

LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

PEKAN 3: FUNGSI CRUD



Oleh :

MUHAMMAD FAZIL AFIF

NIM 2411531008

MATA KULIAH PEMOGRAMAN BERBASIS OBJEK

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

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, 2025

A. Pendahuluan

CRUD (Create, Read, Update, Delete) adalah sekumpulan operasi dasar yang digunakan untuk mengelola data dalam sebuah aplikasi. Fungsi CRUD memungkinkan aplikasi berinteraksi dengan basis data secara menyeluruh, mulai dari menambahkan data baru (Create), menampilkan atau membaca data yang ada (Read), memperbarui data yang sudah tersimpan (Update), hingga menghapus data yang tidak diperlukan (Delete). Dengan adanya fungsi CRUD, aplikasi dapat mengatur siklus hidup data secara terstruktur dan efisien, sehingga memudahkan dalam pengembangan maupun pemeliharaan sistem berbasis database.

B. Tujuan

Tujuan dari dilakukannya praktikum ini adalah

1. Mahasiswa mampu membuat fungsi CRUD dengan menggunakan konsep Pemrograman Berorientasi Objek

C. Alat dan Bahan

1. IDE (Integrated Development Environment) seperti Eclipse atau IntelliJIDEA
2. Java Development Kit (JDK)
3. Komputer atau Laptop

D. Langkah Kerja Praktikum

a. Menggunakan Fungsi CRUD DAO pada GUI

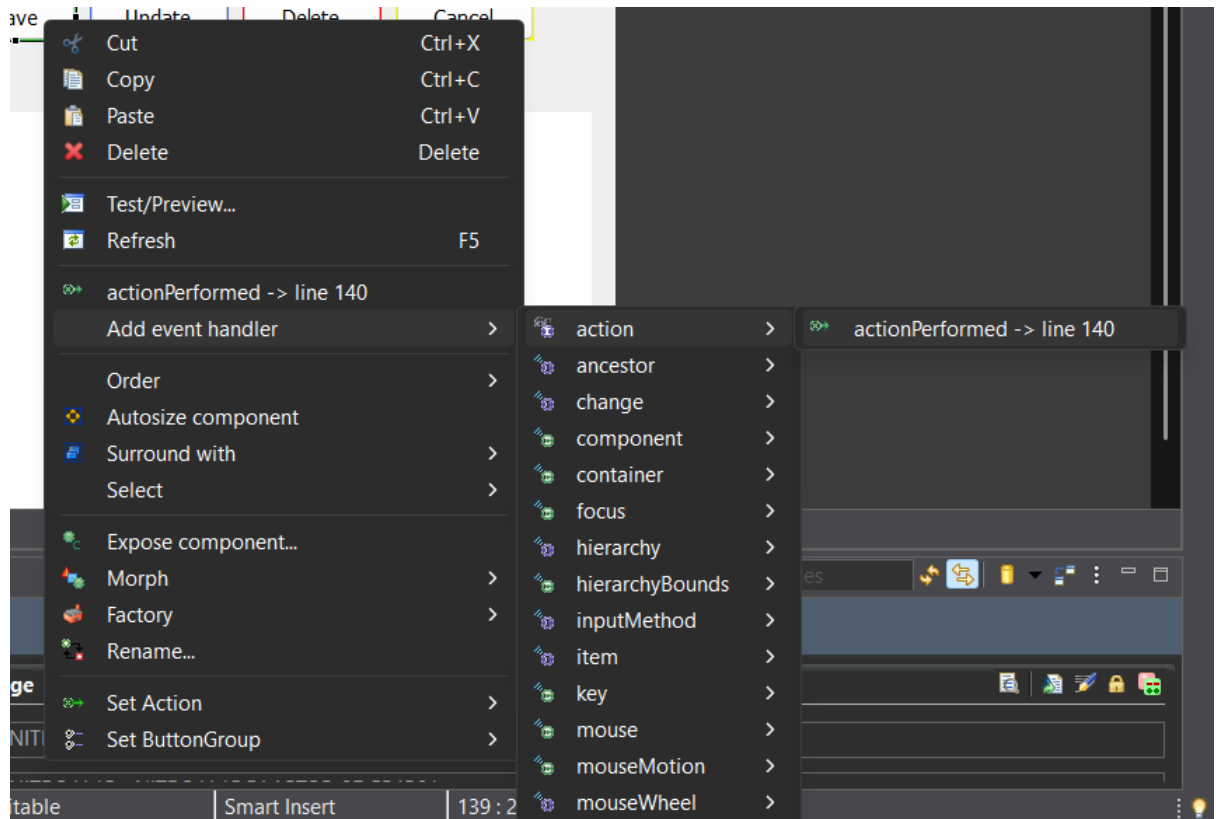
1. Buat method reset pada UserFrame untuk menghapus isi dari inputan user

