

LAPORAN PRAKTIKUM PEKAN 3
PEMOGRAMAN BERORIENTASI OBJEK



Oleh :

KEVIN ANDHIKA

NIM 2311532005

MATA KULIAH PEMOGRAMAN BERORIENTASI OBJEK

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

FAKULTAS TEKNOLOGI INFORMASI

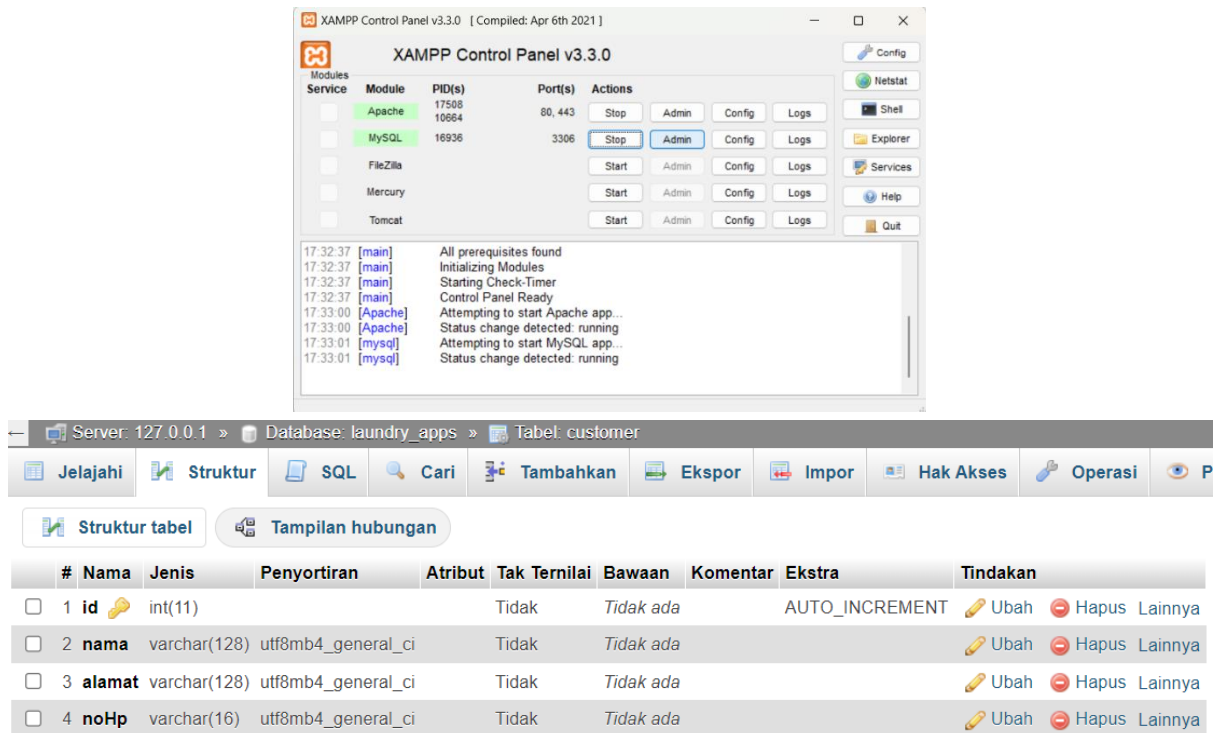
DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, SEPTEMBER 2024

PEMBAHASAN

1. Buka phpmyadmin melalui xampp, jangan lupa mencentang kolom Apache dan MySQL dan buat database untuk **Customer**



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

Service	Module	PID(s)	Port(s)	Actions
<input checked="" type="checkbox"/>	Apache	17508 10664	80, 443	Stop Admin Config Logs
<input checked="" type="checkbox"/>	MySQL	16936	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

17:32:37 [main] All prerequisites found
 17:32:37 [main] Initializing Modules
 17:32:37 [main] Starting Check-Timer
 17:32:37 [main] Control Panel Ready
 17:33:00 [Apache] Attempting to start Apache app...
 17:33:00 [Apache] Status change detected: running
 17:33:01 [mysql] Attempting to start MySQL app...
 17:33:01 [mysql] Status change detected: running

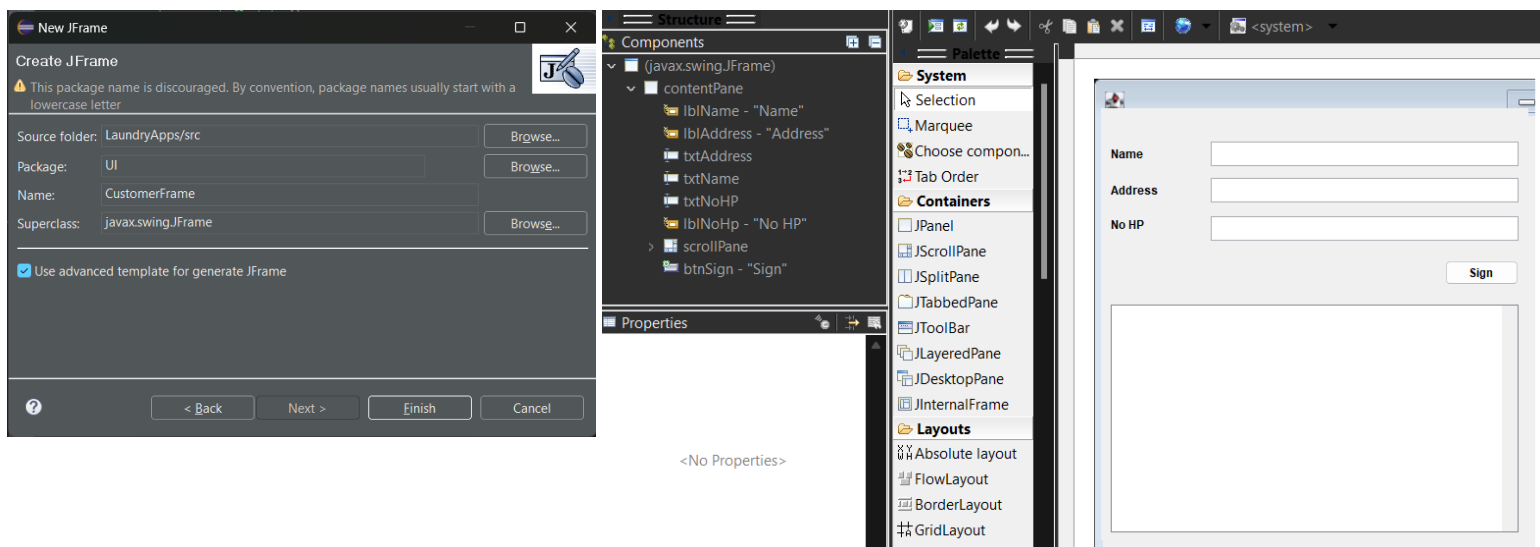
Server: 127.0.0.1 » Database: laundry_apps » Tabel: customer

