

LAPORAN PRAKTIKUM PEKAN 4
PEMOGRAMAN BERORIENTASI OBJEK



Oleh :

KEVIN ANDHIKA

NIM 2311532005

MATA KULIAH PEMOGRAMAN BERORIENTASI OBJEK

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

FAKULTAS TEKNOLOGI INFORMASI

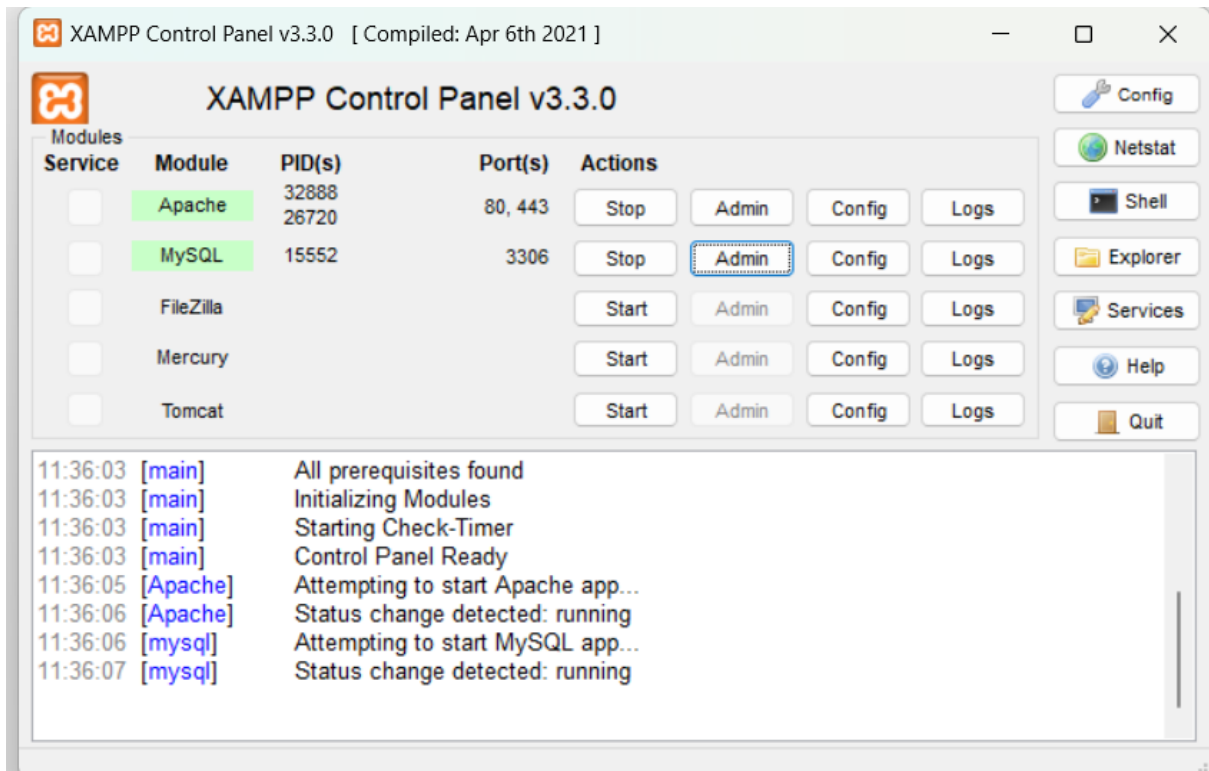
DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, OKTOBER 2024

