String query = "INSERT INTO mahasiswa (Username,
Password, Nama, NIM, Prodi, Angkatan) VALUES (?, ?, ?, ?, ?, ?)";
        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setString(1, Username);
            pstmt.setString(2, Password);
            pstmt.setString(3, nama);
            pstmt.setString(4, nim);
            pstmt.setString(5, prodi);
            pstmt.setInt(6, angkatan);
            pstmt.executeUpdate();
        }
    }
}

private static void insertAdmin(Connection connection, String
idAdmin, String username, String password,
        String nama) throws SQLException {
    if (!isDataExist(connection, "admin", "idAdmin", idAdmin))
{
        String query = "INSERT INTO admin (idAdmin, Username,
Password, Nama) VALUES (?, ?, ?, ?)";
        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setString(1, idAdmin);
            pstmt.setString(2, username);
            pstmt.setString(3, password);
```

```
        pstmt.setString(4, nama);
        pstmt.executeUpdate();
    }
}

private static void insertMataKuliah(Connection connection,
String kodeMataKuliah,
        String namaMataKuliah, int sks) throws SQLException {
    if (!isDataExist(connection, "mataKuliah",
"KodeMataKuliah", kodeMataKuliah)) {
        String query = "INSERT INTO mataKuliah (KodeMataKuliah,
NamaMataKuliah, SKS) VALUES (?, ?, ?)";
        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setString(1, kodeMataKuliah);
            pstmt.setString(2, namaMataKuliah);
            pstmt.setInt(3, sks);
            pstmt.executeUpdate();
        }
    }
}

private static void insertMataKuliahDosen(Connection
connection, String nidnDosen, String kodeMataKuliah)
        throws SQLException {
    String query = "INSERT INTO mataKuliahDosen (nidnDosen,
kodeMataKuliah) VALUES (?, ?)";
    try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
        pstmt.setString(1, nidnDosen);
        pstmt.setString(2, kodeMataKuliah);
        pstmt.executeUpdate();
    }
}

// memeriksa apakah data sudah ada dalam tabel
private static boolean isDataExist(Connection connection,
String tableName, String columnName, String value)
        throws SQLException {
    String query = "SELECT COUNT(*) FROM " + tableName + "
WHERE " + columnName + " = ?";
```

```

        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setString(1, value);
            try (ResultSet resultSet = pstmt.executeQuery()) {
                if (resultSet.next()) {
                    int count = resultSet.getInt(1);
                    return count > 0;
                }
            }
        }
        return false;
    }
}

```

Apabila data berhasil ditambahkan maka akan muncul seperti ini dan kemudian menu login akan ditampilkan

```

PS E:\KULIAH\FIRA\TUGAS KULIAH\SEM 3\PEMROGRAMAN BERORIENTASI OBJEK\Proyek\PortalAkademik> & 'C:\Program Files\Java\jdk-21\bin
\java.exe' '@C:\Users\WINDOW~1\AppData\Local\Temp\cp_6p4eq5xq311bzimrkd11qlq0.argfile' 'PortalAkademik'
Data berhasil dimasukkan ke dalam tabel.

```

D. Proses Login Pengguna

Proses login pada Portal Akademik ini menggunakan proses otentikasi. Pengguna diminta untuk memasukkan username dan password mereka yang kemudian dicocokkan dengan data yang ada dalam Database. Jika verifikasi berhasil, pengguna akan diarahkan ke menu sesuai dengan peran mereka masing-masing (Mahasiswa, Dosen, atau Admin). Apabila verifikasi pengguna gagal maka akan muncul tampilan "Login gagal. Periksa kembali username dan password.". disini kami menggunakan untuk melakukan verifikasi berdasarkan username dan password dengan metode `SELECT * FROM`.

```

6  public class AuthenticationService {
7      public User authenticateWithDatabase(Connection connection, String username, String password) {
8          try {
9              Connection connectionn = ConnectionDB.getConnection();
10
11             User Mahasiswa = authenticateMahasiswa(connectionn, username, password);
12             if (Mahasiswa != null) {
13                 return Mahasiswa;
14             }
15
16             User dosen = authenticateDosen(connectionn, username, password);
17             if (dosen != null) {
18                 return dosen;
19             }
20
21             User admin = authenticateAdmin(connectionn, username, password);
22             if (admin != null) {
23                 return admin;
24             }
25         } catch (SQLException e) {
26             e.printStackTrace();
27         }
28
29         return null;
30     }
}

```

Proses Autentifikasi sebagai Mahasiswa

```
32     private Mahasiswa authenticateMahasiswa(Connection connection, String username, String password)
33         throws SQLException {
34     String mahasiswaQuery = "SELECT * FROM mahasiswa WHERE Username = ? AND Password = ?";
35     try (PreparedStatement mahasiswaPstmt = connection.prepareStatement(mahasiswaQuery)) {
36         mahasiswaPstmt.setString(parameterIndex:1, username);
37         mahasiswaPstmt.setString(parameterIndex:2, password);
38         ResultSet mahasiswaResultSet = mahasiswaPstmt.executeQuery();
39
40         if (mahasiswaResultSet.next()) {
41             String nim = mahasiswaResultSet.getString(columnLabel:"Nim");
42             String prodi = mahasiswaResultSet.getString(columnLabel:"Prodi");
43             int angkatan = mahasiswaResultSet.getInt(columnLabel:"Angkatan");
44             String nama = mahasiswaResultSet.getString(columnLabel:"Nama");
45
46             return new Mahasiswa(username, password, nama, nim, prodi, angkatan);
47         }
48     }
49     return null;
50 }
```

Proses Autentifikasi sebagai Dosen

```
52     private Dosen authenticateDosen(Connection connection, String username, String password) throws SQLException {
53     String dosenQuery = "SELECT * FROM dosen WHERE username = ? AND password = ?";
54     try (PreparedStatement dosenPstmt = connection.prepareStatement(dosenQuery)) {
55         dosenPstmt.setString(parameterIndex:1, username);
56         dosenPstmt.setString(parameterIndex:2, password);
57         ResultSet dosenResultSet = dosenPstmt.executeQuery();
58
59         if (dosenResultSet.next()) {
60             String nidn = dosenResultSet.getString(columnLabel:"nidn");
61             String nama = dosenResultSet.getString(columnLabel:"nama");
62
63             return new Dosen(username, password, nama, nidn);
64         }
65     }
66     return null;
67 }
```

Proses Autentifikasi sebagai Admin

```
69     private Admin authenticateAdmin(Connection connection, String username, String password) throws SQLException {
70     String adminQuery = "SELECT * FROM admin WHERE username = ? AND password = ?";
71     try (PreparedStatement adminPstmt = connection.prepareStatement(adminQuery)) {
72         adminPstmt.setString(parameterIndex:1, username);
73         adminPstmt.setString(parameterIndex:2, password);
74         ResultSet adminResultSet = adminPstmt.executeQuery();
75
76         if (adminResultSet.next()) {
77             String IdAdmin = adminResultSet.getString(columnLabel:"IdAdmin");
78             String namaAdmin = adminResultSet.getString(columnLabel:"Nama");
79
80             return new Admin(username, password, namaAdmin, IdAdmin);
81         }
82     }
83     return null;
84 }
```

E. Akses Database Tiap Pengguna

1. Melihat KRS

- Melihat data dari tabel "krs" dengan melakukan inner join dengan tabel "mataKuliah". Inner join dilakukan berdasarkan kondisi bahwa nilai pada kolom "kodeMataKuliah" di tabel "krs" sama dengan nilai pada kolom "kodeMataKuliah" di tabel "mataKuliah". Setelah itu, hasil join akan difilter dengan kondisi bahwa nilai pada kolom "nimMahasiswa" di tabel "krs" sama dengan nilai yang diberikan sebagai parameter.