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi

Struktur tabel Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/> 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/> 2	nama	varchar(128)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 3	alamat	varchar(128)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 4	noHp	varchar(16)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

2. Buka aplikasi eclipse, lalu buat file baru pada package UI dengan nama JFrame **CustomerFrame**



New JFrame

Create JFrame

This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder: LaundryApps/src Browse...

Package: UI Browse...

Name: CustomerFrame

Superclass: javax.swing.JFrame Browse...

☒ Use advanced template for generate JFrame

< Back Next > Finish Cancel

Structure

Components

- (javax.swing.JFrame)
 - contentPane
 - lblName - "Name"
 - lblAddress - "Address"
 - txtAddress
 - txtName
 - txtNoHP
 - lblNoHp - "No HP"
 - scrollPane
 - btnSign - "Sign"

Properties

<No Properties>

System

- Selection
- Marquee
- Choose compon...
- Tab Order

Containers

- JPanel
- JScrollPane
- JSplitPane
- JTabbedPane
- JToolBar
- JLayeredPane
- JDesktopPane
- JInternalFrame

Layouts

- Absolute layout
- FlowLayout
- BorderLayout
- GridLayout

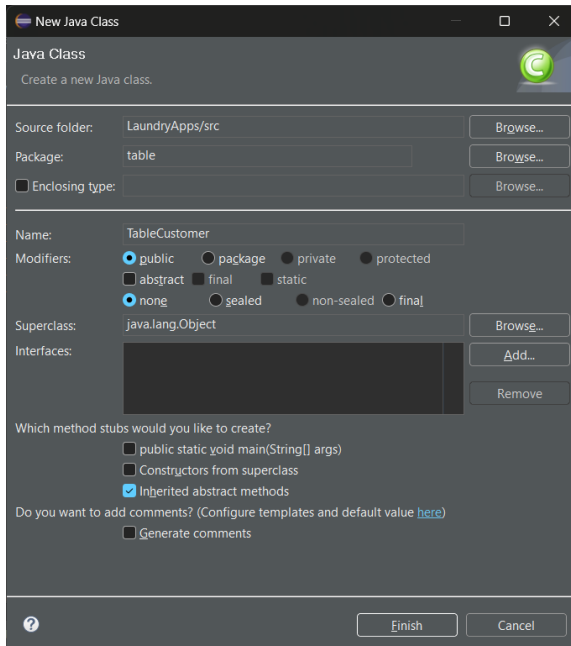
Name

Address

No HP

Sign

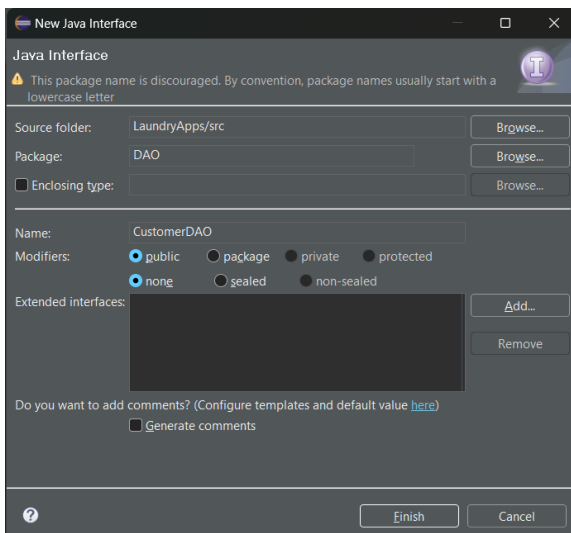
3. Kemudian buat class untuk bagian.JTable yang ada pada UI dengan nama **TableCustomer** beserta kode programnya



```
1 package table;
2
3 import java.util.List;
4
5
6
7
8
9 public class TableCustomer extends AbstractTableModel {
10     List<Customer> ls;
11     private String[] columnNames = {"ID", "Nama", "Alamat", "noHp"};
12     public TableCustomer(List<Customer> ls) {
13         this.ls = ls;
14     }
15
16     public int getRowCount() {
17         return ls.size();
18     }
19
20     public int getColumnCount() {
21         return 4;
22     }
23
24     public String getColumnName(int column) {
25         return columnNames[column];
26     }
27
28     public Object getValueAt(int rowIndex, int columnIndex) {
29         switch(columnIndex) {
30             case 0:
31                 return ls.get(rowIndex).getId();
32             case 1:
33                 return ls.get(rowIndex).getNama();
34             case 2:
```

