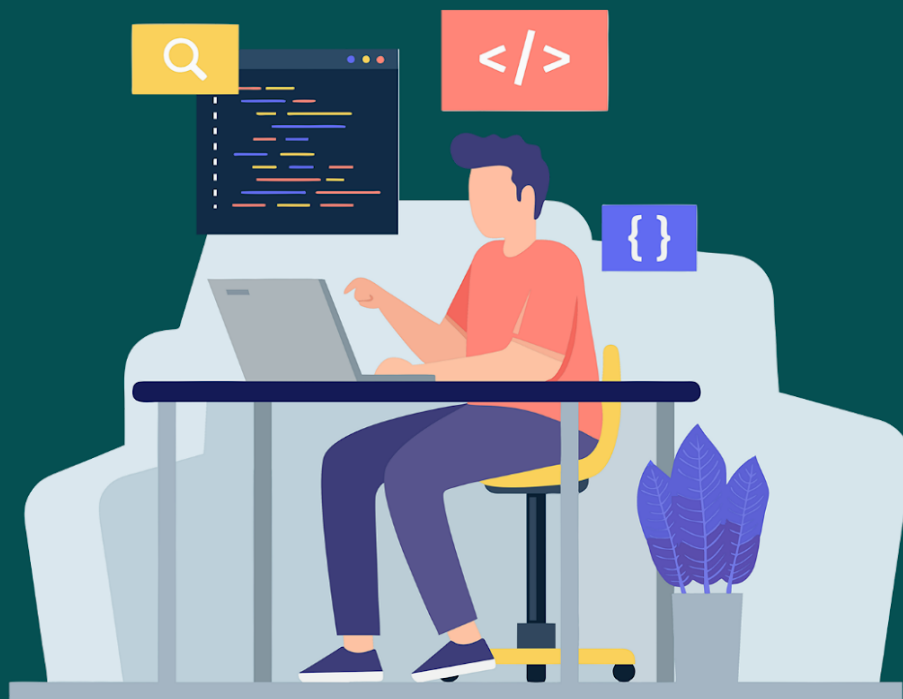


Modul Praktikum
Desain dan Pemrograman Berorientasi Objek

INTRODUCTION TO JAVA GUI: DATABASE



TIM ASISTEN PEMROGRAMAN
ANGKATAN 12

Departemen Pendidikan Ilmu Komputer
Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam
Universitas Pendidikan Indonesia
2023

Introduction

Program Java pada modul sebelumnya masih menyimpan data secara temporary di RAM, yang mana ketika program ditutup maka data akan direset. Oleh karena itu dibutuhkan database untuk membuat sistem yang dapat menyimpan data secara permanen. Terdapat banyak sekali database yang bisa digunakan, diantaranya MySQL, SQLite, PostgreSQL, Ms. SQL Server, Oracle, MongoDB, dan sebagainya. Pada pertemuan ini database yang akan digunakan adalah MySQL.

Untuk menghubungkan Java dengan MySQL dibutuhkan JDBC atau Java Database Connectivity. JDBC bertugas menyediakan koneksi ke database sehingga datanya dapat diakses dan dikelola menggunakan program Java. Ada beberapa istilah dalam JDBC, yaitu:

1. DriverManager : sebuah class untuk mengelola driver
2. Driver : interface untuk handle koneksi dengan database
3. Connection : interface yang memuat method untuk menghubungkan dengan database
4. Statement : interface untuk mengeksekusi query
5. ResultSet : interface untuk menampung data hasil query
6. SQLException : class yang menyediakan informasi error di database

Preparation

Sebelum memulai Praktikum, ada beberapa tools yang harus disiapkan, yaitu:

1. Download connector MySQL pada link [ini](#).
Untuk Versi yang digunakan adalah versi 8.0.28

mysql-connector-java from group mysql (version 8.0.28)
JDBC Type 4 driver for MySQL