```

315  public static void BacaKRSDaridatabase(Mahasiswa mahasiswa, Connection connection) {
316      System.out.println("\nKartu Rencana Studi " + mahasiswa.getNama());
317
318      try {
319          String query = "SELECT * FROM krs "
320              + "INNER JOIN mataKuliah ON krs.kodeMataKuliah = mataKuliah.kodeMataKuliah "
321              + "WHERE krs.nimMahasiswa = ?";
322          try (PreparedStatement pstmt = connection.prepareStatement(query)) {
323              pstmt.setString(parameterIndex:1, mahasiswa.getNim());
324              ResultSet resultSet = pstmt.executeQuery();
325
326              if (!resultSet.next()) {
327                  System.out.println("KRS dengan nim " + mahasiswa.getNim() + " tidak ditemukan.");
328
329                  System.out.print(s:"\nIngin memilih KRS? (y/n): ");
330                  String pilih = scanner.nextLine();
331                  if (pilih.equalsIgnoreCase("y")) {
332                      pilihKRS(mahasiswa, connection);
333                      return;
334                  }
335              } else {
336                  System.out.printf(format:"\n%-15s %-40s %s\n", ...args:"Kode", "Nama Mata Kuliah", "SKS");
337
338                  int totalSKS = 0;
339
340                  do {
341                      String kodeMataKuliah = resultSet.getString(columnLabel:"kodeMataKuliah");

```

2. Menambahkan Mata Kuliah Ditambahkan kedalam KRS

- INSERT INTO digunakan untuk menambahkan data baru kedalam KRS.

```

383
384     private static void simpanMataKuliahKRS(Connection connection, String nim, MataKuliah mataKuliah)
385         throws SQLException {
386     String query = "INSERT INTO krs (nimMahasiswa, kodeMataKuliah) VALUES (?, ?)";
387     try (PreparedStatement pstmt = connection.prepareStatement(query)) {
388         pstmt.setString(parameterIndex:1, nim);
389         pstmt.setString(parameterIndex:2, mataKuliah.getKode());
390     }
391 }
392

```

3. Melihat KHS

- SELECT * FROM digunakan untuk memilih "KHS" dengan metode WHERE untuk memilih data tertentu

```

7   public static void lihatKHS(Mahasiswa mahasiswa, Connection connection) throws SQLException {
8       String query = "SELECT * FROM khs WHERE nimMahasiswa = ?";
9       try (PreparedStatement pstmt = connection.prepareStatement(query)) {
10           pstmt.setString(parameterIndex:1, mahasiswa.getNim());
11
12           try (ResultSet resultSet = pstmt.executeQuery()) {
13
14               if (!resultSet.next()) {
15                   System.out.println("\nKHS dengan Nama " + mahasiswa.getNama() + " dan NIM " + mahasiswa.getNim()
16                   + " tidak ditemukan.");
17               } else {
18                   System.out.println(x:"\nKartu Hasil Studi (KHS)");
19                   System.out.println("Nama: " + mahasiswa.getNama());
20                   System.out.println("NIM : " + mahasiswa.getNim());
21                   System.out.println(x:"-----");
22                   System.out.format(format:"|%-5s|%-30s|%6s|%12s|\n", ...args:"KODE", "MATAKULIAH", "NILAI",
23                   "NILAI HURUF");
24                   System.out.println(x:"|-----|-----|-----|-----|-----|");
25
26               do {
27                   String kodeMataKuliah = resultSet.getString(columnLabel:"kodeMataKuliah");
28                   double nilaiAkhir = resultSet.getDouble(columnLabel:"nilaiAkhir");
29
30                   String queryMataKuliah = "SELECT namaMataKuliah FROM matakuliah WHERE kodeMataKuliah = ?";
31                   try (PreparedStatement pstmtMataKuliah = connection.prepareStatement(queryMataKuliah)) {
32                       pstmtMataKuliah.setString(parameterIndex:1, kodeMataKuliah);
33                       try (ResultSet resultSetMataKuliah = pstmtMataKuliah.executeQuery()) {

```

4. Menyimpan KRS

- connection.setAutoCommit(false) Mengatur mode otomatis commit pada koneksi database menjadi false.
- for (MataKuliah mataKuliah : krsMahasiswa) adalah loop for-each yang mengiterasi melalui setiap mata kuliah dalam daftar KRS mahasiswa.
- simpanMataKuliahKRS(connection, mahasiswa.getNim(), mataKuliah) digunakan untuk menyimpan data mata kuliah ke dalam database.

- `connection.commit()` digunakan jika semua operasi dalam loop selesai tanpa kesalahan, maka pernyataan `connection.commit()` dijalankan untuk meng-commit transaksi. Ini berarti perubahan yang dilakukan dalam transaksi tersebut akan diterapkan secara permanen ke dalam database.

```

359     private static void simpanKRSKeDatabase(Mahasiswa mahasiswa, List<MataKuliah> krsMahasiswa, Connection connection) {
360         try {
361             connection.setAutoCommit(autoCommit:false);
362
363             for (MataKuliah mataKuliah : krsMahasiswa) {
364                 simpanMataKuliahKRS(connection, mahasiswa.getNim(), mataKuliah);
365             }
366
367             connection.commit();
368         } catch (SQLException e) {
369             e.printStackTrace();
370             try {
371                 connection.rollback();
372             } catch (SQLException ex) {
373                 ex.printStackTrace();
374             }
375         } finally {
376             try {
377                 connection.setAutoCommit(autoCommit:true);
378             } catch (SQLException e) {
379                 e.printStackTrace();
380             }
381         }
382     }

```

5. Mengecek Mahasiswa dalam KHS

- query ini bertujuan untuk menghitung berapa kali kombinasi nilai "nimMahasiswa" dan "kodeMataKuliah" muncul dalam tabel "KHS".

```

88     private static boolean cekKHS(Connection connection, String nimMahasiswa, String kodeMataKuliah)
89         throws SQLException {
90         String query = "SELECT COUNT(*) AS count FROM KHS WHERE nimMahasiswa = ? AND kodeMataKuliah = ?";
91         try (PreparedStatement pstmt = connection.prepareStatement(query)) {
92             pstmt.setString(parameterIndex:1, nimMahasiswa);
93             pstmt.setString(parameterIndex:2, kodeMataKuliah);
94
95             try (ResultSet resultSet = pstmt.executeQuery()) {
96                 if (resultSet.next()) {
97                     int count = resultSet.getInt(columnLabel:"count");
98                     return count > 0;
99                 }
100            }
101        }
102        return false;
103    }

```

6. Menyimpan Nilai ke dalam KHS

- `INSERT INTO` digunakan untuk menambahkan data baru kedalam KHS.