PEMBAHASAN

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



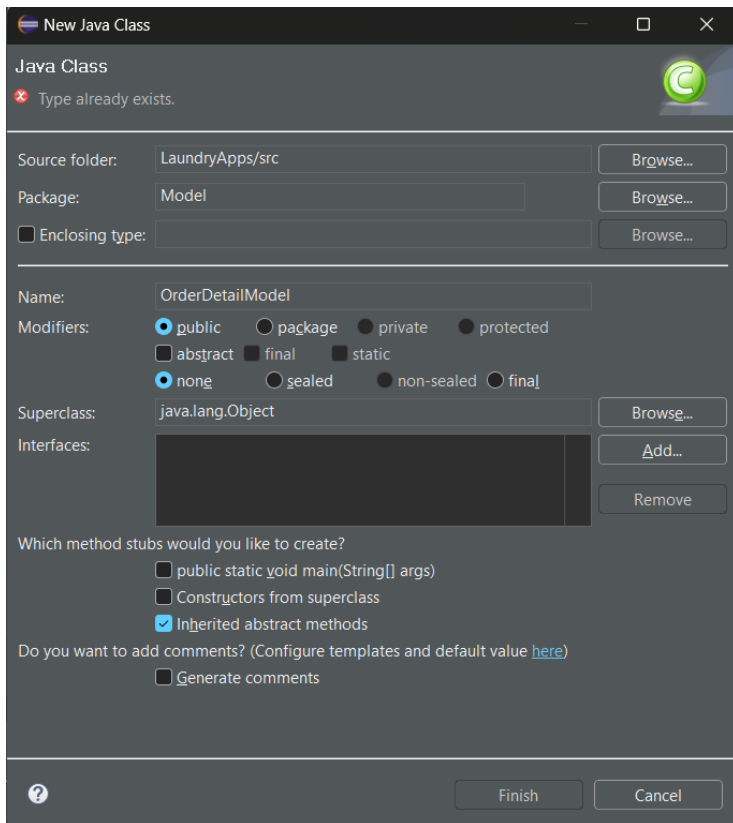
Tabel Order

Server: 127.0.0.1 » Database: laundry_apps » Tabel: order									
Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Ope									
Struktur tabel Tampilan hubungan									
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/> 1	id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 2	tanggal	date			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 3	qty	int(10)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 4	total	int(20)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 5	status	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Tabel Order_Detail

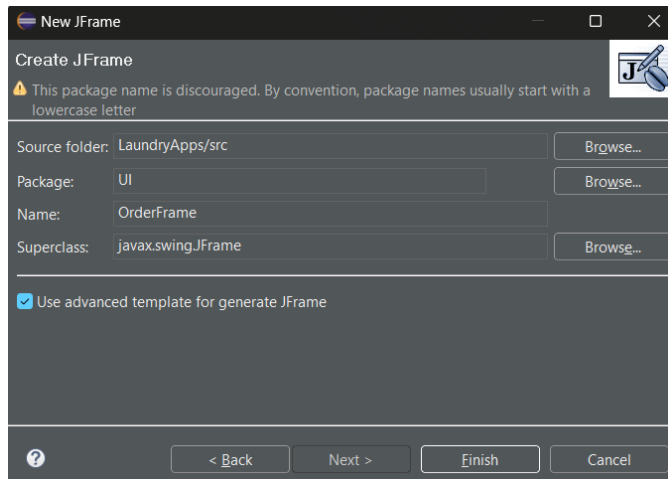
Server: 127.0.0.1 » Database: laundry_apps » Tabel: order_detail									
Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Ope									
Struktur tabel Tampilan hubungan									
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/> 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/> 2	qty	int(10)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 3	total	int(20)			Tidak	Tidak ada			Ubah Hapus Lainnya

2. Buka aplikasi eclipse, buat Class **OrderDetailModel** sebagai model dari Database OrderDetail



```
1 package Model;
2
3 public class OrderDetailModel {
4     String id, qty, total;
5
6     public String getId() {
7         return id;
8     }
9
10    public void setId(String id) {
11        this.id = id;
12    }
13
14    public String getQty() {
15        return qty;
16    }
17
18    public void setQty(String qty) {
19        this.qty = qty;
20    }
21
22    public String getTotal() {
23        return total;
24    }
25
26    public void setTotal(String total) {
27        this.total = total;
28    }
29 }
30
```