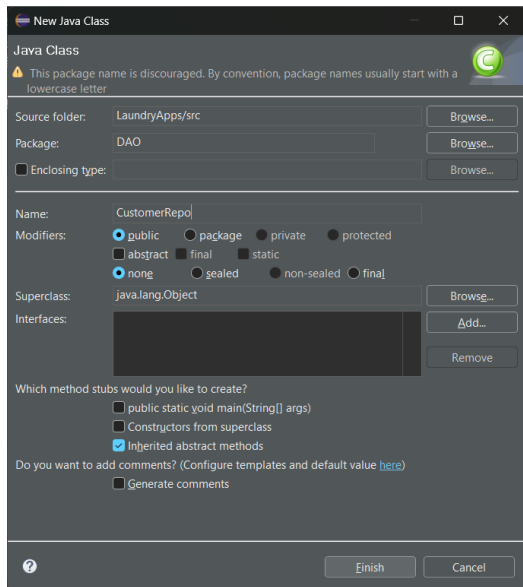
```
122     tableCustomers = new.JTable();
123     tableCustomers.setToolTipText("");
124     tableCustomers.setFillViewportHeight(true);
125     tableCustomers.setBackground(new Color(255, 255, 255));
126     tableCustomers.setBounds(10, 200, 416, 232);
127     scrollPane.setViewportView(tableCustomers);
128     tableCustomers.addMouseListener(new MouseAdapter() {
129         @Override
130         public void mouseClicked(MouseEvent e) {
131             id = tableCustomers.getValueAt(tableCustomers.getSelectedRow(), 0).toString();
132             txtName.setText(tableCustomers.getValueAt(tableCustomers.getSelectedRow(), 1).toString());
133             txtAddress.setText(tableCustomers.getValueAt(tableCustomers.getSelectedRow(), 2).toString());
134             txtNoHP.setText(tableCustomers.getValueAt(tableCustomers.getSelectedRow(), 3).toString());
135         }
136     });
137     tableCustomers.setFont(new Font("SansSerif", Font.PLAIN, 12));
```

4. Buat class DAO dengan nama **CustomerDAO** beserta kode programnya



```
1 package DAO;
2
3 import java.util.List;
4
5
6
7 public interface CustomerDAO {
8     void save(Customer cus);
9     public List<Customer> show();
10     public void delete(String id);
11     public void update(Customer cus);
12 }
13
```

5. Kemudian buat juga class **CustomerRepo** pada package DAO



```
1 package DAO;
2
3 import java.sql.Connection;
4
15
16 public class CustomerRepo implements CustomerDAO{
17     private Connection connection;
18     final String insert = "INSERT INTO customer (nama, alamat, noHp) VALUES (?, ?, ?)";
19     final String select = "SELECT * FROM customer";
20     final String delete = "DELETE FROM customer WHERE id=?";
21     final String update = "UPDATE customer SET nama=?, alamat=?, noHp=? WHERE id=?";
22
23     public CustomerRepo() {
24         connection = Database.koneksi();
25     }
26
27     @Override
28     public void save(Customer cus) {
29         PreparedStatement st = null;
30         try {
31             st = connection.prepareStatement(insert);
32             st.setString(1, cus.getNama());
33             st.setString(2, cus.getAlamat());
34             st.setString(3, cus.getNoHp());
35             st.executeUpdate();
36         } catch (SQLException e) {
37             e.printStackTrace();
38         } finally {
39             try {
40                 st.close();
```

6. Pada class UI CustomerFrame, buat instance serta method **reset** dan **loadTable** pada bagian luar main method

```
40
41     CustomerRepo cusr = new CustomerRepo();
42     List<Customer> ls;
43     String id;
44
45     public void reset() {
46         txtName.setText("");
47         txtAddress.setText("");
48         txtNoHP.setText("");
49     }
50     public void loadTable() {
51         ls = cusr.show();
52         TableCustomer tc = new TableCustomer(ls);
53         tableCustomers.setModel(tc);
54         tableCustomers.getTableHeader().setVisible(true);
55     }
56
```

7. Pada button Sign, Update dan Delete UI tambahkan program berikut

```
140     JButton btnSign = new JButton("Sign");
141     btnSign.addActionListener(new ActionListener() {
142     public void actionPerformed(ActionEvent e) {
143         Customer cus = new Customer();
144         cus.setName(txtName.getText());
145         cus.setAlamat(txtAddress.getText());
146         cus.setNoHp(txtNoHP.getText());
147         reset();
148         cusr.save(cus);
149         loadTable();
150     }
151     });
152     btnSign.setFont(new Font("SansSerif", Font.BOLD, 12));
153     btnSign.setBounds(351, 155, 75, 25);
154     contentPane.add(btnSign);
155 }
156 }
```

```
155     JButton btnUpdate = new JButton("Update");
156     btnUpdate.addActionListener(new ActionListener() {
157     public void actionPerformed(ActionEvent e) {
158         Customer cus = new Customer();
159         cus.setName(txtName.getText());
160         cus.setAlamat(txtAddress.getText());
161         cus.setNoHp(txtNoHP.getText());
162         cus.setId(id);
163         cusr.update(cus);
164         reset();
165         loadTable();
166     }
167     });
168     btnUpdate.setFont(new Font("SansSerif", Font.BOLD, 12));
169     btnUpdate.setBounds(272, 157, 75, 25);
170     contentPane.add(btnUpdate);
```

```
172     JButton btnDelete = new JButton("Delete");
173     btnDelete.addActionListener(new ActionListener() {
174     public void actionPerformed(ActionEvent e) {
175         if(id != null) {
176             cusr.delete(id);
177             reset();
178             loadTable();
179         } else {
180             JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di hapus");
181         }
182     }
183     });
184     btnDelete.setFont(new Font("SansSerif", Font.BOLD, 12));
185     btnDelete.setBounds(351, 157, 75, 25);
186     contentPane.add(btnDelete);
187 }
```