```

220     private static void simpanNilaiKHS(Connection connection, String nimMahasiswa, String kodeMataKuliah,
221         double nilaiAkhir) throws SQLException {
222         if (cekKHS(connection, nimMahasiswa, kodeMataKuliah)) {
223             System.out.println("Nilai akhir untuk matakuliah ini sudah ada di dalam KHS.");
224             return;
225         }
226
227         String query = "INSERT INTO KHS (nimMahasiswa, kodeMataKuliah, nilaiAkhir) VALUES (?, ?, ?)";
228         try (PreparedStatement pstmt = connection.prepareStatement(query)) {
229             pstmt.setString(parameterIndex:1, nimMahasiswa);
230             pstmt.setString(parameterIndex:2, kodeMataKuliah);
231             pstmt.setDouble(parameterIndex:3, nilaiAkhir);
232             pstmt.executeUpdate();
233         }
234     }

```

7. Menampilkan Mata Kuliah Dosen

- Query ini mencari dan menggabungkan data dari tabel `mataKuliahDosen`, `mataKuliah`, dan `dosen`. Kolom yang dipilih untuk ditampilkan adalah `mataKuliah.kodeMataKuliah`, `mataKuliah.namaMataKuliah`, dan `dosen.NIDN`. JOIN digunakan untuk mengaitkan baris-baris dalam `mataKuliahDosen` dengan baris-baris yang sesuai dalam tabel `mataKuliah` dan `dosen`. Pada JOIN pertama (`JOIN mataKuliah ON mataKuliahDosen.kodeMataKuliah = mataKuliah.kodeMataKuliah`), data dari tabel `mataKuliahDosen` dihubungkan dengan data yang sesuai dari `mataKuliah` berdasarkan kolom `kodeMataKuliah`. Kemudian, pada JOIN kedua (`JOIN dosen ON mataKuliahDosen.nidnDosen = dosen.NIDN`), data dari `mataKuliahDosen` dihubungkan dengan data yang sesuai dari `dosen` berdasarkan kolom `nidnDosen`.

```
192     private static void tampilanDaftarMataKuliahDosen(Dosen dosen, Connection connection) {  
193         try {  
194             String query = "SELECT mataKuliahDosen.kodeMataKuliah, mataKuliah.namaMataKuliah " +  
195                         "FROM mataKuliahDosen " +  
196                         "JOIN mataKuliah ON mataKuliahDosen.kodeMataKuliah = mataKuliah.kodeMataKuliah " +  
197                         "WHERE mataKuliahDosen.nidnDosen = ?";  
198  
199             try (PreparedStatement pstmt = connection.prepareStatement(query)) {  
200                 pstmt.setString(parameterIndex:1, dosen.getNidn());  
201                 ResultSet resultSet = pstmt.executeQuery();  
202  
203                 if (!resultSet.next()) {  
204                     System.out  
205                         .println("\nDosen dengan Nama " + dosen.getNama() + " tidak mengajar mata ku-  
206                 } else [  
207                     System.out.println("\nDaftar Mata Kuliah yang Diajar oleh " + dosen.getNidn() + "\n");  
208                     System.out.println(x: "-----");  
209                     System.out.printf(format:"|%-15s| %-40s|\n", ...args:"Kode", "Nama Mata Kuliah");  
210             }  
211         } catch (SQLException e) {  
212             e.printStackTrace();  
213         }  
214     }  
215 }
```

8. Menyimpan Komponen Penilaian Dosen

- INSERT INTO digunakan untuk menambahkan data baru

```
72     private static void simpanKomponenPenilaian(Connection connection, String nidnDosen, String kodeMataKuliah,  
73             String[] komponenPenilaian, double[] persentaseKomponen)  
74     throws SQLException {  
75         String query = "INSERT INTO penilaianDosen (nidnDosen, kodeMataKuliah,"  
76                         "+ komponenPenilaian, persentaseKomponen) VALUES (?, ?, ?, ?)";  
77         try (PreparedStatement pstmt = connection.prepareStatement(query)) {  
78             for (int i = 0; i < komponenPenilaian.length; i++) {  
79                 pstmt.setString(parameterIndex:1, nidnDosen);  
80                 pstmt.setString(parameterIndex:2, kodeMataKuliah);  
81                 pstmt.setString(parameterIndex:3, komponenPenilaian[i]);  
82                 pstmt.setDouble(parameterIndex:4, persentaseKomponen[i]);  
83                 pstmt.executeUpdate();  
84             }  
85         }  
86     }
```

9. Mengakses Komponen Penilaian Dosen

- SELECT komponenPenilaian FROM penilaianDosen digunakan untuk memilih "komponenPenilaian" dari "penilaianDosen" dengan metode WHERE untuk memilih data tertentu

```

249     private static List<String> getKomponenPenilaian(Connection connection, String nidnDosen, String kodeMataKuliah)
250         throws SQLException {
251     List<String> komponenPenilaianList = new ArrayList<>();
252
253     String query = "SELECT komponenPenilaian FROM penilaianDosen WHERE nidnDosen = ? AND kodeMataKuliah = ?";
254     try (PreparedStatement pstmt = connection.prepareStatement(query)) {
255         pstmt.setString(parameterIndex:1, nidnDosen);
256         pstmt.setString(parameterIndex:2, kodeMataKuliah);
257
258         try (ResultSet resultSet = pstmt.executeQuery()) {
259             while (resultSet.next()) {
260                 String komponenPenilaian = resultSet.getString(columnLabel:"komponenPenilaian");
261                 komponenPenilaianList.add(komponenPenilaian);
262             }
263         } catch (SQLException e) {
264             e.printStackTrace();
265         }
266     }
267     return komponenPenilaianList;
268 }

```

10. Mengakses Presentasi Komponen yang Telah Disimpan

- SELECT digunakan untuk mengambil nilai dari kolom persentaseKomponen dari tabel penilaianDosen berdasarkan beberapa kondisi tertentu.

```

310     private static double getPersentaseKomponen(Connection connection, String nidnDosen, String kodeMataKuliah,
311         String komponenPenilaian) throws SQLException {
312     String query = "SELECT persentaseKomponen FROM penilaianDosen "
313         + "WHERE nidnDosen = ? AND kodeMataKuliah = ? AND komponenPenilaian = ?";
314
315     try (PreparedStatement pstmt = connection.prepareStatement(query)) {
316         pstmt.setString(parameterIndex:1, nidnDosen);
317         pstmt.setString(parameterIndex:2, kodeMataKuliah);
318         pstmt.setString(parameterIndex:3, komponenPenilaian);
319
320         try (ResultSet resultSet = pstmt.executeQuery()) {
321             if (resultSet.next()) {
322                 return resultSet.getDouble(columnLabel:"persentaseKomponen");
323             }
324         }
325     }
326     return 0;
327 }

```

11. Memilih Mahasiswa yang Terdaftar dalam Database dari Mata Kuliah Tertentu

- SELECT yang digunakan untuk mengambil data mahasiswa dari tabel mahasiswa yang terkait dengan suatu mata kuliah dalam tabel krs.

```

134     private static List<Mahasiswa> pilihMahasiswaDariMataKuliah(Connection connection, String kodeMataKuliah)
135         throws SQLException {
136     List<Mahasiswa> mahasiswaList = new ArrayList<>();
137
138     String query = "SELECT mahasiswa.NIM, mahasiswa>Nama, mahasiswa.Prodi FROM krs "
139         + "JOIN mahasiswa ON krs.nimMahasiswa = mahasiswa.NIM WHERE krs.kodeMataKuliah = ?";
140     try (PreparedStatement pstmt = connection.prepareStatement(query)) {
141         pstmt.setString(parameterIndex:1, kodeMataKuliah);
142         try (ResultSet resultSet = pstmt.executeQuery()) {
143             while (resultSet.next()) {
144                 String nimMahasiswa = resultSet.getString(columnLabel:"NIM");
145                 String namaMahasiswa = resultSet.getString(columnLabel:"Nama");
146                 String prodiMahasiswa = resultSet.getString(columnLabel:"Prodi");
147                 mahasiswaList.add(new Mahasiswa(username:"", password:"", namaMahasiswa, nimMahasiswa, prodiMahasiswa, a
148             }
149         } catch (SQLException e) {
150             e.printStackTrace();
151         }
152     }
153     return mahasiswaList;
154 }

```

12. Memeriksa Mata Kuliah yang Diajarkan Dosen

- Query ini mencari dan menggabungkan data dari tabel `mataKuliahDosen`, `mataKuliah`, dan `dosen`. Kolom yang dipilih untuk ditampilkan adalah `mataKuliah.kodeMataKuliah`, `mataKuliah.namaMataKuliah`, dan `dosen.NIDN`. JOIN digunakan untuk mengaitkan baris-baris dalam `mataKuliahDosen` dengan baris-baris yang sesuai dalam tabel `mataKuliah`

dan `dosen`. Pada JOIN pertama (`JOIN mataKuliah ON mataKuliahDosen.kodeMataKuliah = mataKuliah.kodeMataKuliah`), data dari tabel `mataKuliahDosen` dihubungkan dengan data yang sesuai dari `mataKuliah` berdasarkan kolom `kodeMataKuliah`. Kemudian, pada JOIN kedua (`JOIN dosen ON mataKuliahDosen.nidnDosen = dosen.NIDN`), data dari `mataKuliahDosen` dihubungkan dengan data yang sesuai dari `dosen` berdasarkan kolom `nidnDosen`.