3. lalu buat file baru pada package UI dengan nama **OrderFrame**

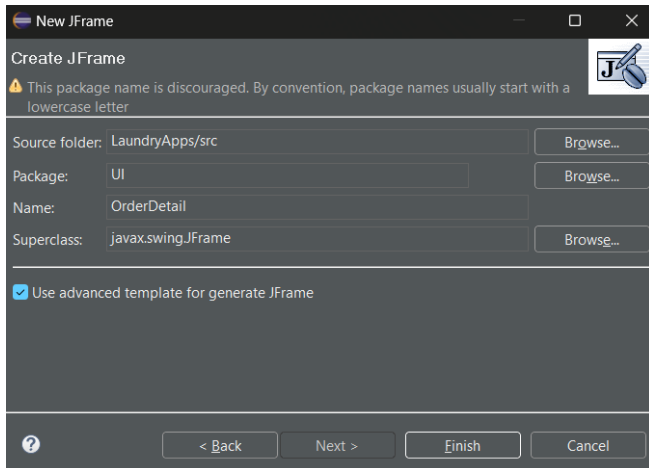


Tambahkan link penghubung saat menekan tombol **Order** ke UI OrderDetail

```
 JButton btnOrder = new JButton("Order");
 btnOrder.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) {
         OrderDetail odf = new OrderDetail();
         odf.setVisible(true);
         odf.loadTable1();
         odf.loadTable();
         dispose();
     }
 })
```

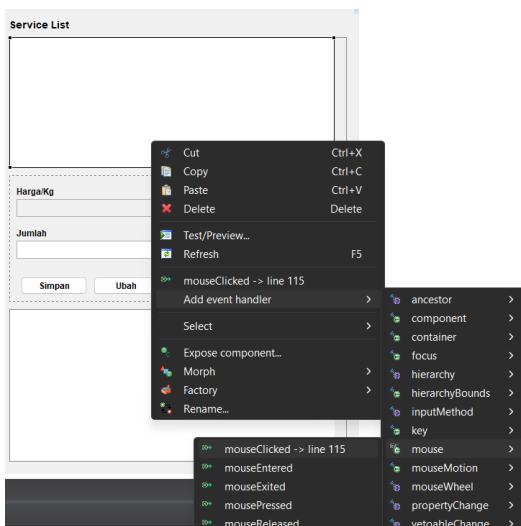
4. Kemudian buat juga UI **OrderDetail**

Setelah selesai membuat desainnya, buat kembali **tabel services** didalam UI OrderDetail dengan cara menambahkan i niasi ServiceRepo, loadtable,



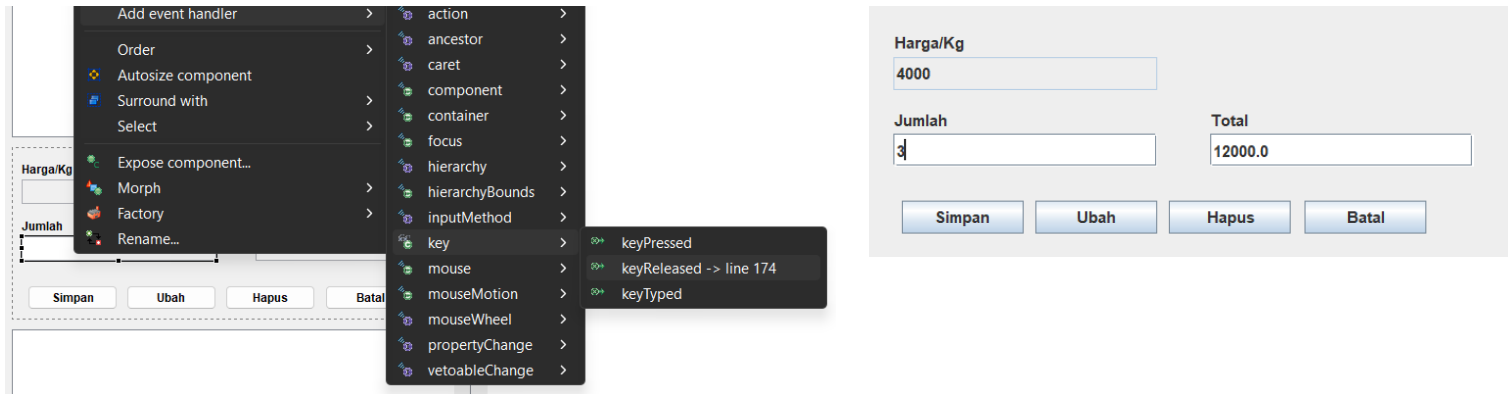
ID	Jenis	Harga	Status
1	Cuci	4000	Tersedia
2	Setrika	3000	Tersedia
3	Cuci Kilat	10000	Tersedia