8. Jangan lupa tambahkan juga loadtable() pada method main agar tabel akan muncul saat program di running

```
57 public static void main(String[] args) {  
58     EventQueue.invokeLater(new Runnable() {  
59         public void run() {  
60             try {  
61                 CustomerFrame frame = new CustomerFrame();  
62                 frame.setVisible(true);  
63                 frame.loadTable();  
64             } catch (Exception e) {  
65                 e.printStackTrace();  
66             }  
67         }  
68     });  
69 }
```

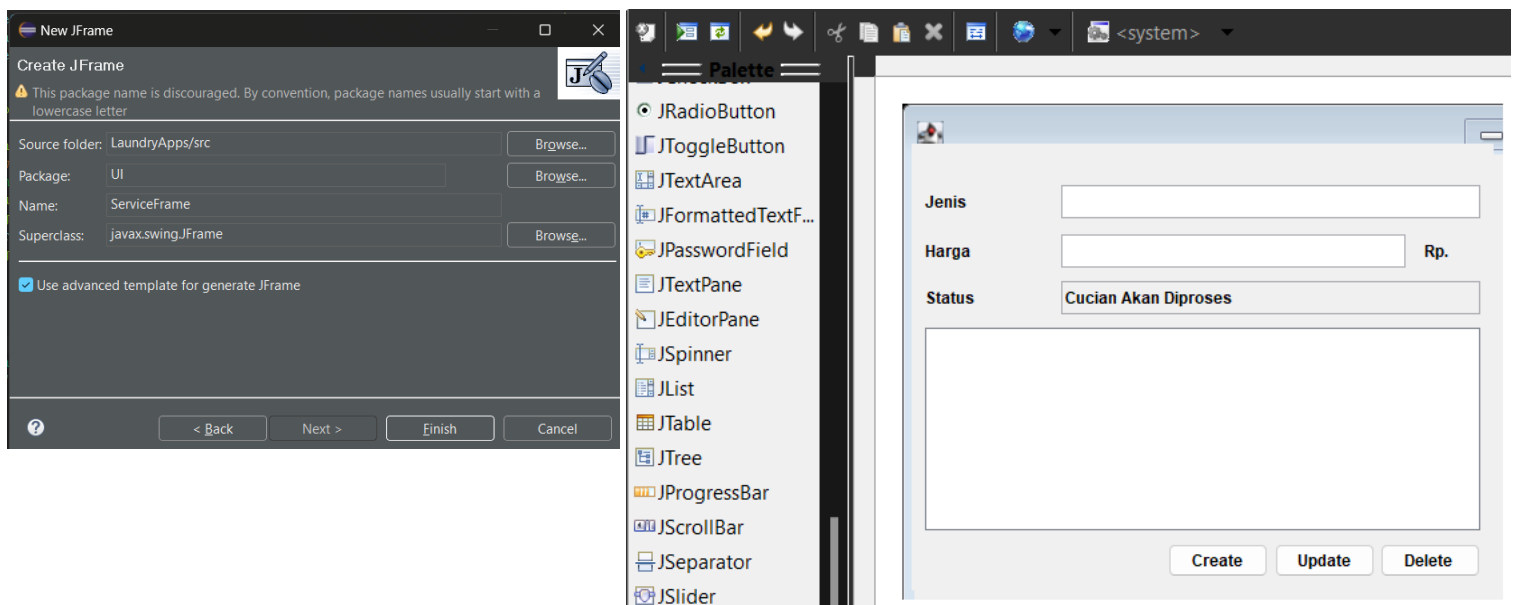
9. Buka kembali phpmyadmin dan buat database untuk **Service**



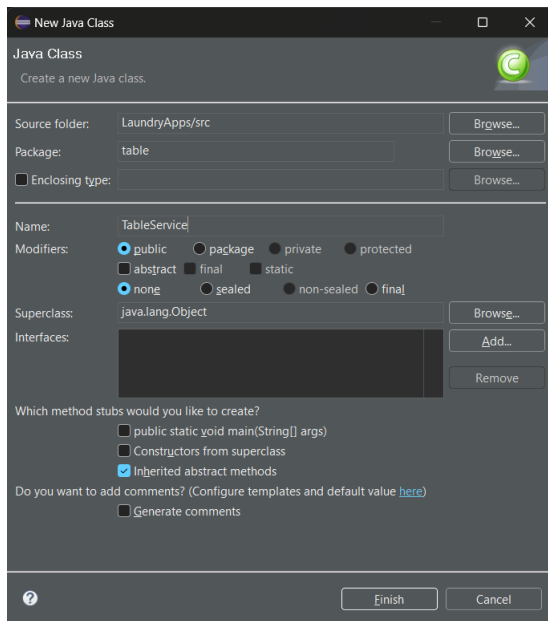
The screenshot shows the phpMyAdmin interface for a database named 'laundry_apps'. The 'service' table is selected, and its structure is displayed. The table has four columns: 'id' (int(11), primary key, auto-increment), 'jenis' (varchar(128)), 'harga' (int(64)), and 'status' (varchar(64)). The 'Tindakan' (Actions) column shows options to 'Ubah' (Edit), 'Hapus' (Delete), and 'Lainnya' (More).