```

105     public static List<MataKuliah> periksaMataKuliahDosen(Connection connection, Dosen dosen) throws SQLException {
106         List<MataKuliah> mataKuliahList = new ArrayList<>();
107
108         String query = "SELECT mataKuliah.kodeMataKuliah, mataKuliah.namaMataKuliah, dosen.NIDN " +
109             "FROM mataKuliahDosen " +
110             "JOIN mataKuliah ON mataKuliahDosen.kodeMataKuliah = mataKuliah.kodeMataKuliah " +
111             "JOIN dosen ON mataKuliahDosen.nidnDosen = dosen.NIDN " +
112             "WHERE mataKuliahDosen.nidnDosen = ?";
113
114         String nidnDosen = dosen.getNidn();
115
116         try (PreparedStatement pstmt = connection.prepareStatement(query)) {
117             pstmt.setString(parameterIndex:1, nidnDosen);
118
119             try (ResultSet resultSet = pstmt.executeQuery()) {
120                 while (resultSet.next()) {
121                     String kodeMataKuliah = resultSet.getString(columnLabel:"kodeMataKuliah");
122                     String namaMataKuliah = resultSet.getString(columnLabel:"namaMataKuliah");
123
124                     MataKuliah mataKuliah = new MataKuliah(kodeMataKuliah, namaMataKuliah, sks:0);
125                     mataKuliahList.add(mataKuliah);
126                 }
127             } catch (SQLException e) {
128                 e.printStackTrace();
129             }
130         }

```

13. Menambahkan Data Mahasiswa kedalam Database

```

29     try {
30         String query = "INSERT INTO mahasiswa (username, password, Nama, Nim, Prodi, angkatan) VALUES (?, ?, ?, ?, ?, ?)";
31         try (PreparedStatement pstmt = connection.prepareStatement(query)) {
32             pstmt.setString(parameterIndex:1, username);
33             pstmt.setString(parameterIndex:2, password);
34             pstmt.setString(parameterIndex:3, nama);
35             pstmt.setString(parameterIndex:4, nim);
36             pstmt.setString(parameterIndex:5, prodi);
37             pstmt.setInt(parameterIndex:6, angkatan);
38
39             int affectedRows = pstmt.executeUpdate();
40             if (affectedRows > 0) {
41                 System.out.println(x:"Data mahasiswa berhasil ditambahkan ke database.\n");
42             } else {
43                 System.out.println(x:"Gagal menambahkan data mahasiswa.");
44             }
45         } catch (SQLException e) {
46             e.printStackTrace();
47         }
48     }

```

14. Menambahkan Data Dosen kedalam Database

```

66     try {
67         String query = "INSERT INTO dosen (username, password, NIDN, Nama) VALUES (?, ?, ?, ?)"
68         try (PreparedStatement pstmt = connection.prepareStatement(query)) {
69             pstmt.setString(parameterIndex:1, username);
70             pstmt.setString(parameterIndex:2, password);
71             pstmt.setString(parameterIndex:3, nidn);
72             pstmt.setString(parameterIndex:4, nama);
73
74             int affectedRows = pstmt.executeUpdate();
75             if (affectedRows > 0) {
76                 System.out.println(x:"Data dosen berhasil ditambahkan ke database.");
77             } else {
78                 System.out.println(x:"Gagal menambahkan data dosen.");
79             }
80         } catch (SQLException e) {
81             e.printStackTrace();
82         }
83     }

```

15. Menambahkan Data Mata Kuliah kedalam Database

```
97     try {
98         String query = "INSERT INTO mataKuliah (kodeMataKuliah, namaMataKuliah, sks) VALUES (?, ?, ?)";
99         try (PreparedStatement pstmt = connection.prepareStatement(query)) {
100             pstmt.setString(parameterIndex:1, kodeMataKuliah);
101             pstmt.setString(parameterIndex:2, namaMataKuliah);
102             pstmt.setInt(parameterIndex:3, sks);
103
104             int affectedRows = pstmt.executeUpdate();
105             if (affectedRows > 0) {
106                 System.out.println("Data matakuliah berhasil ditambahkan ke database.");
107             } else {
108                 System.out.println("Gagal menambahkan data mata kuliah.");
109             }
110         }
111     } catch (SQLException e) {
112         e.printStackTrace();
113     }
114 }
```

16. Menambahkan Data Admin kedalam Database

```
131     try {
132         String query = "INSERT INTO admin (idAdmin, username, password, Nama) VALUES (?, ?, ?, ?)";
133         try (PreparedStatement pstmt = connection.prepareStatement(query)) {
134             pstmt.setString(parameterIndex:1, idAdmin);
135             pstmt.setString(parameterIndex:2, username);
136             pstmt.setString(parameterIndex:3, password);
137             pstmt.setString(parameterIndex:4, nama);
138
139             int affectedRows = pstmt.executeUpdate();
140             if (affectedRows > 0) {
141                 System.out.println("Data Admin baru berhasil ditambahkan ke database.");
142             } else {
143                 System.out.println("Gagal menambahkan data Admin.");
144             }
145         }
146     } catch (SQLException e) {
147         e.printStackTrace();
148     }
149 }
```

17. Menambahkan Data Mata Kuliah yang Diajarkan Dosen kedalam Database

```
159     try {
160         String query = "INSERT INTO mataKuliahDosen (nidnDosen, kodeMataKuliah) VALUES (?, ?)";
161         try (PreparedStatement pstmt = connection.prepareStatement(query)) {
162             pstmt.setString(parameterIndex:1, nidnDosen);
163             pstmt.setString(parameterIndex:2, kodeMataKuliah);
164
165             int affectedRows = pstmt.executeUpdate();
166             if (affectedRows > 0) {
167                 System.out.println("Data mata kuliah dosen berhasil ditambahkan ke database.\n");
168             } else {
169                 System.out.println("Gagal menambahkan data mata kuliah dosen.");
170             }
171         }
172     } catch (SQLException e) {
173         e.printStackTrace();
174     }
175 }
```

18. Menghapus Data Mahasiswa

- Pernyataan DELETE FROM dalam SQL digunakan untuk menghapus satu atau lebih baris data dari sebuah tabel

```
public static void hapusDataMahasiswa(Connection
connection) {
    System.out.println("\n--- Hapus Data Mahasiswa ---");

    System.out.print("NIM Mahasiswa yang akan dihapus: ");
    String nim = scanner.nextLine();
```

```
try {
    String queryMahasiswa = "DELETE FROM mahasiswa
WHERE Nim = ?";
    try (PreparedStatement pstmtMahasiswa =
connection.prepareStatement(queryMahasiswa)) {
        pstmtMahasiswa.setString(1, nim);

        int affectedRowsMahasiswa =
pstmtMahasiswa.executeUpdate();

        if (affectedRowsMahasiswa > 0) {
            System.out.println("Data mahasiswa
berhasil dihapus dari tabel mahasiswa.");
        }
    }