5. Klik kanan table services tadi lalu tambahkan event **MouseClicked** agar ketika menekan suatu row pada tabel services, harga yang ada pada row tersebut akan di ambil pada jtextfield harga di bawahnya

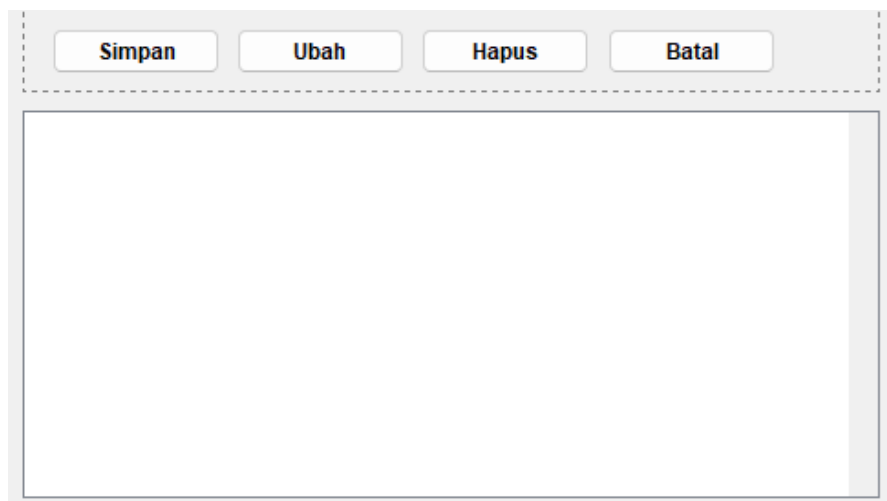


```
@Override
public void mouseClicked(MouseEvent e) {
    txtHargaKg.setText(tableServices.getValueAt(tableServices.getSelectedRow(), 2).toString());
    txtJumlah.setText("");
    txtTotal2.setText("");
}
);
```

