# LAPORAN PRAKTIKUM PBO

# PEKAN KE – 2

# INTEGRASI DATABASE



# Oleh:

## Daffa Aira Adrin

## NIM 2411531006

## MATA KULIAH PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

## DOSEN PENGAMPU: NURFIAH, S.ST, M.KOM.

## FAKULTAS TEKNOLOGI INFORMASI

## DEPARTEMEN INFORMATIKA

**UNIVERSITAS ANDALAS**

**PADANG**

### Pendahuluan

1. **Latar Belakang**

Dalam pengembangan aplikasi berbasis Java, pengelolaan data yang tersimpan di dalam database merupakan aspek penting yang harus diperhatikan. Salah satu pendekatan yang umum digunakan adalah dengan memisahkan logika akses data ke dalam lapisan tersendiri, yaitu **Data Access Object (DAO)**. DAO berfungsi sebagai jembatan antara aplikasi dan database, sehingga proses penyimpanan, pengambilan, penghapusan, dan pembaruan data dapat dilakukan secara terstruktur dan terpisah dari logika bisnis.

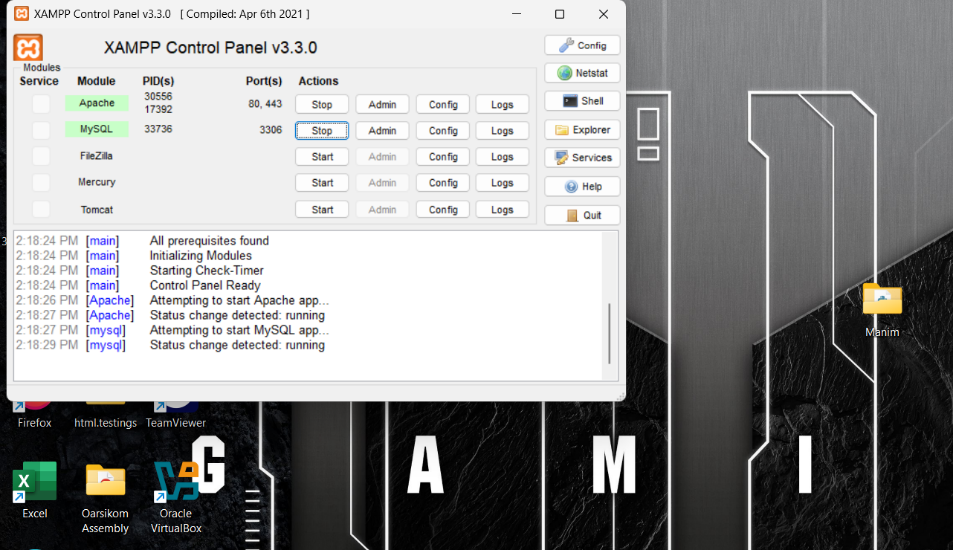
1. **Tujuan**

Tujuan dari pelaksanaan praktikum ini sebagai berikut:

1. Memahami konsep dasar **Data Access Object (DAO)** dalam pengembangan aplikasi Java berbasis database.
2. Mengimplementasikan koneksi database menggunakan **JDBC** untuk menghubungkan aplikasi dengan database MySQL.
3. Menerapkan prinsip **Object-Oriented Programming (OOP)** melalui pemisahan logika data dan tampilan menggunakan class User, interface UserDao, dan class UserRepo
4. **Landasan Teori**
5. a Object-Oriented Programming (OOP) adalah paradigma pemrograman yang berfokus pada objek sebagai representasi dari entitas dunia nyata. Dalam OOP, objek memiliki atribut (data) dan method (fungsi) yang saling terkait.
6. Data Access Object (DAO) adalah pola desain yang digunakan untuk memisahkan logika akses data dari logika bisnis aplikasi. DAO menyediakan antarmuka untuk melakukan operasi terhadap database, seperti menyimpan, mengambil, memperbarui, dan menghapus data.
7. JDBC (Java Database Connectivity) adalah API dalam bahasa pemrograman Java yang digunakan untuk menghubungkan aplikasi dengan database. JDBC menyediakan berbagai class dan interface untuk melakukan koneksi, eksekusi query, dan pengelolaan hasil query dari database relasional seperti MySQL.
8. AbstractTableModel adalah class abstrak dalam pustaka javax.swing.table yang digunakan untuk membuat model data khusus untuk komponen JTable. Dengan meng-extend AbstractTableModel,

### Langkah-Langkah Pengerjaan

1. Membuat Databse
2. Buka aplikasi XAMPP, dan aktifkan APACHE, dan MySQL



1. Unduh mysql-connector, dan atur pada eclipse IDE dengan *build path* pada JRE System Library.

A screenshot of a computer

AI-generated content may be incorrect.

1. Kemudian, pada window Java Build Path, klik Add External JARs, dan pilih mysql-connector yang telah diunduh.

A screenshot of a computer

AI-generated content may be incorrect.