    String queryNilaiPerKomponen = "DELETE
FROM nilaiPerKomponen WHERE nimMahasiswa = ?";
    try (PreparedStatement pstmtNilaiPerKomponen =
connection.prepareStatement(queryNilaiPerKomponen)) {
        pstmtNilaiPerKomponen.setString(1,
nim);
        pstmtNilaiPerKomponen.executeUpdate();
    }

    String queryKHS = "DELETE FROM KHS WHERE
nimMahasiswa = ?";
    try (PreparedStatement pstmtKHS =
connection.prepareStatement(queryKHS)) {
        pstmtKHS.setString(1, nim);
        pstmtKHS.executeUpdate();
    }

    String queryKrs = "DELETE FROM Krs WHERE
nimMahasiswa = ?";
    try (PreparedStatement pstmtKrs =
connection.prepareStatement(queryKrs)) {
        pstmtKrs.setString(1, nim);
        pstmtKrs.executeUpdate();
    }

    System.out.println("Data terkait mahasiswa
berhasil dihapus dari tabel lainnya.");
} else {
```

```
        System.out.println("Gagal menghapus data  
mahasiswa. Pastikan NIM valid.");
    }
}
} catch (SQLException e) {
    e.printStackTrace();
}
}
```

19. Menghapus Data Dosen

```
public static void hapusDataDosen(Connection connection) {
    System.out.println("\n--- Hapus Data Dosen ---");

    System.out.print("NIDN Dosen yang akan dihapus: ");
    String nidn = scanner.nextLine();

    try {
        String queryDosen = "DELETE FROM dosen WHERE NIDN  
= ?";
        try (PreparedStatement pstmtDosen =  
connection.prepareStatement(queryDosen)) {
            pstmtDosen.setString(1, nidn);

            int affectedRowsDosen =  
pstmtDosen.executeUpdate();

            if (affectedRowsDosen > 0) {
                System.out.println("Data dosen berhasil  
dihapus dari tabel dosen.");

                String queryPenilaianDosen = "DELETE FROM  
penilaianDosen WHERE nidnDosen = ?";
                try (PreparedStatement pstmtPenilaianDosen  
= connection.prepareStatement(queryPenilaianDosen)) {
                    pstmtPenilaianDosen.setString(1,  
nidn);
                    pstmtPenilaianDosen.executeUpdate();
                }

                String queryMataKuliahDosen = "DELETE FROM  
mataKuliahDosen WHERE nidnDosen = ?";
            }
        }
    }
}
```

```

        try (PreparedStatement
pstmtMataKuliahDosen =
connection.prepareStatement(queryMataKuliahDosen)) {
            pstmtMataKuliahDosen.setString(1,
nidn);
            pstmtMataKuliahDosen.executeUpdate();
        }
        System.out.println("Data terkait dosen
berhasil dihapus dari tabel lainnya.");
    } else {
        System.out.println("Gagal menghapus data
dosen. Pastikan NIDN valid.");
    }
}
} catch (SQLException e) {
    e.printStackTrace();
}
}

```

20. Menghapus Data Admin

```

public static void hapusDataAdmin(Connection connection) {
    System.out.println("\n--- Hapus Data Admin ---");

    System.out.print("ID Admin yang akan dihapus: ");
    String idAdmin = scanner.nextLine();

    try {
        String queryAdmin = "DELETE FROM admin WHERE
idAdmin = ?";
        try (PreparedStatement pstmtAdmin =
connection.prepareStatement(queryAdmin)) {
            pstmtAdmin.setString(1, idAdmin);

            int affectedRowsAdmin =
pstmtAdmin.executeUpdate();

            if (affectedRowsAdmin > 0) {
                System.out.println("Data admin berhasil
dihapus dari tabel admin.");
            } else {

```

```
        System.out.println("Gagal menghapus data
admin. Pastikan ID Admin valid.");
    }
}
} catch (SQLException e) {
    e.printStackTrace();
}
}
```

21. Menghapus Data Mata Kuliah

```
public static void hapusDataMataKuliah(Connection
connection) {
    System.out.println("\n--- Hapus Data Mata Kuliah
---");

    System.out.print("Kode Mata Kuliah yang akan dihapus:
");
    String kodeMataKuliah = scanner.nextLine();

    try {
        String query = "DELETE FROM mataKuliah WHERE
kodeMataKuliah = ?";
        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setString(1, kodeMataKuliah);

            int affectedRows = pstmt.executeUpdate();
            if (affectedRows > 0) {
                System.out.println("\nData mata kuliah
berhasil dihapus dari database.\n");
            } else {
                System.out.println("Gagal menghapus data
mata kuliah. Pastikan kode mata kuliah valid.");
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

22. Menghapus Data Mata Kuliah yang Diajarkan Dosen

```
    public static void hapusMataKuliahDosen(Connection connection) {
        System.out.println("\n--- Hapus Data Mata Kuliah Dosen ---");
        System.out.print("NIDN Dosen: ");
        String nidnDosen = scanner.nextLine();

        System.out.print("Kode Mata Kuliah: ");
        String kodeMataKuliah = scanner.nextLine();

        try {
            String query = "DELETE FROM mataKuliahDosen WHERE
nidnDosen = ? AND kodeMataKuliah = ?";
            try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
                pstmt.setString(1, nidnDosen);
                pstmt.setString(2, kodeMataKuliah);

                int rowCount = pstmt.executeUpdate();
                if (rowCount > 0) {
                    System.out.println("Data mata kuliah dosen
berhasil dihapus.\n");
                } else {
                    System.out.println("Data mata kuliah dosen
tidak ditemukan.");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

23. Melihat Data Mahasiswa

```

9  public class TampilkanData {
10    public static void tampilkanDataMahasiswa(Connection connection) {
11      try {
12        String query = "SELECT * FROM mahasiswa ORDER BY Nim";
13        try (PreparedStatement pstmt = connection.prepareStatement(query)) {
14          ResultSet resultSet = pstmt.executeQuery();
15
16          System.out.println(x:"\nData Mahasiswa:");
17          System.out.println(x:"-----");
18          System.out.printf(format:"|%-15s|%-30s|%-20s|%-10s|\n", ...args:Nim, Nama, Prodi, Angkatan);
19          System.out.println(x:"|-----|-----|-----|-----|");
20
21          while (resultSet.next()) {
22            String nim = resultSet.getString(columnLabel:"nim");
23            String nama = resultSet.getString(columnLabel:"nama");
24            String prodi = resultSet.getString(columnLabel:"prodi");
25            int angkatan = resultSet.getInt(columnLabel:"angkatan");
26
27            System.out.printf(format:"|%-15s|%-30s|%-20s|%-10d|\n", nim, nama, prodi, angkatan);
28          }
29        } catch (SQLException e) {
30          e.printStackTrace();
31        }
32      }
33    }

```

24. Melihat Data Dosen

```

34
35  public static void tampilkanDataDosen(Connection connection) {
36    try {
37      String query = "SELECT * FROM dosen ORDER BY NIDN";
38      try (PreparedStatement pstmt = connection.prepareStatement(query)) {
39        ResultSet resultSet = pstmt.executeQuery();
40
41        System.out.println(x:"\nData Dosen:");
42        System.out.println(x:"-----");
43        System.out.printf(format:"|%-15s|%-30s|\n", ...args:NIDN, Nama);
44        System.out.println(x:"|-----|-----|");
45
46        while (resultSet.next()) {
47          String nidn = resultSet.getString(columnLabel:"nidn");
48          String nama = resultSet.getString(columnLabel:"nama");
49
50          System.out.printf(format:"|%-15s|%-30s|\n", nidn, nama);
51        }
52      } catch (SQLException e) {
53        e.printStackTrace();
54      }
55    }
56  }

```

25. Melihat Data Admin