Q Group: mysql

Q Artifact: mysql-connector-java

40641 downloads

Show source

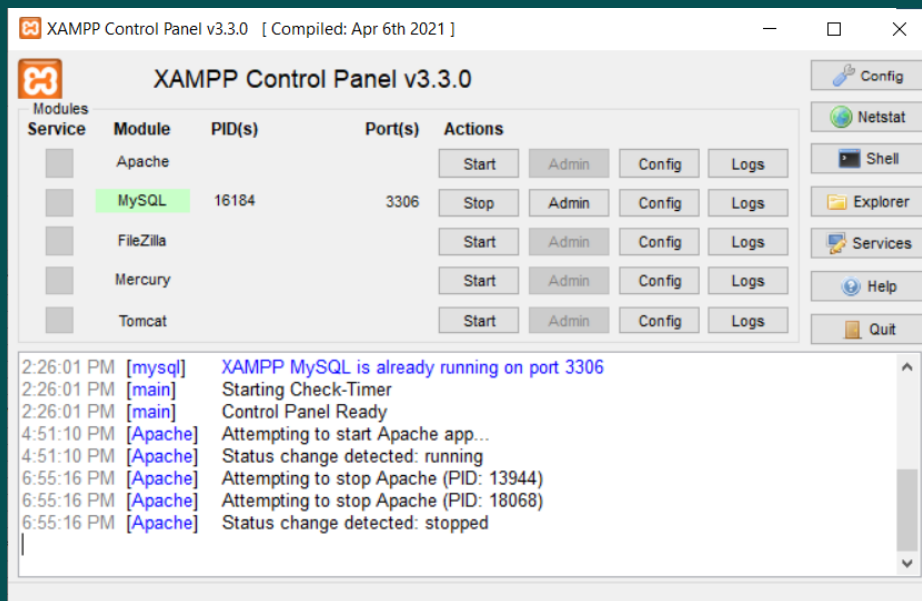
Show build tool code

Download mysql-connector-java.jar (8.0.28)

Add to Project

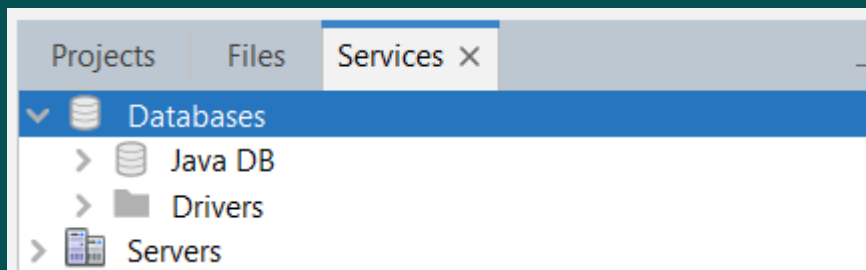
Artifact mysql-connector-java
Group mysql
Version 8.0.28
Last update 16. December 2021
Tags: driver type jdbc mysql
Organization Oracle Corporation
URL <http://dev.mysql.com/doc/connector-j/en/>
License The GNU General Public License, v2 with FOSS exception
Dependencies amount 1
Dependencies protobuf-java,
There are maybe transitive dependencies!

2. Download Driver JDBC pada link [ini](#).
3. Download database pada link [ini](#).
4. Buka xampp, jalankan MySQL.

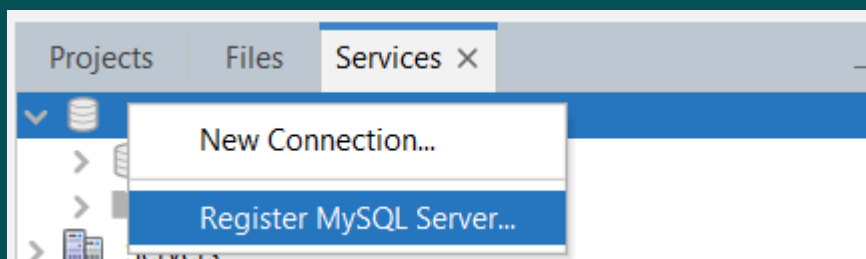


Database Configuration

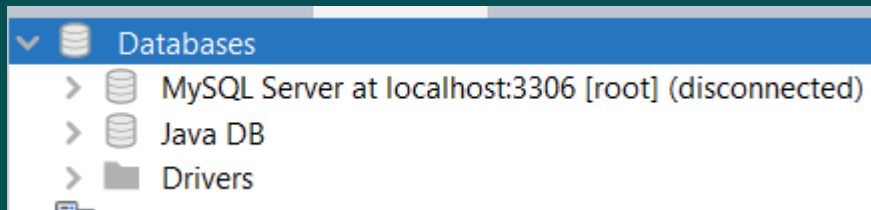
1. Buka **Tab Services** di Netbeans IDE
Jika setting masih default, bisa dengan menekan **ctrl + 5**