1. Masuk ke database dengan mengetik <http://127.0.0.1/phpmyadmin> pada web browser.

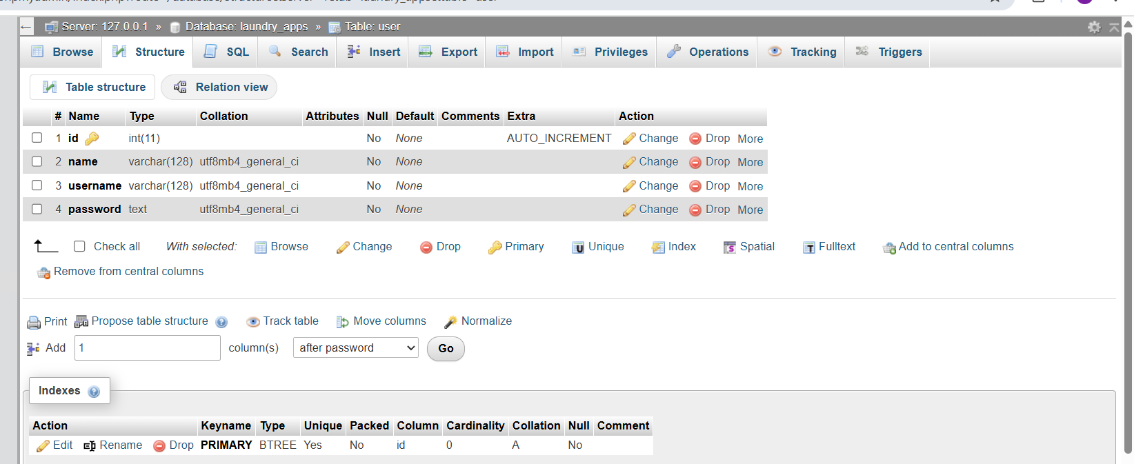
A screenshot of a computer

AI-generated content may be incorrect.

1. Buat database baru dengan nama laundryapps, dan buat tabel baru dengan nama user. Isikan nama masing-masing kolom, dan input yang lainnya sesuai dengan modul yang telah diberikan. Jadikan kolom id sebagai primary key.

A screenshot of a computer

AI-generated content may be incorrect.



1. Mengakses Database menggunakan Java
2. Buat sebuah **package** baru dengan nama confg pada project laundryapps.
3. Di dalam package confg, buat sebuah **class** baru dengan nama Database.
4. Tambahkan **import library** java.sql.\* untuk kebutuhan koneksi ke database dan javax.swing.JOptionPane untuk menampilkan pesan dialog jika terjadi error.
5. Deklarasikan atribut conn dengan tipe data Connection di dalam class.

A screenshot of a computer

AI-generated content may be incorrect.

1. Buatsebuah method static dengan nama koneksi() yang bertipe Connection.
2. Di dalam method koneksi(), tuliskan blok try-catch untuk menangani proses koneksi ke database.
3. Di dalam blok try, panggil Class.forName("com.mysql.cj.jdbc.Driver") untuk memuat driver JDBC MySQL.
4. Buat koneksi ke database dengan menggunakan DriverManager.getConnection("jdbc:mysql://localhost/laundry\_apps", "root", "").Jika koneksi berhasil, tampilkan pesan "anda berhasil login" ke console menggunakan System.out.println().
5. Kembalikan objek Connection yang telah berhasil dibuat.

A computer code on a white background

AI-generated content may be incorrect.

1. Di dalam blok catch, tangkap exception dan tampilkan pesan error menggunakan JOptionPane.showMessageDialog(null,e).

Kembalikan nilai null jika koneksi gagal.

1. Buat method main() untuk menguji koneksi ke database.
2. Di dalam method main(), panggil method Database.koneksi().

A screenshot of a computer code

AI-generated content may be incorrect.

1. DAO
2. Buat sebuah package baru dengan nama DAO.
3. Di dalam package DAO, buat sebuah interface baru dengan nama UserDao.
4. Tambahkan import untuk java.util.List karena interface ini akan menggunakan tipe data List untuk menampung kumpulan objek User.
5. Tambahkan juga import untuk class User dari package model.
6. Di dalam interface UserDao, deklarasikan method-method berikut:
7. Method save(User user) untuk menyimpan data user ke dalam database.
8. Method show() yang mengembalikan List<User> untuk menampilkan seluruh data user.
9. Method delete(String id) untuk menghapus data user berdasarkan ID.
10. Method update(User user) untuk memperbarui data user.

A screenshot of a computer

AI-generated content may be incorrect.

1. Query
2. Buat sebuah class baru dengan nama UserRepo di dalam package DAO.
3. Pastikan class UserRepo mengimplementasikan interface UserDao.
4. Tambahkan import untuk library berikut:

* java.sql.\* untuk koneksi dan manipulasi database.
* java.util.\* untuk penggunaan List dan ArrayList.
* java.util.logging.Logger untuk logging error.
* confg.Database untuk memanggil method koneksi database.
* model.User untuk menggunakan class User.
* Deklarasikan atribut connection bertipe Connection.