```

83  public static void tampilkanDataAdmin(Connection connection) {
84    try {
85      String query = "SELECT * FROM admin ORDER BY idAdmin";
86      try (PreparedStatement pstmt = connection.prepareStatement(query)) {
87        ResultSet resultSet = pstmt.executeQuery();
88
89        System.out.println(x:"\nData Admin:");
90        System.out.println(x:"-----");
91        System.out.printf(format:"|%-15s|%-30s|\n", ...args:ID Admin, Nama);
92        System.out.println(x:"|-----|-----|");
93
94        while (resultSet.next()) {
95          String idAdmin = resultSet.getString(columnLabel:"idAdmin");
96          String nama = resultSet.getString(columnLabel:"nama");
97
98          System.out.printf(format:"|%-15s|%-30s|\n", idAdmin, nama);
99        }
100      } catch (SQLException e) {
101        e.printStackTrace();
102      }
103    }
104  }

```

26. Melihat Data Mata Kuliah

```

58     public static void tampilanDataMataKuliah(Connection connection) {
59         try {
60             String query = "SELECT kodeMataKuliah, namaMataKuliah, sks FROM mataKuliah "
61                 + "ORDER BY CAST(SUBSTRING(kodeMataKuliah, 3) AS SIGNED)";
62             try (PreparedStatement pstmt = connection.prepareStatement(query)) {
63                 ResultSet resultSet = pstmt.executeQuery();
64
65                 System.out.println(x:"\nData Mata Kuliah:");
66                 System.out.println(x:"-----");
67                 System.out.printf(format:"%-15s|%-40s|%-5s|\n", ...args:"Kode", "Nama Mata Kuliah", "SKS");
68                 System.out.println(x:"|-----|-----|-----|");
69
70                 while (resultSet.next()) {
71                     String kodeMataKuliah = resultSet.getString(columnLabel:"kodeMataKuliah");
72                     String namaMataKuliah = resultSet.getString(columnLabel:"namaMataKuliah");
73                     int sks = resultSet.getInt(columnLabel:"sks");
74
75                     System.out.printf(format:"%-15s|%-40s|%-5d|\n", kodeMataKuliah, namaMataKuliah, sks);
76                 }
77             } catch (SQLException e) {
78                 e.printStackTrace();
79             }
80         }
81     }

```

27. Melihat Data Mata Kuliah yang Diajarkan Dosen

```

106    public static void lihatMataKuliahDosen(Connection connection) {
107        try {
108            String query = "SELECT mkd.kodeMataKuliah, mk.namaMataKuliah, mkd.nidnDosen, d.Nama "
109                + "FROM mataKuliahDosen mkd "
110                + "JOIN mataKuliah mk ON mkd.kodeMataKuliah = mk.kodeMataKuliah "
111                + "JOIN dosen d ON mkd.nidnDosen = d.NIDN "
112                + "ORDER BY CAST(SUBSTRING(mkd.kodeMataKuliah FROM 3) AS UNSIGNED)";
113            try (PreparedStatement pstmt = connection.prepareStatement(query)) {
114                ResultSet resultSet = pstmt.executeQuery();
115
116                System.out.println(x:"\nMata Kuliah yang Diajarkan Dosen");
117                System.out.println(
118                    x:"-----");
119                System.out.printf(format:"|%-15s|%-35s|%-15s|%-30s|\n", ...args:"Kode", "Nama Mata Kuliah", "NIDN", "Nama Dosen");
120                System.out.println(
121                    x:"|-----|-----|-----|-----|");
122
123                while (resultSet.next()) {
124                    String kode = resultSet.getString(columnLabel:"kodeMataKuliah");
125                    String namaMatakuliah = resultSet.getString(columnLabel:"namaMataKuliah");
126                    String nidnDosen = resultSet.getString(columnLabel:"nidnDosen");
127                    String namaDosen = resultSet.getString(columnLabel:"Nama");
128
129                    System.out.printf(format:"|%-15s|%-35s|%-15s|%-30s|\n", kode, namaMatakuliah, nidnDosen, namaDosen);
130                }
131            }
132        }
133    }

```

F. Penggunaan Fitur

1. Tampilan Awal

Username: Mahasiswa
Password: [redacted]

2. Tampilan Jika Login Sebagai Mahasiswa Berhasil

Username: Mahasiswa
Password: NIM123

Anda berhasil login sebagai mahasiswa

Nama : Mahasiswa 1
NIM : NIM123
Prodi : Teknik Informatika
Angkatan: 2023

Selamat datang, Mahasiswa 1
Silahkan pilih salah satu menu berikut:

1. Melihat KRS
2. Lihat KHS
3. Pilih KRS
4. Logout

Masukkan pilihan Anda: [redacted]

3. Tampilan Jika Login Sebagai Dosen Berhasil

```
Username: Dosen  
Password: NIDN123

Anda berhasil login sebagai dosen

Nama: Dosen 1  
NIDN: NIDN123

Selamat datang, Dosen 1

Silahkan pilih salah satu menu berikut:  
1. Lihat daftar mata kuliah yang diajar  
2. Atur Komponen Penilaian  
3. Lihat Komponen Penilaian  
4. Input nilai mahasiswa  
5. Logout  
Masukkan pilihan Anda: █
```

4. Tampilan Jika Login Sebagai Admin Berhasil

```
Username: Admin  
Password: pass1

Anda berhasil login sebagai admin

Nama : Nama admin1  
ID Admin: 12345

Selamat datang, Nama admin1  
Silahkan pilih salah satu menu berikut:  
1. Tampilkan data  
2. Tambahkan data  
3. Hapus data  
4. Logout  
Masukkan pilihan Anda: █
```

5. Tampilan Jika Login Gagal

```
Username: Mahasiswa  
Password: nim123

Login gagal. Periksa kembali username dan password.
```

6. Tampilan Jika Mahasiswa Ingin Menampilkan KRS

```
Silahkan pilih salah satu menu berikut:  
1. Melihat KRS  
2. Lihat KHS  
3. Pilih KRS  
4. Logout  
Masukkan pilihan Anda: 1

Kartu Rencana Studi Mahasiswa 1

Kode      Nama Mata Kuliah      SKS
MK1       Aljabar Linear        3
MK2       Interaksi Manusia dan Komputer 3
MK3       Pemrograman Berorientasi Objek 3
MK6       Jaringan Komputer       3
MK7       Technopreneurship     2
MK8       Riset teknologi informasi 2
MK9       Transformasi Digital   3
MK10      Supply Chain Management 2
MK12      Matematika Diskrit    3

Jumlah SKS: 24 SKS

Silahkan pilih salah satu menu berikut:
```

7. Tampilan Jika Mahasiswa Ingin Melihat KRS, tetapi KRS Belum Ada

Kartu Rencana Studi Mahasiswa 1
KRS dengan nim NIM123 tidak ditemukan.

Ingin memilih KRS? (y/n):

8. Tampilan Jika Mahasiswa Memilih KRS

Daftar Mata Kuliah:

MK1 - Aljabar Linear (SKS: 3)
MK2 - Interaksi Manusia dan Komputer (SKS: 3)
MK3 - Pemrograman Berorientasi Objek (SKS: 3)
MK4 - Sistem Basis Data (SKS: 3)
MK5 - Sistem Operasi (SKS: 3)
MK6 - Jaringan Komputer (SKS: 3)
MK7 - Technopreneurship (SKS: 2)
MK8 - Riset teknologi informasi (SKS: 2)
MK9 - Transformasi Digital (SKS: 3)
MK10 - Supply Chain Management (SKS: 2)
MK11 - Teori Peluang (SKS: 3)
MK12 - Matematika Diskrit (SKS: 3)
MK13 - Persamaan Diferensial Biasa (SKS: 3)
MK14 - Analisis Suku Bunga Berbasis Web (SKS: 3)
MK15 - Matematika Bisnis dan Teknologi (SKS: 3)
MK16 - Kalkulus lanjutan (SKS: 3)

Total SKS: 0

Pilih mata kuliah (0/n/no untuk berhenti):

9. Tampilan Jika Mata Kuliah Berhasil Ditambahkan kedalam KRS

```
Total SKS: 6
Pilih mata kuliah (0/n/no untuk berhenti): MK3
Mata kuliah berhasil ditambahkan.