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 jenis	varchar(128)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3 harga	int(64)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4 status	varchar(64)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

10. Buka aplikasi eclipse, lalu buat file baru pada package UI dengan nama JFrame **ServiceFrame**



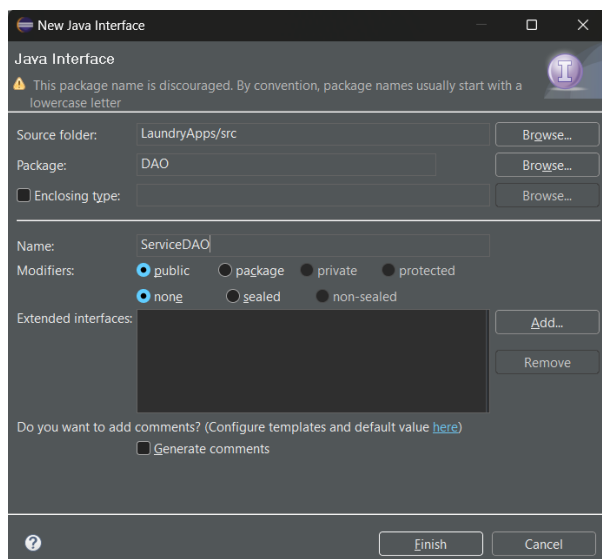
11. Kemudian buat class untuk bagian.JTable yang ada pada UI dengan nama **TableService** beserta kode programnya



```
ServiceDAO.java ServiceRepo.java TableService.java *ServiceFrame.java CustomerFr
3 import java.util.List;
8
9 public class TableService extends AbstractTableModel{
10     List<Service> ls;
11     private String[] columnNames = {"ID", "Jenis", "Harga", "Status"};
12     public TableService(List<Service> ls) {
13         this.ls = ls;
14     }
15
16     public int getRowCount() {
17         return ls.size();
18     }
19
20     public int getColumnCount() {
21         return 4;
22     }
23
24     public String getColumnName(int column) {
25         return columnNames[column];
26     }
27
28     public Object getValueAt(int rowIndex, int columnIndex) {
29         switch(columnIndex) {
30             case 0:
31                 return ls.get(rowIndex).getId();
32             case 1:
33                 return ls.get(rowIndex).getJenis();
34             case 2:
35                 return ls.get(rowIndex).getHarga();
36             case 3:
37                 return ls.get(rowIndex).getStatus();
38         }
39     }
40 }
```

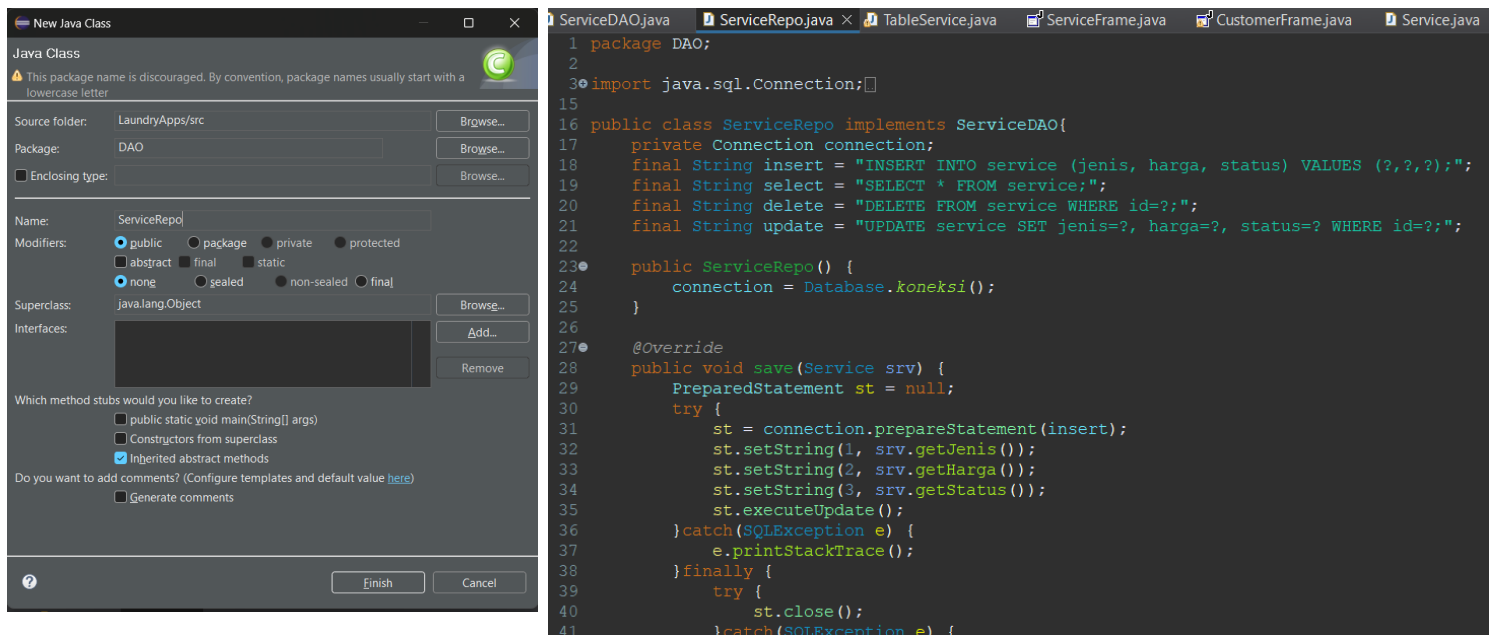
```
82 tableServices = new JTable();
83 tableServices.addMouseListener(new MouseAdapter() {
84     @Override
85     public void mouseClicked(MouseEvent e) {
86         id = tableServices.getValueAt(tableServices.getSelectedRow(), 0).toString();
87         txtJenis.setText(tableServices.getValueAt(tableServices.getSelectedRow(), 1).toString());
88         txtHarga.setText(tableServices.getValueAt(tableServices.getSelectedRow(), 2).toString());
89         txtStatus.setText(tableServices.getValueAt(tableServices.getSelectedRow(), 3).toString());
90     }
91 });
92 scrollPane.setViewportView(tableServices);
93 tableServices.setToolTipText("");
94 tableServices.setFont(new Font("SansSerif", Font.PLAIN, 12));
95 tableServices.setFillsViewportHeight(true);
96 tableServices.setBackground(Color.WHITE);
```

