

LAPORAN PRAKTIKUM 3
PEMROGRAMAN BERORIENTASI OBJEK (PBO)



OLEH

Jovantri Immanuel Gulo

2411532014

DOSEN PENGAMPU

Nurfiah, S.ST, M.Kom.

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

A. Pendahuluan

Pemrograman Berorientasi Object (PBO) saat ini banyak diterapkan dalam berbagai macam penerapan aplikasi-aplikasi dalam Java, dengan adanya PBO ini kita semakin dimudahkan dengan kode program yang lebih terstruktur dan mudah untuk dibaca, ini juga memudahkan kita dalam menemukan kesalahan atau error yang terjadi, dikarenakan kode program yang tersusun dengan baik dan teratur. Dalam aplikasi laundry yang dibuat, kita menerapkan berbagai macam bentuk PBO di antaranya seperti adanya implements dari interface dan juga inheritance dan lainnya. Semua ini bertujuan untuk pembelajaran dan mempermudah serta memperluas pengetahuan kita dalam bidang pemrograman, terkhususnya Java.

B. Tujuan

Tujuan praktikum ini yaitu mahasiswa mampu membuat fungsi CRUD data user menggunakan database MySQL, Adapun poin-poin praktikum yaitu :

1. Mahasiswa mampu membuat table user pada database MySQL
2. Mahasiswa mampu membuat koneksi Java dengan database MySQL
3. Mahasiswa mampu membuat tampilan GUI CRUD user
4. Mahasiswa mampu membuat dan mengimplementasikan interface
5. Mahasiswa mampu membuat fungsi DAO (Data Access Object) dan mengimplementasikannya.
6. Mahasiswa mampu membuat fungsi CRUD dengan menggunakan konsep Pemrograman Berorientasi Objek

C. Langkah-langkah (Dari Modul Halaman 11-14)

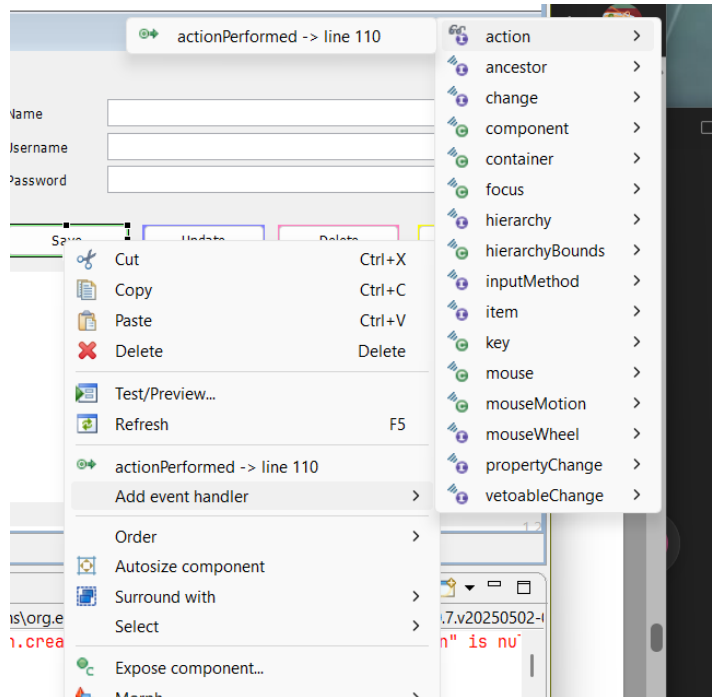
1. Menggunakan fungsi CRUD DAO pada GUI yaitu dengan menggunakan method reset.

```
64 public void reset() {  
65     txtName.setText("");  
66     txtUsername.setText("");  
67     txtPassword.setText("");  
68 }
```