Total SKS: 9
Pilih mata kuliah (0/n/no untuk berhenti): MK6
Mata kuliah berhasil ditambahkan.

Total SKS: 12
Pilih mata kuliah (0/n/no untuk berhenti): MK7
Mata kuliah berhasil ditambahkan.

Total SKS: 14
Pilih mata kuliah (0/n/no untuk berhenti): MK8
Mata kuliah berhasil ditambahkan.

Total SKS: 16
Pilih mata kuliah (0/n/no untuk berhenti): MK9
Mata kuliah berhasil ditambahkan.

Total SKS: 19
Pilih mata kuliah (0/n/no untuk berhenti): MK10
Mata kuliah berhasil ditambahkan.

Total SKS: 21
Pilih mata kuliah (0/n/no untuk berhenti): MK12
Mata kuliah berhasil ditambahkan.

Total SKS: 24
Pilih mata kuliah (0/n/no untuk berhenti): ■
```

```
Total SKS: 24
Pilih mata kuliah (0/n/no untuk berhenti): 0
Terima kasih! Daftar mata kuliah Anda telah disimpan.
```

10. Tampilan Jika KRS yang Dipilih Lebih dari 24 sks

```
Total SKS: 21
Pilih mata kuliah (0/n/no untuk berhenti): MK12
Mata kuliah berhasil ditambahkan.

Total SKS: 24
Pilih mata kuliah (0/n/no untuk berhenti): MK13
Jumlah SKS melebihi batas maksimal.
```

11. Tampilan Jika Mahasiswa Melihat KHS

```
Masukkan pilihan Anda: 2

Kartu Hasil Studi (KHS)
Nama: Mahasiswa 1
NIM : NIM123
-----
|KODE|MATAKULIAH| NILAI| NILAI HURUF|
|----|-----|-----|-----|
|MK1|Aljabar Linear| 86.9| A|
```

12. Tampilan Jika Mahasiswa Ingin Melihat KHS, tetapi KHS Belum Ada

```
KHS dengan Nama Mahasiswa 1 dan NIM NIM123 tidak ditemukan.
```

13. Tampilan Jika Dosen Tidak Mengajar Mata Kuliah Apapun

Dosen dengan Nama Dosen 1 tidak mengajar mata kuliah apapun.

14. Tampilan Jika Dosen Ingin Menampilkan Mata Kuliah yang Diajarkan

Masukkan pilihan Anda: 1

Daftar Mata Kuliah yang Diajar oleh NIDM123

Kode	Nama Mata Kuliah
MK1	Aljabar Linear
MK12	Matematika Diskrit
MK16	Kalkulus Lanjutan
MK6	Jaringan Komputer

Silahkan pilih salah satu menu berikut:

1. Lihat daftar mata kuliah yang diajar
2. Atur Komponen Penilaian
3. Lihat Komponen Penilaian
4. Input nilai mahasiswa
5. Logout

Masukkan pilihan Anda:

15. Tampilan Jika Dosen Ingin Mengatur Komponen Penilaian

Daftar Mata Kuliah yang Diajar:

1. Aljabar Linear
2. Matematika Diskrit
3. Kalkulus Lanjutan
4. Jaringan Komputer

Pilih Mata Kuliah untuk Mengatur Komponen Penilaian: 1

Mengatur komponen penilaian Aljabar Linear

Masukkan Jumlah Komponen Penilaian untuk Mata Kuliah ini: 3

Masukkan Nama Komponen Penilaian ke-1: Kehadiran

Masukkan Persentase Nilai untuk Komponen Penilaian ke-1: 30

Masukkan Nama Komponen Penilaian ke-2: Kuis

Masukkan Persentase Nilai untuk Komponen Penilaian ke-2: 50

Masukkan Nama Komponen Penilaian ke-3: UAS

Masukkan Persentase Nilai untuk Komponen Penilaian ke-3: 20

Komponen penilaian berhasil diatur untuk matakuliah Aljabar Linear

Silahkan pilih salah satu menu berikut:

16. Tampilan Jika Dosen Ingin Melihat Komponen Penilaian yang sudah diatur

Daftar Mata Kuliah yang Diajarkan:

1. Aljabar Linear
2. Matematika Diskrit
3. Kalkulus Lanjutan
4. Jaringan Komputer

Pilih Mata Kuliah yang ingin dilihat komponen penilaianya: 1

Komponen Penilaian untuk Mata Kuliah: Aljabar Linear

Komponen Penilaian	Persentase
Kehadiran	30.00
Kuis	50.00
UAS	20.00

17. Tampilan Jika Dosen Ingin Menginput Nilai Apabila Komponen Penilaian Tidak Ada

Mata Kuliah : Matematika Diskrit
Jumlah Mahasiswa: 1
Tidak ada komponen penilaian yang terdaftar untuk mata kuliah ini.
Masukkan Jumlah Komponen Penilaian untuk Mata Kuliah ini: |

18. Tampilan Jika Dosen Membatalkan untuk Mengisi Komponen Penilaian

Mata Kuliah : Matematika Diskrit
Jumlah Mahasiswa: 1
Tidak ada komponen penilaian yang terdaftar untuk mata kuliah ini.
Masukkan Jumlah Komponen Penilaian untuk Mata Kuliah ini: 0
Total persentase tidak sama dengan 100%. Pembatalan pengaturan komponen penilaian.
Nilai berhasil disimpan.

19. Tampilan Jika Dosen Ingin Menginput Nilai

Pilih Mata Kuliah untuk Menginput nilai: 1

Mata Kuliah : Aljabar Linear

Jumlah Mahasiswa: 2

Nama Mahasiswa: Mahasiswa 3

NIM : NIM789

Masukkan Nilai untuk Kehadiran: 90

Masukkan Nilai untuk Kuis: 70

Masukkan Nilai untuk UAS: 88

Nama Mahasiswa: Mahasiswa 1

NIM : NIM123

Masukkan Nilai untuk Kehadiran: 100

Masukkan Nilai untuk Kuis: 77

Masukkan Nilai untuk UAS: 92

Nilai berhasil disimpan.

20. Tampilan Jika Admin Ingin Menampilkan Data Mahasiswa

Data Mahasiswa:			
Nim	Nama	Prodi	Angkatan
NIM101	Mahasiswa 4	Teknik Elektro	2022
NIM123	Mahasiswa 1	Teknik Informatika	2023
NIM202	Mahasiswa 5	Teknik Mesin	2023
NIM303	Mahasiswa 6	Matematika	2022
NIM456	Mahasiswa 2	Sistem Informasi	2022
NIM789	Mahasiswa 3	Ilmu Komputer	2022

21. Tampilan Jika Admin Ingin Menampilkan Data Dosen

Data Dosen:	
NIDN	Nama
NIDN101	Dosen 4
NIDN123	Dosen 1
NIDN202	Dosen 5
NIDN456	Dosen 2
NIDN789	Dosen 3

22. Tampilan Jika Admin Ingin Menampilkan Data Admin

Data Admin:	
ID Admin	Nama
12345	Nama admin1
67891	Nama admin2

23. Tampilan Jika Admin Ingin Menampilkan Data Mata Kuliah

Data Mata Kuliah:

Kode	Nama Mata Kuliah	SKS
MK1	Aljabar Linear	3
MK2	Interaksi Manusia dan Komputer	3
MK3	Pemrograman Berorientasi Objek	3
MK4	Sistem Basis Data	3
MK5	Sistem Operasi	3
MK6	Jaringan Komputer	3
MK7	Technopreneurship	2
MK8	Riset teknologi informasi	2
MK9	Transformasi Digital	3
MK10	Supply Chain Management	2
MK11	Teori Peluang	3
MK12	Matematika Diskrit	3
MK13	Persamaan Diferensial Biasa	3
MK14	Analisis Suku Bunga Berbasis Web	3
MK15	Matematika Bisnis dan Teknologi	3
MK16	Kalkulus lanjutan	3

24. Tampilan Jika Admin Ingin Menampilkan Data Seluruh Mata Kuliah yang Diajarkan Dosen

Mata Kuliah yang Diajarkan Dosen

Kode	Nama Mata Kuliah	NIDN	Nama Dosen
MK1	Aljabar Linear	NIDN123	Dosen 1
MK2	Interaksi Manusia dan Komputer	NIDN456	Dosen 2
MK3	Pemrograman Berorientasi Objek	NIDN789	Dosen 3
MK4	Sistem Basis Data	NIDN101	Dosen 4
MK5	Sistem Operasi	NIDN202	Dosen 5
MK6	Jaringan Komputer	NIDN123	Dosen 1
MK7	Technopreneurship	NIDN456	Dosen 2
MK8	Riset teknologi informasi	NIDN789	Dosen 3
MK9	Transformasi Digital	NIDN101	Dosen 4
MK10	Supply Chain Management	NIDN202	Dosen 5
MK11	Teori Peluang	NIDN456	Dosen 2
MK12	Matematika Diskrit	NIDN123	Dosen 1
MK13	Persamaan Diferensial Biasa	NIDN789	Dosen 3
MK14	Analisis Suku Bunga Berbasis Web	NIDN101	Dosen 4
MK15	Matematika Bisnis dan Teknologi	NIDN202	Dosen 5
MK16	Kalkulus lanjutan	NIDN123	Dosen 1

25. Tampilan Jika Admin Ingin Menambah Data Mahasiswa