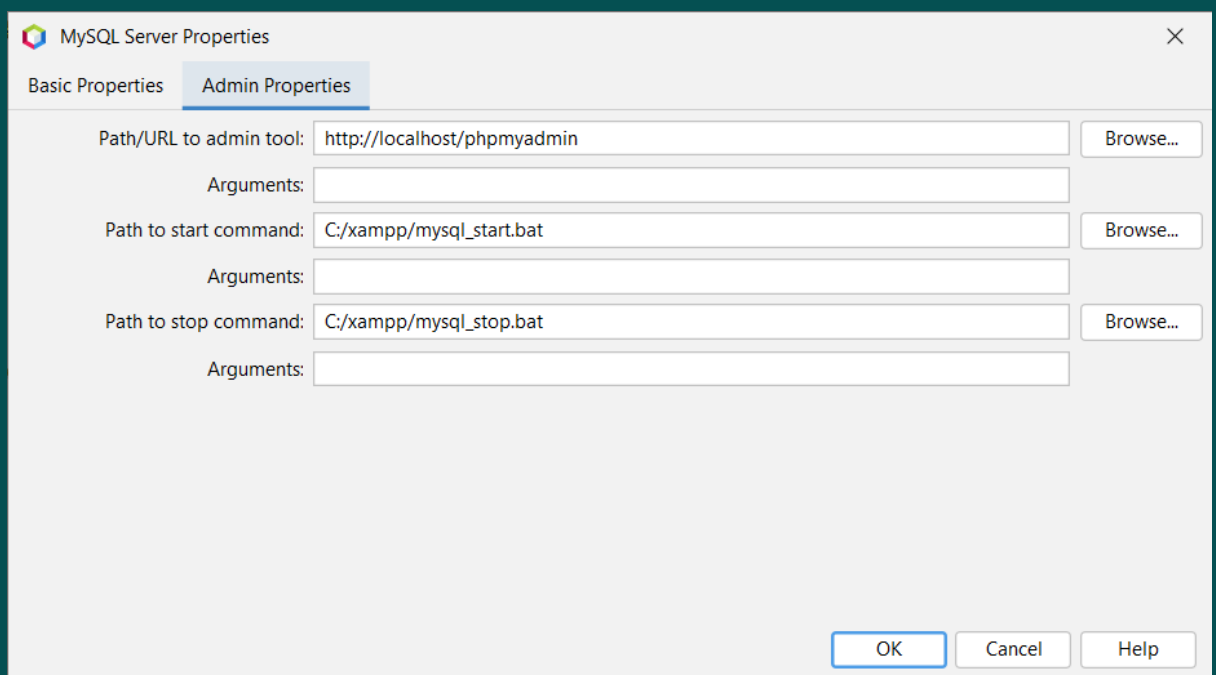
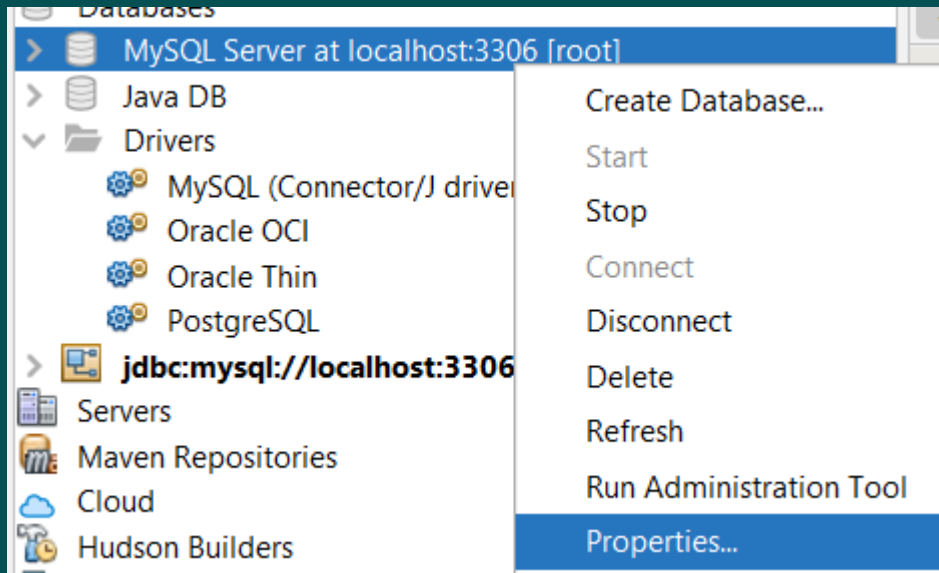
2. Klik Kanan **Databases > Register MySQL Serber...**