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

2. Buat instance pada userframe berfungsi membuat instance dari class UserRepo yang digunakan untuk melakukan operasi database pada data user. Variabel List<User> ls; digunakan sebagai penampung kumpulan data user yang diambil dari database, sedangkan public String id; menyimpan ID user yang dipilih dari tabel sehingga bisa dipakai saat melakukan update atau delete data.

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

3. Klik kanan pada tombol save lalu add event handlers dan actionPerformed kemudian isi dengan kode program berikut. Kode ini berfungsi untuk menyimpan data user baru ke dalam database. Pertama, dibuat objek User kosong. Kemudian, nilai dari form input yaitu nama, username, dan password yang diketik pengguna dimasukkan ke dalam objek tersebut melalui setter (setName, setUsername, setPassword). Setelah objek terisi, method usr.save(user) dipanggil untuk menyimpan data ke database melalui kelas UserRepo. Terakhir, method reset() digunakan untuk mengosongkan kembali field input agar siap diisi data baru.



```

public void actionPerformed(ActionEvent e) {
    User user = new User();
    user.setNama(txtName.getText());
    user.setUsername(txtUsername.getText());
    user.setPassword(txtPassword.getText());
    usr.save(user);
    reset();
}

```

4. Buat method dengan nama loadTable(). Method ini berfungsi menampilkan ulang data user dari database ke tabel pada tampilan. Pertama, ls = usr.Show(); memanggil method Show() dari UserRepo untuk mengambil semua data user dalam bentuk list. Data tersebut kemudian dikirim ke TableUser yang berfungsi sebagai model tabel (new TableUser(ls)). Setelah model dibuat, tableUsers.setModel(tu); mengatur agar tabel menggunakan data tersebut. Terakhir, tableUsers.getTableHeader().setVisible(true); memastikan judul kolom tabel tetap terlihat.

```

public void loadTable() {
    ls = usr.Show();
    TableUser tu = new TableUser(ls);
    tableUsers.setModel(tu);
    tableUsers.getTableHeader().setVisible(true);
}