2. Membuat instance pada UserFrame

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

3. Klik kanan pada button save pada GUI, klik add event handlers, dan klik actionPerformed



4. Kemudian isi dengan kode program berikut

```

108 JButton btnSave = new JButton("Save");
109 btnSave.setBackground(new Color(128, 255, 128));
110 btnSave.addActionListener(new ActionListener() {
111     public void actionPerformed(ActionEvent e) {
112         if(txtUsername.getText().contains(" ")) {
113             txtUsername.setText(txtUsername.getText().replace(" ", ""));
114         }
115
116         if(txtPassword.getText().length() < 6) {
117             JOptionPane.showMessageDialog(null, "Password harus lebih dari 6 karakter");
118             return;
119         }
120         User user = new User();
121         user.setName(txtName.getText());
122         user.setUsername(txtUsername.getText());
123         user.setPassword(txtPassword.getText());
124         usr.save(user);
125         reset();
126         loadTable();
127     }
128 });

```

5. Membuat method loadTable()

```

57 public void loadTable() {
58     ls = usr.show();
59     TableUser tu = new TableUser(ls);
60     tableUsers.setModel(tu);
61     tableUsers.getTableHeader().setVisible(true);
62 }

```

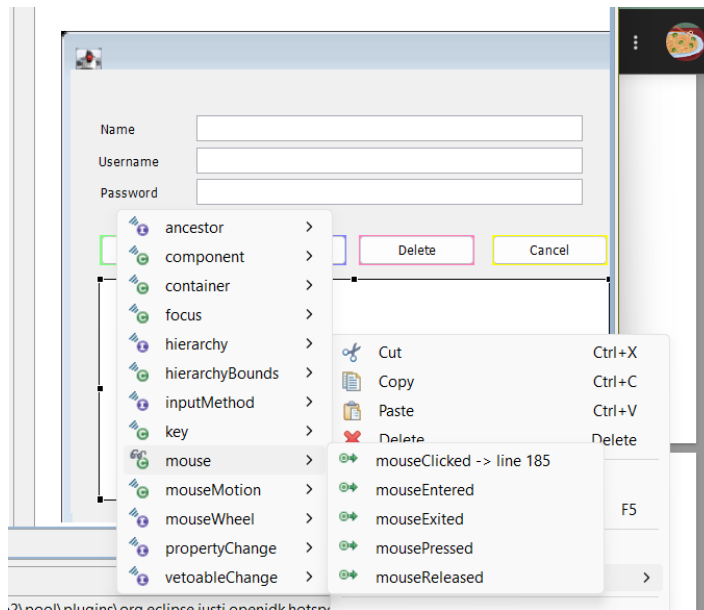
6. Memanggil method loadTable dari class frame pada class main, sehingga saat kode program pertama kali dijalankan, loadTable method akan dipanggil

```

43 public static void main(String[] args) {
44     EventQueue.invokeLater(new Runnable() {
45         public void run() {
46             try {
47                 UserFrame frame = new UserFrame();
48                 frame.setVisible(true);
49                 frame.loadTable();
50             } catch (Exception e) {
51                 e.printStackTrace();
52             }
53         }
54     });
55 }