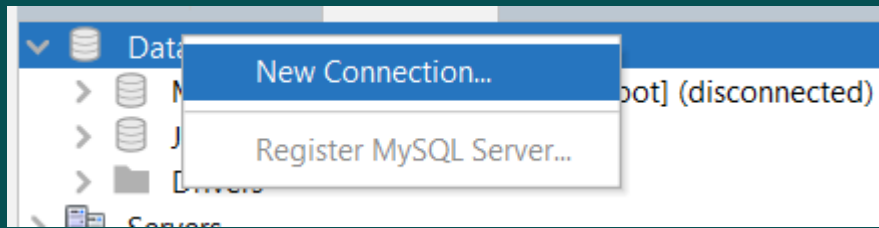
3. Nantinya akan muncul **MySQL Server ...**



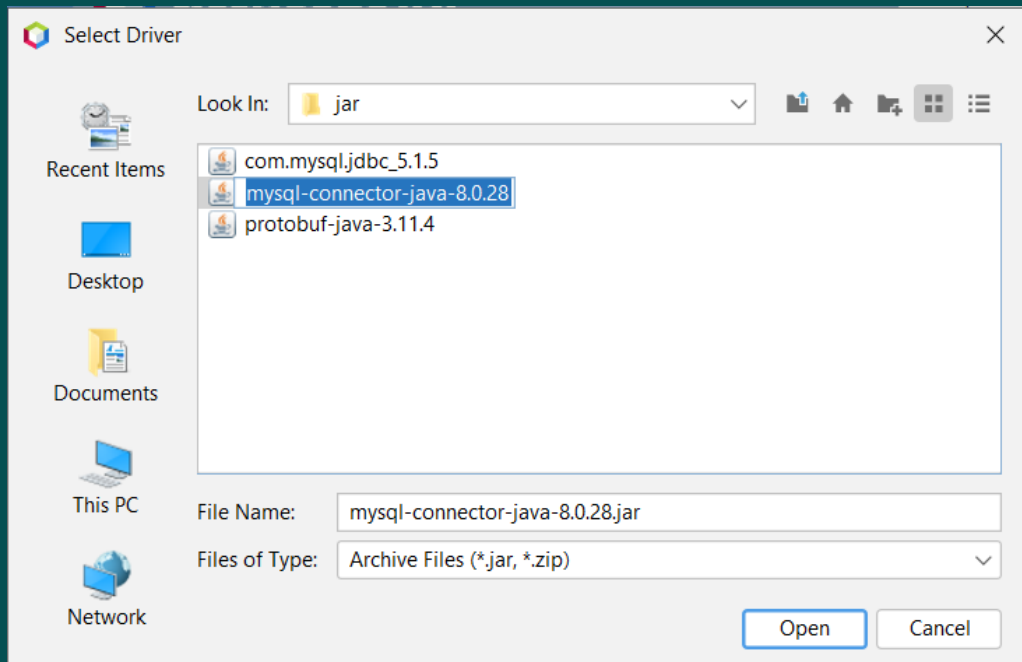
4. Klik Kanan MySQL Server > properties... dan pilih tab Admin Property lalu isi seperti dibawah ini