```

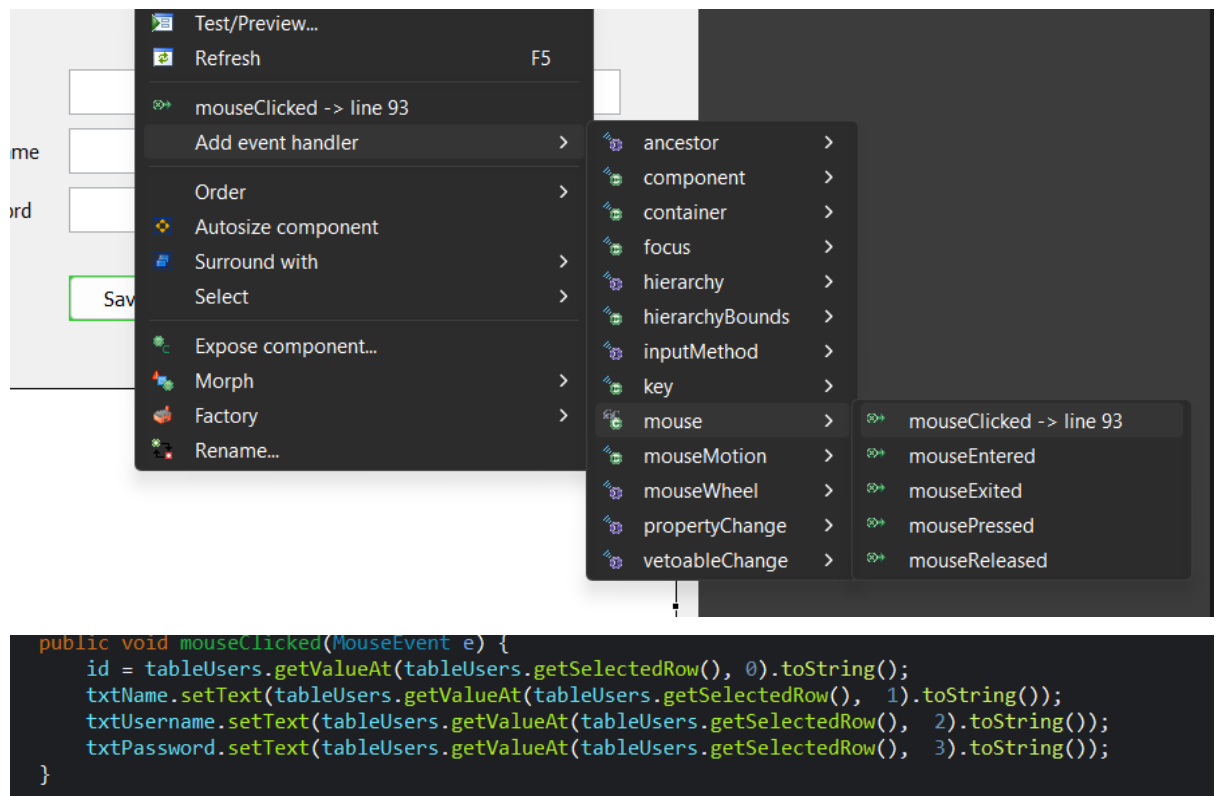
5. Memanggil method loadTable() pada class main berfungsi untuk menjalankan tampilan UserFrame sekaligus menampilkan data user di dalam tabel. Pertama, UserFrame frame = new UserFrame(); membuat objek baru dari kelas UserFrame. Lalu frame.setVisible(true); menampilkan jendela GUI agar bisa dilihat dan digunakan oleh pengguna. Terakhir, frame.loadTable(); dipanggil untuk langsung memuat data dari database ke tabel saat jendela pertama kali ditampilkan.

```

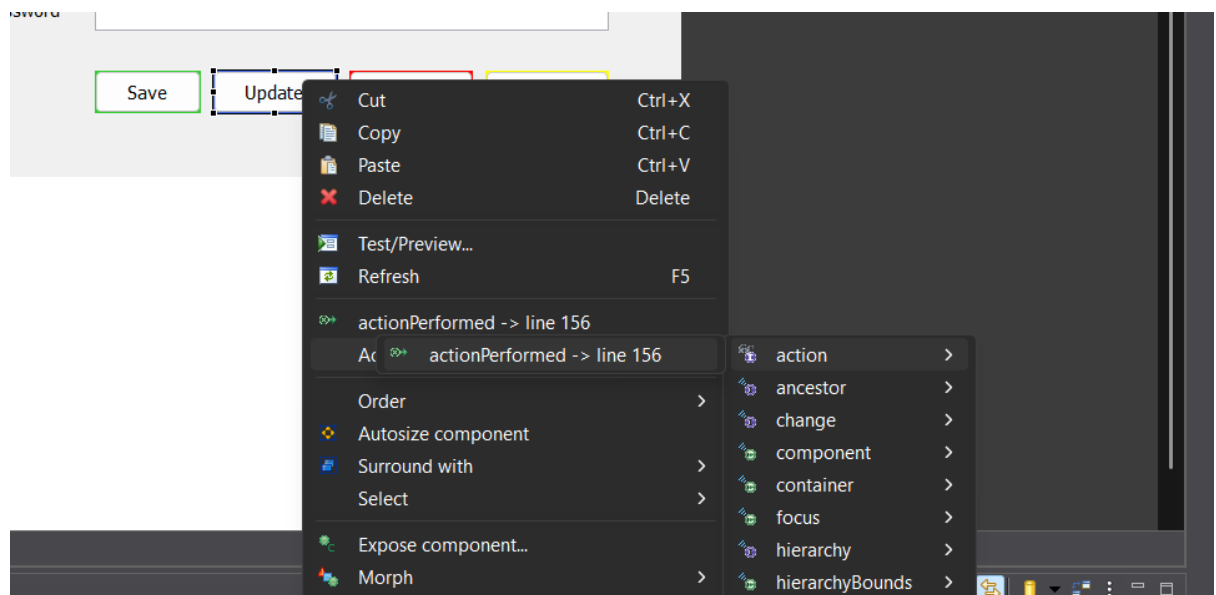
}
UserFrame frame = new UserFrame();
frame.setVisible(true);
frame.loadTable();
}