1. Buat beberapa query SQL sebagai final String:

* insert untuk menyimpan data user.
* select untuk menampilkan semua data user.
* delete untuk menghapus data user berdasarkan ID.
* update untuk memperbarui data user berdasarkan ID.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Buat constructor UserRepo() yang akan menginisialisasi koneksi database dengan memanggil Database.koneksi().

A computer screen shot of a code

AI-generated content may be incorrect.

1. Implementasikan method save(User user):

* Buat objek PreparedStatement untuk query insert.
* Set nilai parameter berdasarkan atribut User.
* Jalankan query dengan executeUpdate().

A screen shot of a computer code

AI-generated content may be incorrect.

1. Implementasikan method show():

* Buat objek Statement dan jalankan query select.
* Iterasi hasil ResultSet dan buat objek User untuk setiap baris data.
* Tambahkan objek User ke dalam List<User>.

A computer code with many text

AI-generated content may be incorrect.

1. Implementasikan method delete(String id):

* Buat objek PreparedStatement untuk query delete.
* Set parameter ID dan jalankan query.

A screen shot of a computer code

AI-generated content may be incorrect.

1. Implementasikan method update(User user):

* Buat objek PreparedStatement untuk query update.
* Set semua parameter berdasarkan atribut User.
* Jalankan query dengan executeUpdate().

1. Tambahkan blok try-catch di setiap method untuk menangani SQLException.
2. Tambahkan blok finally untuk menutup PreparedStatement setelah digunakan.

A screen shot of a computer code

AI-generated content may be incorrect.

1. Akses untuk table
2. Buat sebuah package baru dengan nama table.
3. Buat sebuah class baru dengan nama TableUser di dalam package tersebut.
4. Tambahkan import library javax.swing.table.AbstractTableModel untuk membuat model tabel yang dapat digunakan pada komponen JTable.
5. Tambahkan juga import untuk model.User dan java.util.List karena class ini akan menampilkan data dari objek User dalam bentuk tabel.

A screenshot of a computer

AI-generated content may be incorrect.

1. Jadikan class TableUser sebagai turunan dari AbstractTableModel.
2. Deklarasikan atribut ls bertipe List<User> untuk menampung data user yang akan ditampilkan.
3. Deklarasikan array columnNames bertipe String[] yang berisi nama-nama kolom: "ID", "Name", "Username", dan "Password".

A screenshot of a computer program

AI-generated content may be incorrect.

1. Buat constructor TableUser(List<User> ls) untuk menginisialisasi data user yang akan ditampilkan di tabel.
2. Override method getRowCount() untuk mengembalikan jumlah baris berdasarkan ukuran list ls.
3. Override method getColumnCount() untuk mengembalikan jumlah kolom, yaitu 4.
4. Override method getColumnName(int column) untuk mengembalikan nama kolom berdasarkan indeks.

A screenshot of a computer code

AI-generated content may be incorrect.

1. Override method getValueAt(int rowIndex, int columnIndex) untuk mengembalikan nilai dari objek User berdasarkan baris dan kolom yang dipilih:

* Kolom 0: ID
* Kolom 1: Nama
* Kolom 2: Username
* Kolom 3: Password

A screenshot of a computer code

AI-generated content may be incorrect.

1. Tampilan untuk pengguna
2. Buat GUI mengandung 3 Jtextfield untuk menerima input, dan 4 tombol untuk melakukan operasi yang telah ditentukan pada modul. Berikan nama variabel yang telah ditentukan di modul.

A screenshot of a computer

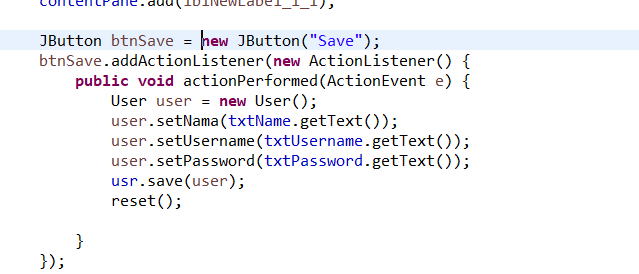
AI-generated content may be incorrect.

1. Pada tab source code, instansiasikan objek UserRepo, List<User>, dan deklarasikan atribut id dengan tipe data String.

A screenshot of a computer code

AI-generated content may be incorrect.

1. Tambahkan actionlistener pada tombol save untuk melakukan operasi terhadap input yang dimasukkan oleh pengguna ketika menekan tombol save.



### KESIMPULAN

Pada praktikum ini, dilakukan implementasi koneksi database menggunakan **JDBC (Java Database Connectivity)** yang terintegrasi dengan MySQL. Koneksi ini diatur melalui class Database, yang bertanggung jawab untuk membuka akses ke database laundry\_apps. Selanjutnya, dibuat interface UserDao yang mendefinisikan operasi dasar terhadap data pengguna, seperti save, show, delete, dan update.

Implementasi dari interface tersebut dilakukan dalam class UserRepo, yang berisi logika SQL untuk berinteraksi langsung dengan tabel user di database. Selain itu, untuk menampilkan data pengguna dalam bentuk tabel pada GUI, digunakan class TableUser yang merupakan turunan dari AbstractTableModel.