```

7. Klik kanan pada elemen.JTable pada GUI, kemudian klik add event handler, klik mouse, dan klik mouseClicked



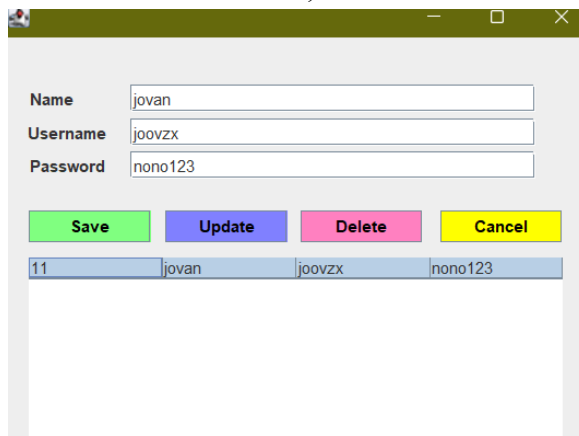
8. Kemudian diisi dengan kode program berikut

```

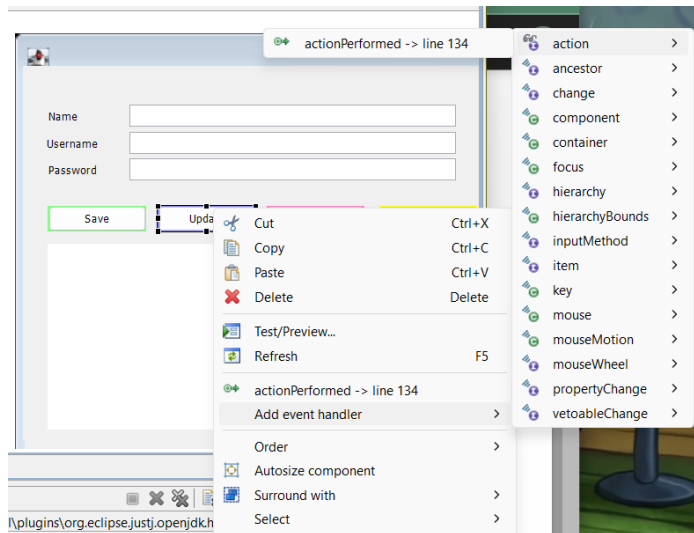
185 tableUsers.addMouseListener(new MouseAdapter() {
186     @Override
187     public void mouseClicked(MouseEvent e) {
188         id = tableUsers.getValueAt(tableUsers.getSelectedRow(), 0).toString();
189         txtName.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(), 1).toString());
190         txtUsername.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(), 2).toString());
191         txtPassword.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(), 3).toString());
192     }
193 });
194 tableUsers.setBounds(23, 161, 391, 169);
195 contentPane.add(tableUsers);
196
197 }

```

9. Klik salah satu isi table, maka akan secara otomatis tampil pada form inputan



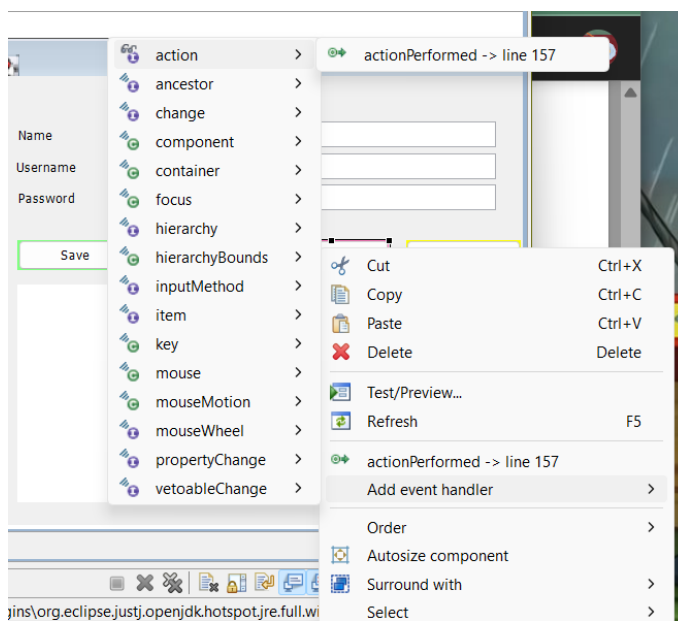
10. Klik kanan pada tombol update, klik add event handler, action, dan actionPerformed



11. Kemudian isikan dengan kode program berikut

```
134 btnUpdate.addActionListener(new ActionListener() {
135     public void actionPerformed(ActionEvent e) {
136         String checkPass = lblNewLabel_2.getText();
137         if(txtPassword.getText().length() < 6) {
138             JOptionPane.showMessageDialog(null, "Password harus lebih dari 6 karakter");
139             return;
140         }
141         User user = new User();
142         user.setName(txtName.getText());
143         user.setUsername(txtUsername.getText());
144         user.setPassword(txtPassword.getText());
145         user.setId(id);
146         usr.update(user);
147         reset();
148         loadTable();
149     }
150 }
```

12. Klik kanan pada tombol delete, klik add event handler, action, dan actionPerformed



13. Kemudian diisi dengan kode program berikut

```

156         JButton btnDelete = new JButton("Delete");
157         btnDelete.addActionListener(new ActionListener() {
158             public void actionPerformed(ActionEvent e) {
159                 if(id != null) {
160                     usr.delete(id);
161                     reset();
162                     loadTable();
163                 } else {
164                     JOptionPane.showMessageDialog(null, "Pilih data yang akan dihapus");
165                 }
166             }
167         });

```

D. Tugas

Membuat fungsi CRUD (Create, Read, Update, dan Delete) untuk **layanan (service)** dan **pelanggan (customers)**. Saya melaporkan salah satu, yaitu **CRUD untuk Layanan**.