12. Buat class DAO dengan nama **ServiceDAO** beserta kode programnya



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


13. Kemudian buat juga class **ServiceRepo** pada package DAO



The image shows two side-by-side screenshots from an IDE. The left screenshot is the 'New Java Class' dialog. It shows the 'Package' field set to 'DAO'. The 'Name' field is 'ServiceRepo'. The 'Modifiers' section has 'public' selected. The 'Superclass' is 'java.lang.Object'. The 'Which method stubs would you like to create?' section has 'Inherited abstract methods' checked. The right screenshot shows the 'ServiceRepo.java' file. It contains the following code:

```
1 package DAO;
2
3 import java.sql.Connection;
4
15
16 public class ServiceRepo implements ServiceDAO{
17     private Connection connection;
18     final String insert = "INSERT INTO service (jenis, harga, status) VALUES (?, ?, ?);";
19     final String select = "SELECT * FROM service;";
20     final String delete = "DELETE FROM service WHERE id=?;";
21     final String update = "UPDATE service SET jenis=?, harga=?, status=? WHERE id=?;";
22
23     public ServiceRepo() {
24         connection = Database.koneksi();
25     }
26
27     @Override
28     public void save(Service srv) {
29         PreparedStatement st = null;
30         try {
31             st = connection.prepareStatement(insert);
32             st.setString(1, srv.getJenis());
33             st.setString(2, srv.getHarga());
34             st.setString(3, srv.getStatus());
35             st.executeUpdate();
36         } catch (SQLException e) {
37             e.printStackTrace();
38         } finally {
39             try {
40                 st.close();
41             } catch (SQLException e) {
```

14. Pada class UI ServiceFrame, buat instance serta method **reset** dan **loadTable** pada bagian luar main method

```
37     ServiceRepo srvr = new ServiceRepo();
38     List<Service> ls;
39     String id;
40
41     public void reset() {
42         txtJenis.setText("");
43         txtHarga.setText("");
44     }
45     public void loadTable() {
46         ls = srvr.show();
47         TableService ts = new TableService(ls);
48         tableServices.setModel(ts);
49         tableServices.getTableHeader().setVisible(true);
50     }
51
```

15. Pada button Create, Update dan Delete UI tambahkan program berikut

```

98      JButton btnCreate = new JButton("Create");
99      btnCreate.setBounds(192, 300, 75, 25);
100     btnCreate.addActionListener(new ActionListener() {
101     public void actionPerformed(ActionEvent e) {
102         Service srv = new Service();
103         srv.setJenis(txtJenis.getText());
104         srv.setHarga(txtHarga.getText());
105         srv.setStatus(txtStatus.getText());
106         reset();
107         srvr.save(srv);
108         loadTable();
109     }
110     });
111     btnCreate.setFont(new Font("SansSerif", Font.BOLD, 12));
112     contentPane.add(btnCreate);
113
114     JButton btnUpdate = new JButton("Update");
115     btnUpdate.setBounds(272, 300, 75, 25);
116     btnUpdate.addActionListener(new ActionListener() {
117     public void actionPerformed(ActionEvent e) {
118         Service srv = new Service();
119         srv.setJenis(txtJenis.getText());
120         srv.setHarga(txtHarga.getText());
121         srv.setStatus(txtStatus.getText());
122         srv.setId(id);
123         srvr.update(srv);
124         reset();
125         loadTable();
126     }
127     });
128     btnUpdate.setFont(new Font("SansSerif", Font.BOLD, 12));
129     contentPane.add(btnUpdate);
130
131     JButton btnDelete = new JButton("Delete");
132     btnDelete.setBounds(351, 300, 75, 25);
133     btnDelete.addActionListener(new ActionListener() {
134     public void actionPerformed(ActionEvent e) {
135         if(id != null) {
136             srvr.delete(id);
137             reset();
138             loadTable();
139         } else {
140             JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di hapus");
141         }
142     }
143     });
144     btnDelete.setFont(new Font("SansSerif", Font.BOLD, 12));
145     contentPane.add(btnDelete);

```

16. Jangan lupa tambahkan juga loadtable() pada method main agar tabel akan muncul saat program di running

```
52● public static void main(String[] args) {  
53●     EventQueue.invokeLater(new Runnable() {  
54●         public void run() {  
55             try {  
56                 ServiceFrame frame = new ServiceFrame();  
57                 frame.setVisible(true);  
58                 frame.loadTable();  
59             } catch (Exception e) {  
60                 e.printStackTrace();  
61             }  
62         }  
63     });  
64 }  
65
```