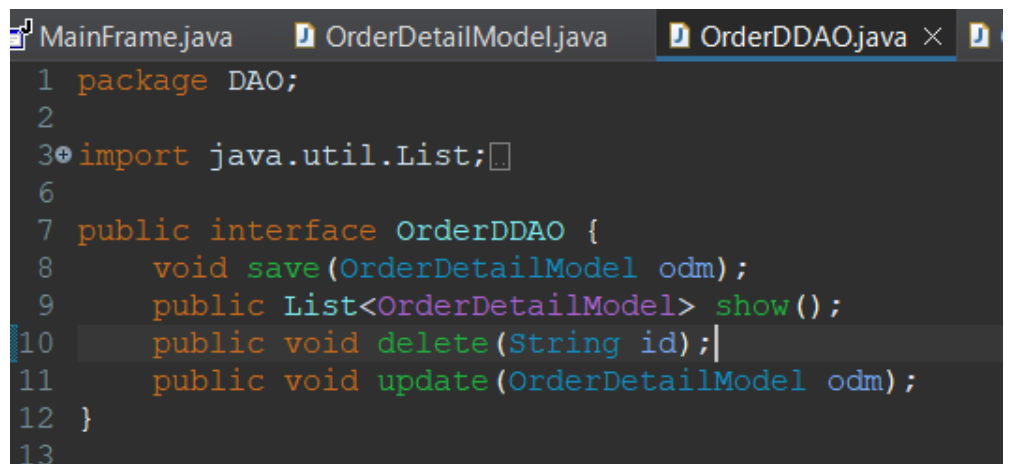
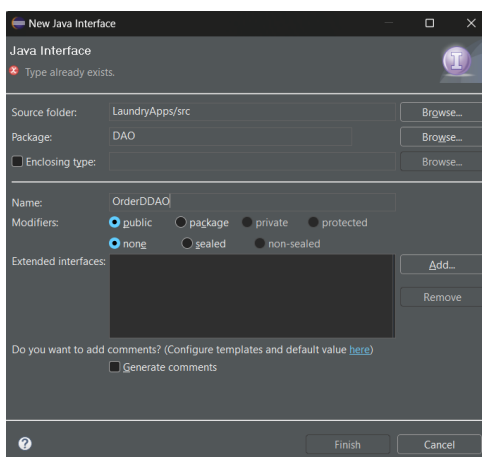
- Klik kanan jtextfield Jumlah dibawahnya lalu tambahkan event **KeyRelease** agar ketika memasukkan suatu nilai integer akan memberikan output hasil perkalian jtextfield harga dan jtextfield jumlah pada jtextfield total



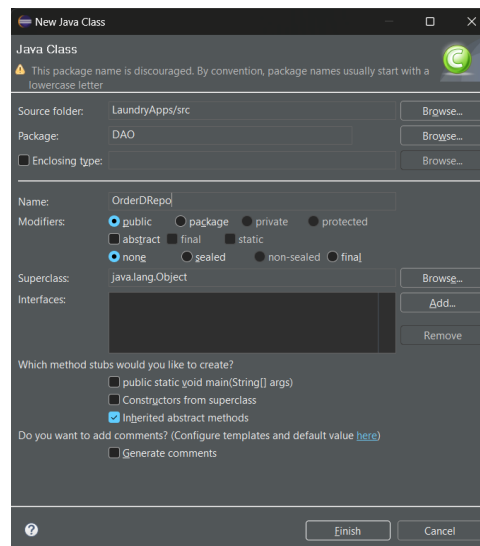
- Buat **CRUD** untuk UI OrderDetail agar bisa menggunakan tabel OrderDetails yang ada di bawah dengan cara



Membuat Class Interface OrderDDAO sebagai tempat menginisialisasikan perintah save, show, list, dan delete pada UI OrderDetail

