

**LAPORAN PRAKTIKUM
PEMOGRAMAN BERORIENTASI OBJEK**

PRAKTIKUM 3

**Membuat Fungsi CRUD (Create,Read,Update Delete) USER
Dengan Database MYSQL**



Disusun oleh:

Renaldi Alwean Saputra

2411532003

Dosen Pengampu :

Nurfiah, S.ST, M.Kom

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI**

UNIVERSITAS ANDALAS

SEPTEMBER

2025

A. Pendahuluan

Pemrograman Berorientasi Objek (PBO) merupakan salah satu paradigma pemrograman yang banyak digunakan dalam pengembangan aplikasi modern. Dalam praktiknya, PBO sangat mendukung pembuatan aplikasi yang modular, mudah dikelola, dan dapat digunakan kembali. Salah satu contoh penerapannya adalah dalam pengembangan aplikasi berbasis database.

CRUD (*Create, Read, Update, Delete*) merupakan operasi dasar yang wajib ada pada sebuah aplikasi untuk mengelola data. Dengan adanya fungsi CRUD, pengguna dapat menambahkan data baru, menampilkan data yang sudah ada, memperbarui data, maupun menghapus data. Pada praktikum ini, mahasiswa diminta untuk membuat fungsi CRUD data user menggunakan bahasa pemrograman Java dengan database MySQL, serta mengimplementasikan konsep PBO melalui penggunaan class, interface, dan DAO (*Data Access Object*).

B. Tujuan

Tujuan praktikum ini yaitu :

- Mahasiswa mampu membuat tabel **user** pada database MySQL.
- Mahasiswa mampu membuat koneksi antara aplikasi Java dan database MySQL.
- Mahasiswa mampu merancang tampilan GUI untuk fungsi CRUD user.
- Mahasiswa mampu membuat dan mengimplementasikan interface dalam Java.
- Mahasiswa mampu membuat fungsi **DAO** (*Data Access Object*) untuk mengakses database.
- Mahasiswa mampu mengimplementasikan operasi CRUD (*Create, Read, Update, Delete*) dengan konsep Pemrograman Berorientasi Objek.

C. Langkah kerja

1. Membuat method pada JFrame

```
public void reset() {  
    txtName.setText("");  
    txtUsername.setText("");  
    txtPassword.setText("");  
}
```

- Fungsi reset() dipakai untuk mengosongkan semua field input setelah operasi CRUD selesai, jika sudah buatlah instance pada UserFrame

```
UserRepo usr = new UserRepo();
List<User> ls;
public String id;
// =====
```

- Membuat instance usr (objek UserRepo) untuk akses database.
- ls digunakan untuk menampung list data user.
- id menyimpan **id user** yang dipilih di tabel.

Selanjutnya membuat fungsi ke 3 tombol Create, Update dan Delete dengan kodingan dan penjelasan seperti berikut :

```
// CREATE USER
btnSave.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        User user = new User();
        user.setName(txtName.getText());
        user.setUsername(txtUsername.getText());
        user.setPassword(txtPassword.getText());
        usr.save(user);
        reset();
        loadTable();
    }
});
```

- membuat objek User baru, ambil nilai dari text field, simpan ke database dengan usr.save(), lalu kosongkan input (reset()).

```
// UPDATE USER
btnUpdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        User user = new User();
        user.setName(txtName.getText());
        user.setUsername(txtUsername.getText());
        user.setPassword(txtPassword.getText());
        user.setId(Integer.parseInt(id));
        usr.update(user);
        reset();
        loadTable();
    }
});
```

- Buat objek user baru, isi dari textfield, set id, lalu update di DB. Setelah itu kosongkan form dan refresh tabel.

```
// DELETE USER
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (id != null) {
            usr.delete(Integer.parseInt(id));
            reset();
            loadTable();
        } else {
            JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di hapus");
        }
    }
});
```

- Kalau ada id (berarti user memilih baris tabel), hapus dari DB, kosongkan form, refresh tabel.
Kalau tidak ada → tampilkan pesan error.

Dan jangan lupa tambahkan kodingan pada JTable seperti berikut:

```
tableUsers.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        id = tableUsers.getValueAt(tableUsers.getSelectedRow(), 0).toString();
        txtName.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(), 1).toString());
        txtUsername.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(), 2).toString());
        txtPassword.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(), 3).toString());
    }
});
```

- Saat user klik baris tabel → ambil data per kolom → tampilkan di text field.
(id disimpan untuk update/delete)

D. Kesimpulan

Fungsi CRUD merupakan komponen penting dalam pengembangan aplikasi berbasis data, karena memungkinkan pengelolaan data secara lengkap mulai dari menambah, membaca, memperbarui, hingga menghapus data.

Koneksi aplikasi Java dengan MySQL dapat dilakukan menggunakan **MySQL Connector (JDBC)** sehingga aplikasi mampu berkomunikasi dengan database.

Penerapan **DAO** membantu memisahkan logika bisnis dengan logika akses data, sehingga aplikasi menjadi lebih modular, mudah dipelihara, dan dapat digunakan kembali.

Dengan menggabungkan GUI (Swing), DAO, dan database, mahasiswa dapat membuat aplikasi sederhana berbasis Java yang interaktif serta memiliki kemampuan mengelola data user.