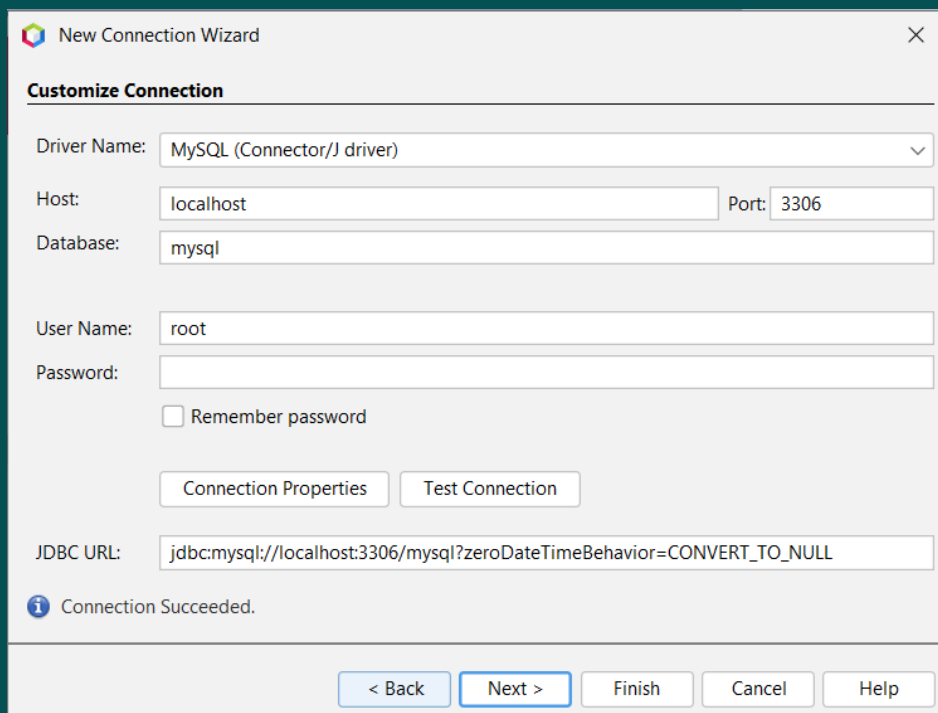
5. Klik Kanan **Databases > New Connection**



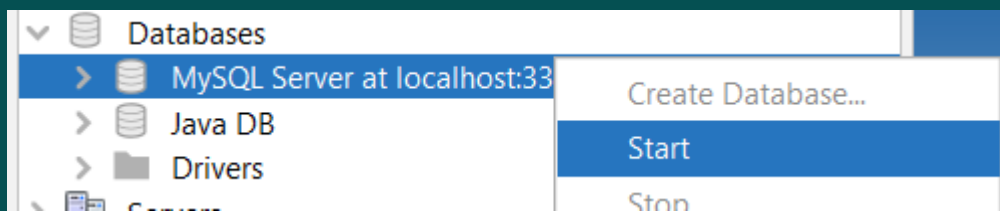
6. Lalu Add **mysql-connector-java-8.0.28.jar** yang sudah di download



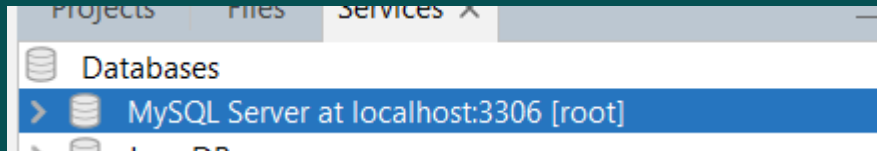
7. Pada Window ini, tekan tombol **Test Connection**, Jika tidak ada kendala maka akan muncul informasi Connection succeeded. Next - Next aja terus finish abis itu.