```
Menu tambah data
1. Tambah Data Mahasiswa
2. Tambah Data Dosen
3. Tambah Data Admin
4. Tambah Data Mata Kuliah
5. Tambah Mata Kuliah yang Diajarkan Dosen
Masukkan pilihan Anda: 1

--- Tambah Data Mahasiswa ---
Username: andif
Password: def
Nama      : Andi Magfirah Maqbul
NIM       : 221011048
Prodi    : Ilmu Komputer
Angkatan: 2022
Data mahasiswa berhasil ditambahkan ke database.
```

26. Tampilan Jika Admin Ingin Menambah Data Dosen

```
--- Tambah Data Dosen ---
Username: Dosen
Password: tes
NIDN     : Dahlah
Nama      : Dosen intinya
Data dosen berhasil ditambahkan ke database.
```

27. Tampilan Jika Admin Ingin Menambah Data Admin

```
--- Tambah Data Admin ---
Id Admin: 1357
Username: Admin
Password: ytta
Nama     : Dikri
Data Admin baru berhasil ditambahkan ke database.
```

28. Tampilan Jika Admin Ingin Menambah Data Mata Kuliah yang Diajarkan Dosen

```
--- Tambah Data Mata Kuliah ---
Kode Mata Kuliah: MK17
Nama Mata Kuliah: SBD
SKS            : 3
Data matakuliah berhasil ditambahkan ke database.
```

29. Tampilan Jika Admin Ingin Menghapus Data Mahasiswa

```
--- Hapus Data Mahasiswa ---
NIM Mahasiswa yang akan dihapus: 221011048
Data mahasiswa berhasil dihapus dari tabel mahasiswa.
Data terkait mahasiswa berhasil dihapus dari tabel lainnya.
```

30. Tampilan Jika Admin Ingin Menghapus Data Dosen

```
--- Hapus Data Dosen ---  
NIDN Dosen yang akan dihapus: Dahlah  
Data dosen berhasil dihapus dari tabel dosen.  
Data terkait dosen berhasil dihapus dari tabel lainnya.
```

31. Tampilan Jika Admin Ingin Menghapus Data Admin

```
--- Hapus Data Admin ---  
ID Admin yang akan dihapus: 1357  
Data admin berhasil dihapus dari tabel admin.
```

32. Tampilan Jika Admin Ingin Menghapus Data Mata Kuliah yang Diajarkan Dosen

```
--- Hapus Data Mata Kuliah Dosen ---  
NIDN Dosen: NIDN123  
Kode Mata Kuliah: MK16  
Data mata kuliah dosen berhasil dihapus.
```

33. Tampilan Logout

```
4. Logout  
Masukkan pilihan Anda: 4  
  
Anda berhasil logout.
```

BAB IV

PENUTUP

A. Kesimpulan

Portal akademik merupakan inovasi di dunia pendidikan yang bertujuan memberikan kemudahan dan efisiensi dalam manajemen informasi akademik. Portal ini dikembangkan oleh kelompok kami dengan basis data yang dirancang menggunakan beberapa tabel utama seperti mahasiswa, dosen, mata kuliah, KRS, KHS, dan lainnya. Basis data ini kemudian diimplementasikan dengan bahasa pemrograman Java yang terhubung dengan database tersebut.

Fitur utama portal akademik kami antara lain mahasiswa dapat melihat KRS dan KHS, dosen dapat melihat jadwal mengajar dan melakukan penginputan nilai, serta admin memiliki akses penuh untuk melakukan manajemen data master seperti menambah, menghapus, dan melihat data mahasiswa, dosen, mata kuliah, dan lainnya. Kami melakukan pengujian terhadap fungsionalitas portal ini dan berhasil memastikan bahwa fitur-fitur tersebut dapat berjalan dengan baik sesuai dengan hak akses masing-masing pengguna.

Secara keseluruhan, portal akademik sederhana yang kami bangun telah berhasil memenuhi tujuan awal yaitu memberikan kemudahan akses informasi akademik bagi mahasiswa, dosen dan admin. Namun demikian, portal ini masih memiliki peluang pengembangan lebih lanjut dengan menambahkan fitur-fitur pendukung lainnya di masa mendatang.

Code of Conduct:

Kami menyatakan bahwa Tugas Kelompok ini kami kerjakan dengan usaha kami sendiri. Kami tidak menyalin jawaban dari sumber manapun. Kami bertanggung jawab menjaga agar jawaban Tugas Kelompok kami tidak disalin oleh peserta kelompok lainnya.

Proporsi penggerjaan Tugas Kelompok adalah sbb:

NIM	Nama Anggota	Deskripsi Tugas	Kontribusi(%)
221011048	Andi Magfirah Maqbul	<ul style="list-style-type: none">• Mengerjakan baris kode Program• Mengerjakan Laporan Akhir• Mengerjakan ERD	40
221011057	Dikriani	<ul style="list-style-type: none">• Mengerjakan Laporan Akhir• Mengerjakan ERD dan Skema Diagram	30
221011053	Garizah Fauziyah Humaima	<ul style="list-style-type: none">• Mengerjakan Laporan Akhir• Mengerjakan ERD	12
221011017	Widya Puspita Sari	<ul style="list-style-type: none">• Mengerjakan Laporan Akhir• Mengerjakan ERD dan Skema Diagram	18