HASIL PRAKTIKUM

CUSTOMER

Name

Kevin Andhika

Address

Kuranji, Padang

No HP

082371507944

Sign

Update

Delete

ID	Nama	Alamat	noHp
----	------	--------	------

Name

Address

No HP

Sign

Update

Delete

ID	Nama	Alamat	noHp
8	Kevin Andhika	Kuranji, Padang	082371507944

Name

Kevin Andhika

Address

Kuranji

No HP

082371507944

Sign

Update

Delete

ID	Nama	Alamat	noHp
8	Kevin Andhika	Kuranji, Padang	082371507944

Name

Address

No HP

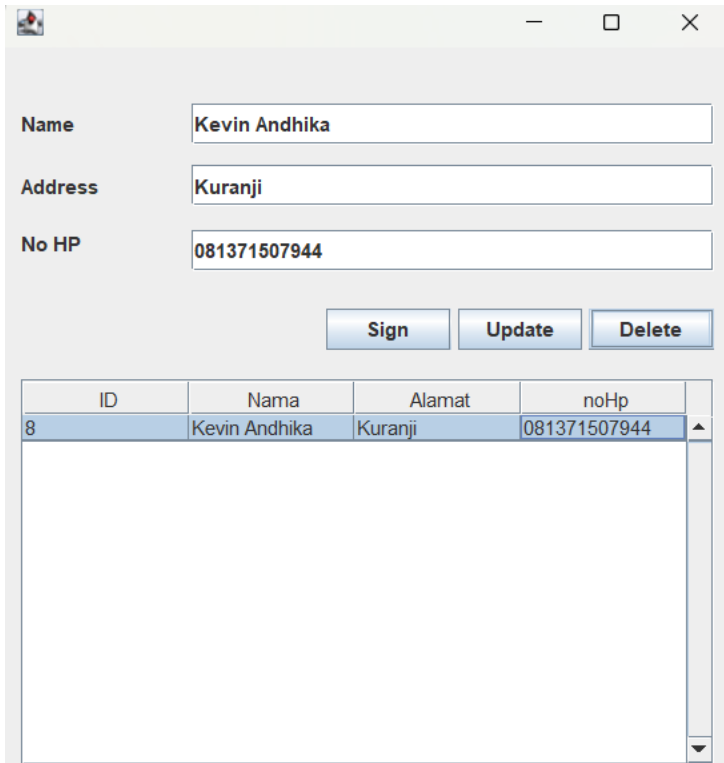
Sign

Update

Delete

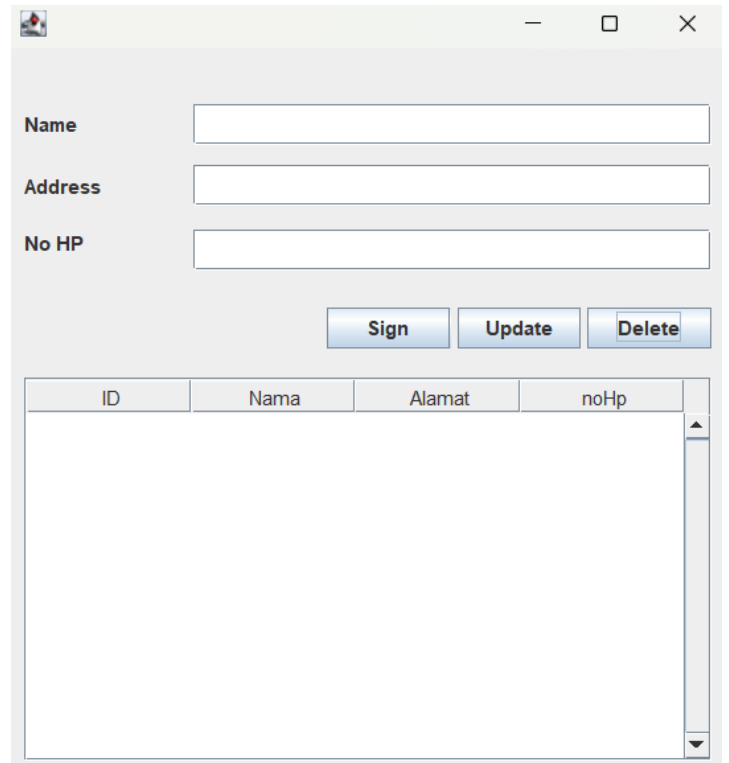
ID	Nama	Alamat	noHp
8	Kevin Andhika	Kuranji	081371507944

HASIL PRAKTIKUM



A Java Swing application window with a title bar containing a small icon and standard window controls (minimize, maximize, close). The window contains a form with three text input fields labeled "Name", "Address", and "No HP". The "Name" field contains "Kevin Andhika", the "Address" field contains "Kuranji", and the "No HP" field contains "081371507944". Below the form are three buttons: "Sign", "Update", and "Delete". At the bottom of the window is a table with four columns: "ID", "Nama", "Alamat", and "noHp". The first row of the table is highlighted in blue and contains the values "8", "Kevin Andhika", "Kuranji", and "081371507944". The table has a vertical scrollbar on the right side.

ID	Nama	Alamat	noHp
8	Kevin Andhika	Kuranji	081371507944



The same Java Swing application window as in the first screenshot, but with the form fields and the table empty. The "Name", "Address", and "No HP" fields are empty. The buttons "Sign", "Update", and "Delete" are still present. The table at the bottom has the same headers: "ID", "Nama", "Alamat", and "noHp", but it is currently empty. The table has a vertical scrollbar on the right side.

ID	Nama	Alamat	noHp
----	------	--------	------

HASIL PRAKTIKUM

SERVICE

The screenshot shows a window titled "SERVICE" with a light green header bar. It contains three input fields: "Jenis" (empty), "Harga" (empty) with a "Rp." label, and "Status" (containing "Sedang Diproses"). Below these is a table with four columns: "ID", "Jenis", "Harga", and "Status". The table is currently empty. At the bottom are three buttons: "Create", "Update", and "Delete".