8. Klik Kanan **MySQL Server** > **Start**



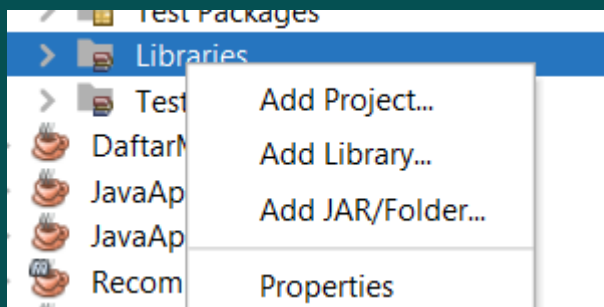
9. Tulisan (Disconnected) akan hilang



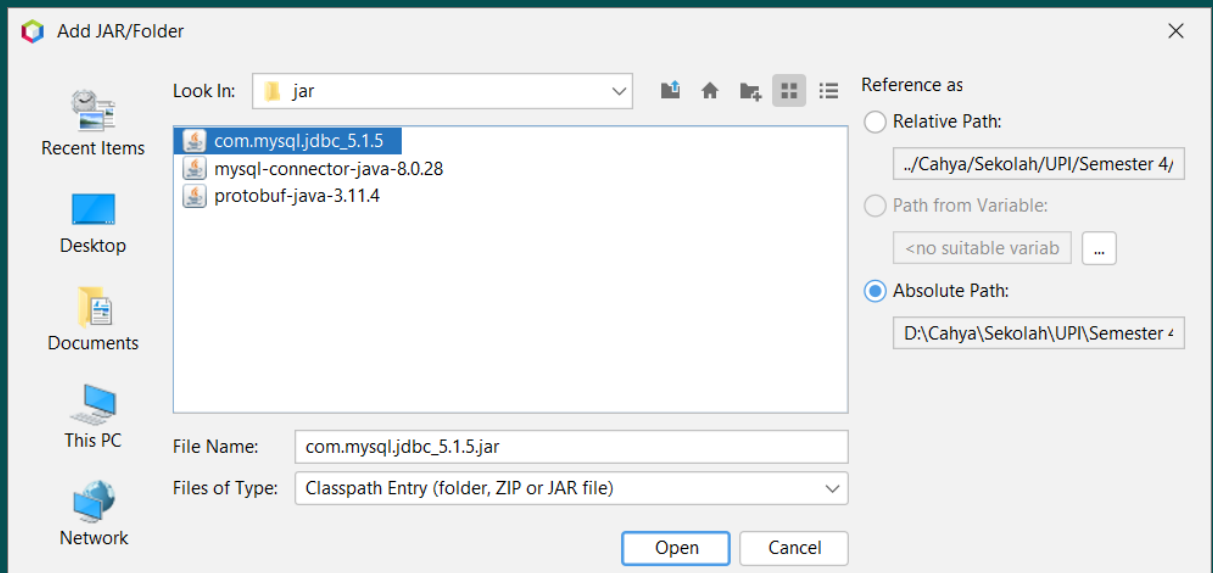
Modify Previous Code

Buka kembali project **DaftarMahasiswa** pada praktikum ke 5

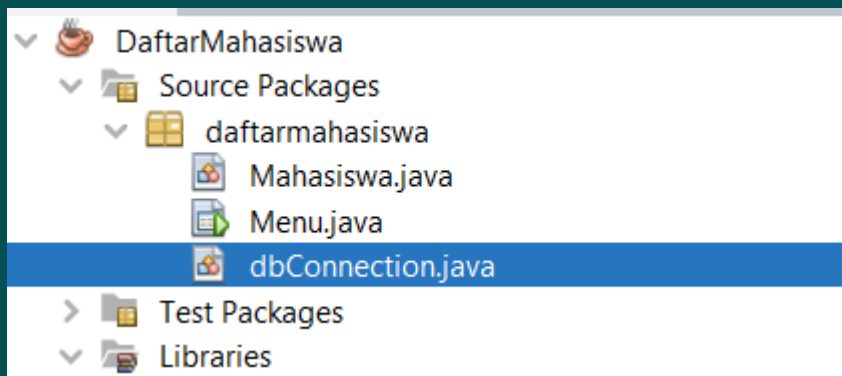
1. Klik Kanan **Libraries** > **Add JAR/Folder**