```

6. Klik kanan pada Jtable lalu add event handler pilih mouse dan mouseClicked actionPerformed kemudian isi dengan kode program berikut. Kode ini berfungsi untuk mengambil data dari baris yang dipilih di tabel dan menampilkannya ke dalam text field. Baris pertama menyimpan nilai kolom pertama (ID) dari baris yang dipilih ke variabel id. Baris berikutnya mengisi txtName, txtUsername, dan txtPassword dengan nilai dari kolom ke-1, ke-2, dan ke-3 pada baris yang sama. Dengan begitu, ketika pengguna mengklik salah satu baris di tabel, data user langsung muncul di form input sehingga bisa diedit atau dihapus.



7. Klik kanan pada tombol update lalu add event handlers dan actionPerformed kemudian isi dengan kode program berikut. Kode ini berfungsi untuk mengupdate data user yang dipilih di tabel. Pertama, dibuat objek baru `User user = new User();` lalu atributnya diisi dengan data dari text field (`txtName`, `txtUsername`, `txtPassword`) dan ID yang diambil dari tabel. Setelah lengkap, `usr.update(user);` dipanggil untuk memperbarui data di database. Kemudian `reset();` digunakan untuk mengosongkan kembali text field, dan `loadTable();` dipanggil agar tabel menampilkan data terbaru setelah proses update berhasil.