1. Membuat ServiceDAO

```

1 package DAO;
2
3 import java.util.List;
4 import model.Service;
5
6 public interface ServiceDAO {
7     void save(Service s);
8     List<Service> show();
9     void delete(String id);
10    void update(Service s);
11 }
12

```

2. Membuat ServiceRepo

```

1 package DAO;
2
3 import config.Database;
4 import model.Service;
5
6 import java.sql.*;
7 import java.util.ArrayList;
8 import java.util.List;
9
10 public class ServiceRepo implements ServiceDAO {
11     private Connection conn;
12
13     public ServiceRepo() {
14         conn = Database.koneksi();
15     }
16
17     @Override
18     public void save(Service s) {
19         String sql = "INSERT INTO service (id, jenis, harga, status) VALUES (?, ?, ?, ?)";
20         try (PreparedStatement ps = conn.prepareStatement(sql)) {
21             ps.setString(1, s.getId());
22             ps.setString(2, s.getJenis());
23             ps.setInt(3, s.getHarga());
24             ps.setBoolean(4, s.getStatus());
25             ps.executeUpdate();
26             System.out.println("Service berhasil ditambahkan");
27         } catch (SQLException e) {
28             e.printStackTrace();
29         }
30     }

```

```

32     @Override
33     public List<Service> show() {
34         List<Service> list = new ArrayList<>();
35         String sql = "SELECT * FROM service";
36         try (Statement st = conn.createStatement(); ResultSet rs = st.executeQuery(sql)) {
37             while (rs.next()) {
38                 Service s = new Service();
39                 s.setId(rs.getString("id"));
40                 s.setJenis(rs.getString("jenis"));
41                 s.setHarga(rs.getInt("harga"));
42                 s.setStatus(rs.getBoolean("status"));
43                 list.add(s);
44             }
45         } catch (SQLException e) {
46             e.printStackTrace();
47         }
48         return list;
49     }
50
51     @Override
52     public void delete(String id) {
53         String sql = "DELETE FROM service WHERE id=?";
54         try (PreparedStatement ps = conn.prepareStatement(sql)) {
55             ps.setString(1, id);
56             ps.executeUpdate();
57             System.out.println("Service berhasil dihapus");
58         } catch (SQLException e) {
59             e.printStackTrace();
60         }
61     }
62
63     @Override
64     public void update(Service s) {
65         String sql = "UPDATE service SET jenis=?, harga=?, status=? WHERE id=?";
66         try (PreparedStatement ps = conn.prepareStatement(sql)) {
67             ps.setString(1, s.getJenis());
68             ps.setInt(2, s.getHarga());
69             ps.setBoolean(3, s.getStatus());
70             ps.setString(4, s.getId());
71             ps.executeUpdate();
72             System.out.println("Service berhasil diperbarui");
73         } catch (SQLException e) {
74             e.printStackTrace();
75         }
76     }
77 }
78

```

3. ServiceFrame yang sama seperti UserFrame.

E. Kesimpulan

Dari praktikum yang telah dikerjakan, dapat diambil kesimpulan bahwa pemrograman berorientasi objek memudahkan kita dalam mengelompokkan sesuatu yang memiliki atribut dan sifat, sehingga untuk menggunakannya, kita tinggal memanggilnya sesuai dengan nama classnya, ini sangat mempermudah kita, ditambah lagi dengan adanya sifat inheritance dan implements, semakin mempermudah kita dalam membangun aplikasi-aplikasi di lingkungan bahasa pemrograman Java.