2. Tambahkan Library **mysql jdbc** yang telah didownload



3. Buat File Java Class dengan nama **dbConnection**



dbConnction.java - import

```
package daftarmahasiswa;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
```

dbConnection.java - property & constructor

```
public class dbConnection {
    private Statement stmt = null;
    private Connection conn = null;

    public dbConnection() {
        String ConAddress = "jdbc:mysql://localhost/db_latihan_dpo";
        String user = "root";
        String pass = "";
        connect(ConAddress, username:user, pass);
    }
}
```

dbConnection.java - connect method

```
private void connect(String ConAddress, String username, String pass){
    try{
        Class.forName( className: "com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection( url: ConAddress, user: username, password: pass);
        stmt = conn.createStatement();
    } catch (ClassNotFoundException | SQLException ex){
        System.out.println( x: ex.getMessage());
    }
}
```

dbConnction.java - selectQuery method

```
// for SELECT query
public ResultSet selectQuery(String sql) {
    try {
        stmt.executeQuery( string: sql);
        return stmt.getResultSet();
    } catch (SQLException ex) {
        System.out.println( x:ex.getMessage());
    }
    return null;
}
```


dbConnction.java - updateQuery method

```
// for INSERT, UPDATE, DELETE query
public int updateQuery(String sql){
    try {
        return stmt.executeUpdate( string: sql);
    } catch (SQLException ex) {
        System.out.println( x:ex.getMessage());
    }
    return 0;
}
```

dbConnction.java - getStatement method

```
public Statement getStatement(){
    return stmt;
}
```

4. Ubah code pada Menu.java



```

private DefaultTableModel dtm;
private Boolean isUpdate = false;
private int selectedID = -1;
private ArrayList<Mahasiswa> listMhs;
private dbConnection db;

/**
 * Creates new form Menu
 */
public Menu() {
    initComponents();
    // make connection to db
    db = new dbConnection();
    // Set Table
    tblMhs.setModel( dataModel: setTable());

    // Hide Delete button
    btnDelete.setVisible( aFlag: false);
}

```

Menu.java - setTable

```

public final DefaultTableModel setTable() {
    Object[] column = {"NIM", "Nama", "Nilai"};
    DefaultTableModel dataTabel = new DefaultTableModel( data: null, columnNames: column);
    ResultSet res = db.selectQuery( sql: "SELECT * FROM mahasiswa");
    try {
        while(res.next()){
            Object[] row = new Object[3];
            row[0] = res.getString( string: "nim");
            row[0] = res.getString( string: "nama");
            row[0] = res.getString( string: "nilai");
            dataTabel.addRow( rowData: row);
        }
    } catch (SQLException ex) {
        System.out.println( x: ex.getMessage());
    }

    return dataTabel;
}

```

Menu.java - insertData



```
public void insertData() {
    // Get Data
    String nim = fieldNim.getText();
    String nama = fieldNama.getText();
    String nilai = fieldNilai.getText();

    if (nim.isEmpty() && nama.isEmpty() && nilai.isEmpty()) {
        JOptionPane.showMessageDialog(parentComponent: null, message: "Data belum lengkap!");
    } else {
        // Add New Data
        String sql = "INSERT INTO mahasiswa VALUES ('"+nim+"', '"+nama+"', '"+nilai+"')";
        db.updateQuery(sql);
        //Update Table
        tblMhs.setModel(dataModel: setTable());
        resetForm();

        // Show Information
        System.out.println(x: "Insert Success!");
        JOptionPane.showMessageDialog(parentComponent: null, message: "Data berhasil ditambahkan!");
    }
}
```

Exercise

- Ubah proses update menjadi menggunakan Database
- Ubah proses delete menjadi menggunakan Database