ID	Jenis	Harga	Status
----	-------	-------	--------

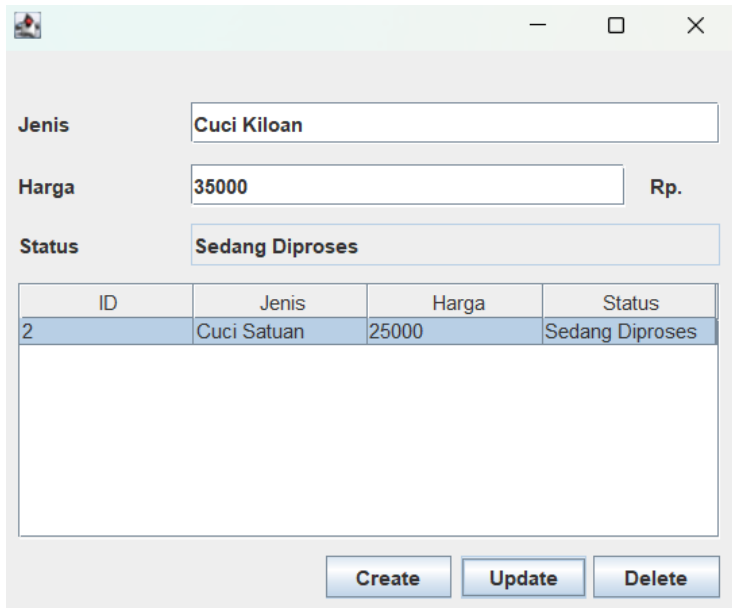
The screenshot shows the same window as before, but with "Cuci Satuan" entered in the "Jenis" field and "25000" entered in the "Harga" field. The "Status" field still contains "Sedang Diproses". The table remains empty.

ID	Jenis	Harga	Status
----	-------	-------	--------

The screenshot shows the same window, but now the data from the previous screenshot is entered into the table. The "Jenis" field is empty again. The table now contains one row with the data: ID 2, Jenis Cuci Satuan, Harga 25000, and Status Sedang Diproses.

ID	Jenis	Harga	Status
2	Cuci Satuan	25000	Sedang Diproses

HASIL PRAKTIKUM



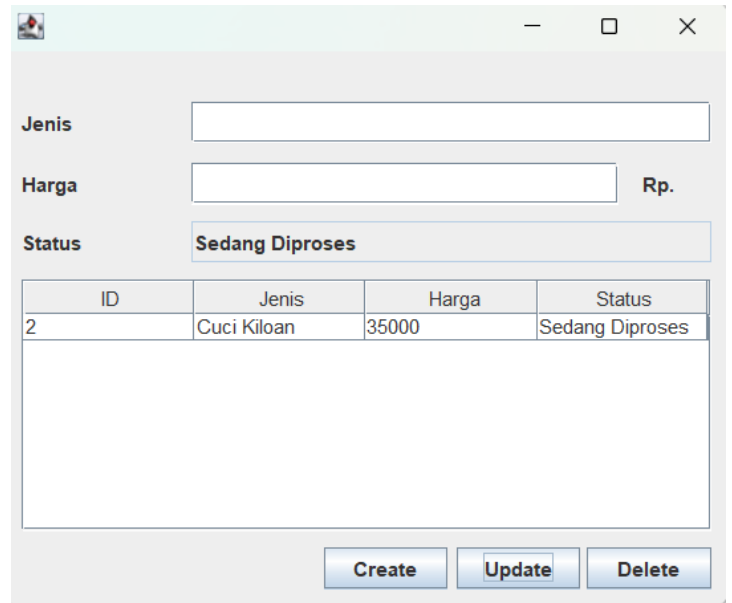
Jenis: Cuci Kiloan

Harga: 35000 Rp.

Status: Sedang Diproses

ID	Jenis	Harga	Status
2	Cuci Satuan	25000	Sedang Diproses

Create Update Delete



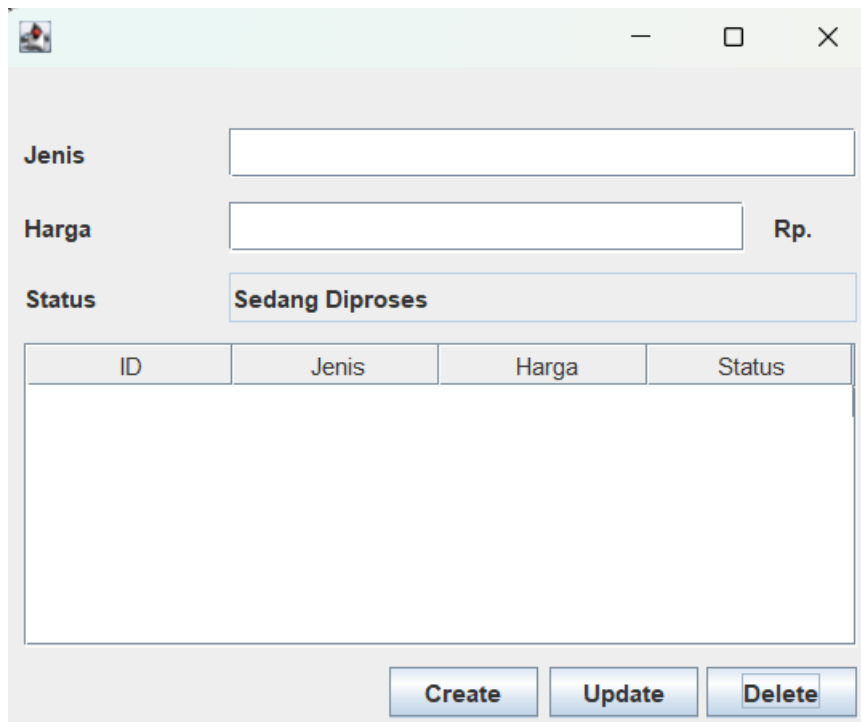
Jenis:

Harga: Rp.

Status: Sedang Diproses

ID	Jenis	Harga	Status
2	Cuci Kiloan	35000	Sedang Diproses

Create Update Delete



Jenis:

Harga: Rp.

Status: Sedang Diproses

ID	Jenis	Harga	Status
----	-------	-------	--------

Create Update Delete