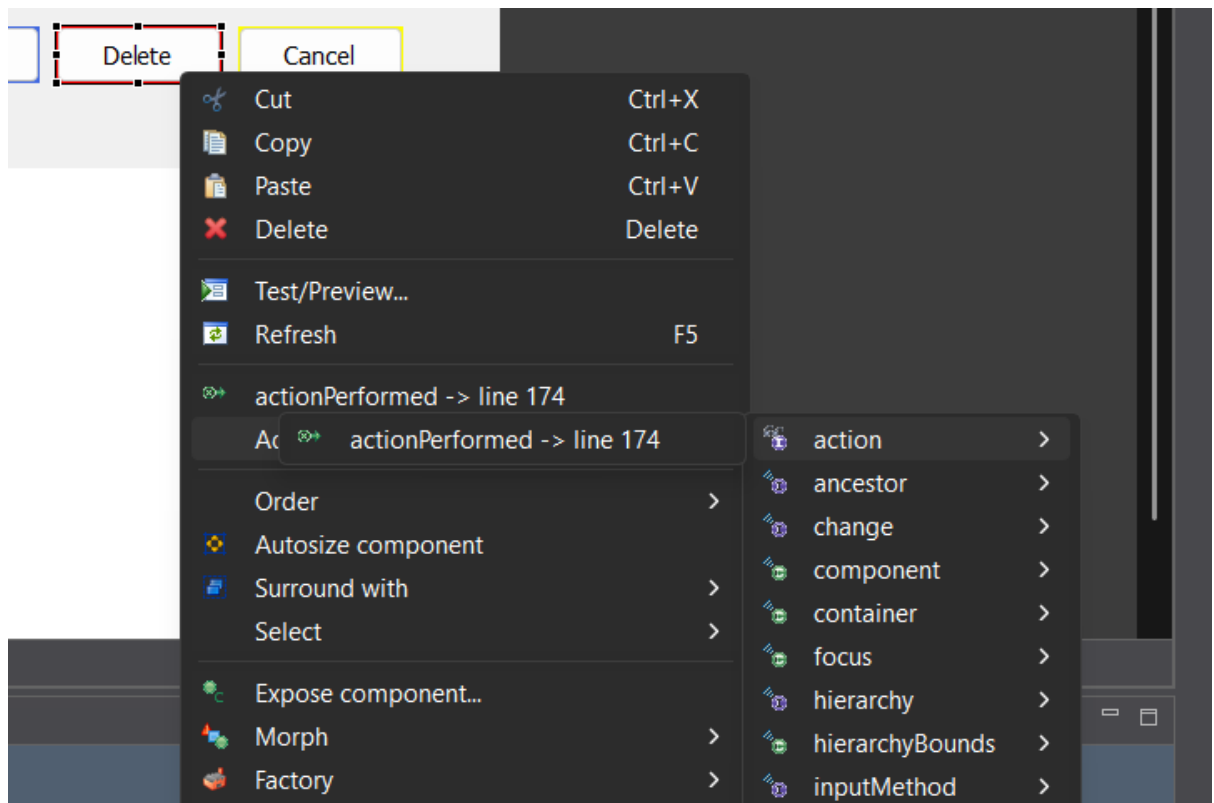


```

public void actionPerformed(ActionEvent e) {
    User user = new User();
    user.setNama(txtName.getText());
    user.setUsername(txtUsername.getText());
    user.setPassword(txtPassword.getText());
    user.setId(id);
    usr.update(user);
    reset();
    loadTable();
}

```

8. Klik kanan pada tombol delete lalu add event handlers dan actionPerformed kemudian isi dengan kode program berikut. Kode ini berfungsi menghapus data user berdasarkan ID yang dipilih dari tabel. Pertama dicek dengan `if(id != null)`, apakah pengguna sudah memilih baris di tabel (artinya ada nilai ID). Jika iya, maka `usr.delete(id)`; dijalankan untuk menghapus data di database, kemudian `reset()`; dipanggil untuk mengosongkan text field, dan `loadTable()`; agar tabel diperbarui dengan data terbaru. Jika tidak ada ID yang dipilih, maka blok else menampilkan pesan peringatan menggunakan `JOptionPane.showMessageDialog`, yang meminta pengguna untuk memilih data terlebih dahulu sebelum melakukan penghapusan.



```
public void actionPerformed(ActionEvent e) {  
    if(id != null) {  
        usr.delete(id);  
        reset();  
        loadTable();  
    }else {  
        JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan dihapus");  
    }  
}  
};
```


E. Kesimpulan

Setelah melakukan praktikum ini, kita memperoleh pemahaman mengenai penerapan fungsi CRUD (Create, Read, Update, Delete) pada aplikasi Java dengan konsep Pemrograman Berorientasi Objek (OOP). Kita mampu membuat antarmuka menggunakan JFrame untuk mengelola data user, serta menghubungkan logika program dengan database MySQL melalui DAO dan implementasinya di kelas UserRepo. Selain itu, kita juga belajar menampilkan data dengan model tabel menggunakan TableUser, serta menghubungkan setiap aksi tombol seperti save, update, dan delete agar berfungsi sesuai kebutuhan. Dengan demikian, praktikum ini memberikan pengalaman nyata dalam membangun aplikasi Java yang terintegrasi dengan database untuk pengelolaan data secara dinamis.