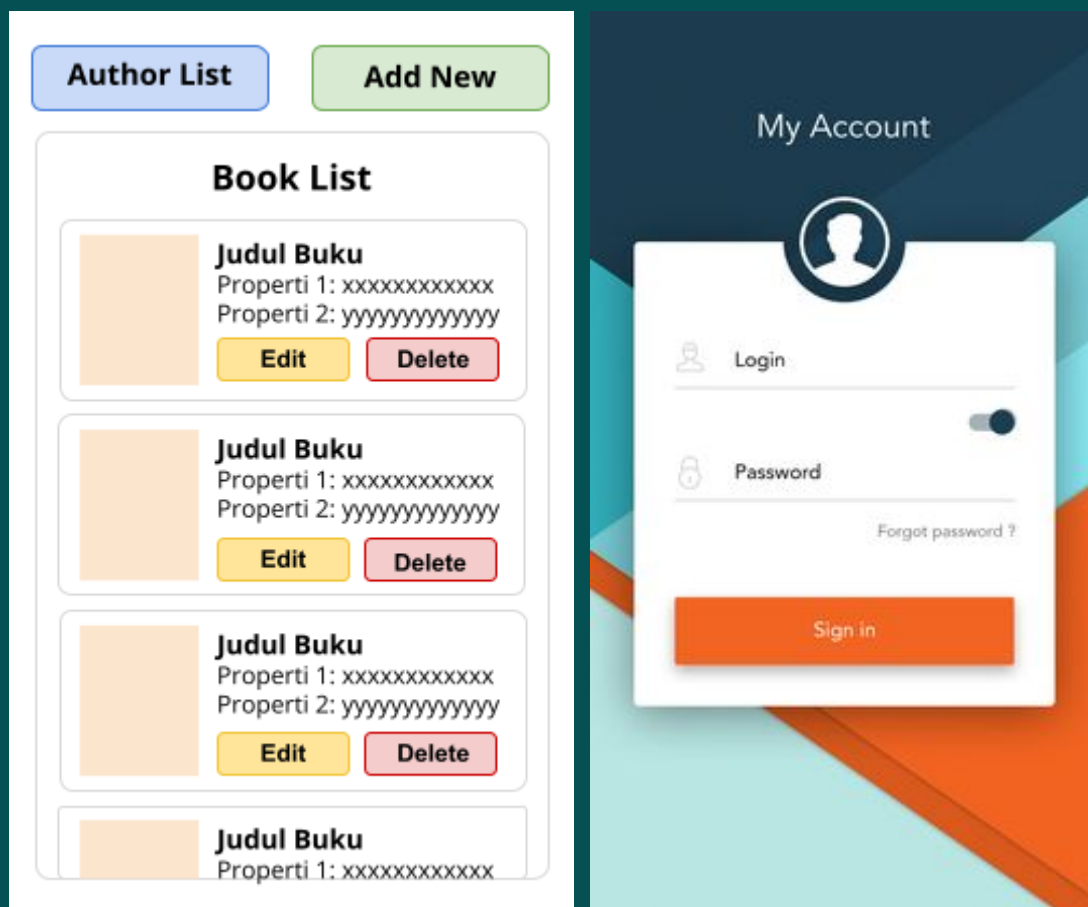
Practical Assignment

Buatlah program Java yang terkoneksi dengan database MySQL. Berikut spesifikasi program yang harus dibuat:

- Program bebas, kecuali program Mahasiswa dan Book Author
 - Terdapat proses Create, Read, Update, dan Delete data
 - Terdapat proses Autentikasi (Login, Register) *
 - Menggunakan minimal 2 tabel pada database
 - Harus terdapat minimal 1 properti gambar pada class yang dibuat (gambar akan ditampilkan pada UI)
 - Terdapat pergantian screen pada UI
 - Terdapat button navigasi untuk beralih screen
 - List data ditampilkan menggunakan card (JPanel)
- contoh card dengan JPanel [link](#)

* Boleh login dan/atau register. Tidak perlu ada enkripsi password

Contoh hasil program



Note.

- Program dikumpulkan pada repository GitHub yang dibuat public dengan nama



“TP2DPBO2023”

- Hanya program pada branch **main** yang akan dinilai dan diperiksa
- Jika waktu pengumpulan sudah habis dan ingin mengupdate kode program, update pada branch lain karena mengupdate branch Main setelah waktu pengumpulan terlewat maka program tidak akan dinilai
- Struktur folder
 - source_code
 - program
 - + Screenshot
 - + JAR
 - + SQL file
 - README.md
- File README berisi desain program, penjelasan alur, dan dokumentasi saat program dijalankan (screenshot/screen record)
- Submit link repository pada form berikut:
<https://forms.gle/rvb1hKxbQVuYNbhKA>



Penutup

Terima kasih atas kerja sama seluruh pihak yang membantu dalam penyusunan modul ini, semoga apa yang telah didapatkan bisa bermanfaat di masa yang akan datang.

Daftar Pustaka

Sukamto, Rosa A. (2018). *Pemrograman Berorientasi Objek*. Bandung: Modula.

Asisten Pemrograman 11. (2022). *Modul Desain dan Pemrograman Berorientasi Objek*.