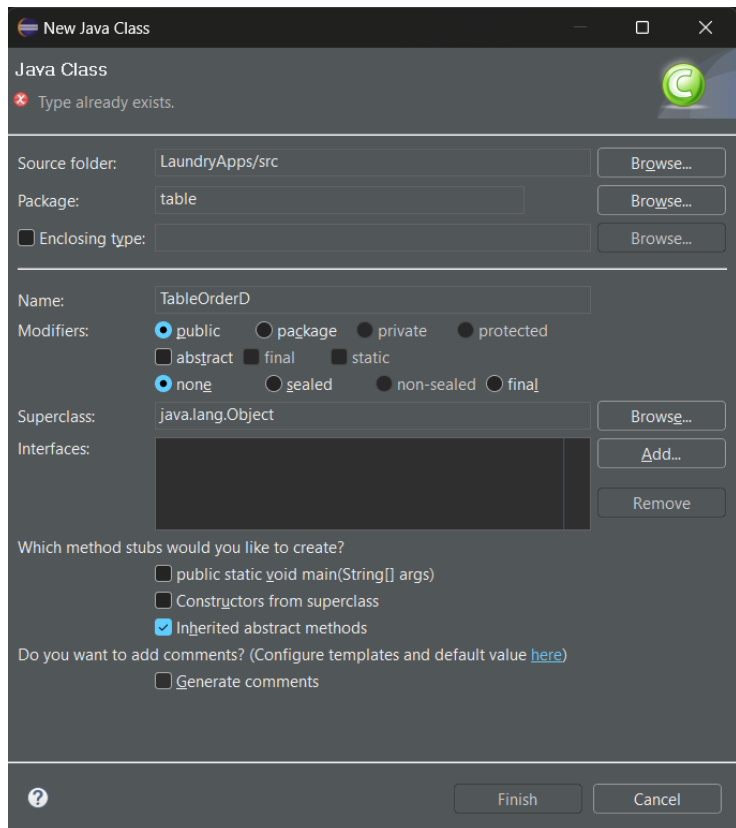


Membuat Class OrderDRepo sebagai tempat menyimpan data yang diinput dari UI ke dalam Database Order_Detail, Dan juga seluruh kodingan agar perintah yang diinisiasika pada DAO sebel unya bekerja



```
MainFrame.java  OrderDetailModel.java  OrderDDAO.java  OrderDRepo.java  TableOrderD.java  Or
1 package DAO;
2
3 import java.sql.Connection;
4
15
16 public class OrderDRepo implements OrderDDAO{
17     private Connection connection;
18     final String insert = "INSERT INTO order_detail (qty, total) VALUES (?,?)";
19     final String select = "SELECT * FROM order_detail;";
20     final String delete = "DELETE FROM order_detail WHERE id=?";
21     final String update = "UPDATE order_detail SET qty=?, total=? WHERE id=?";
22
23     public OrderDRepo() {
24         connection = Database.koneksi();
25     }
26
27     @Override
28     public void save(OrderDetailModel ordd) {
29         PreparedStatement st = null;
30         try {
31             st = connection.prepareStatement(insert);
32             st.setString(1, ordd.getQty());
33             st.setString(2, ordd.getTotal());
34             st.executeUpdate();
35         } catch (SQLException e) {
36             e.printStackTrace();
37         } finally {
38             try {
39                 st.close();
40             } catch (SQLException e) {
41                 e.printStackTrace();
42             }
43         }
44     }
45     @Override
46     public List<OrderDetailModel> show() {
47         List<OrderDetailModel> ls2 = null;
48         try {
49             ls2 = new ArrayList<OrderDetailModel>();
```

Membuat Class TableOrderD sebagai Jtable yang akan dipanggil pada UI nantinya



```

1 package table;
2
3 import java.util.List;
4
5 public class TableOrderD extends AbstractTableModel {
6     List<OrderDetailModel> ls;
7
8     private String[] columnNames = {"ID", "Quantity", "Total"};
9
10    public TableOrderD(List<OrderDetailModel> ls) {
11        this.ls = ls;
12    }
13
14    public int getRowCount() {
15        return ls.size();
16    }
17
18    public int getColumnCount() {
19        return 3;
20    }
21
22    public String getColumnName(int column) {
23        return columnNames[column];
24    }
25
26    public Object getValueAt(int rowIndex, int columnIndex) {
27        switch(columnIndex) {
28            case 0:
29                return ls.get(rowIndex).getId();
30            case 1:
31                return ls.get(rowIndex).getQty();
32            case 2:
33                return ls.get(rowIndex).getTotal();
34            default:
35                return null;
36        }
37    }
38 }

```

Setelah semua langkah diatas selesai, Berikan perintah pada tombol Simpan, Ubah, dan Hapus yang ada pada UI OrderDetail

```

JButton btnSimpan2 = new JButton("Simpan");
btnSimpan2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        OrderDetailModel ordd = new OrderDetailModel();
        ordd.setQty(txtJumlah.getText());
        ordd.setTotal(txtTotal2.getText());
        reset();
        odr.save(ordd);
        loadTable();
    }
});

```

```

JButton btnUbah2 = new JButton("Ubah");
btnUbah2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        OrderDetailModel ordd = new OrderDetailModel();
        ordd.setQty(txtJumlah.getText());
        ordd.setTotal(txtTotal2.getText());
        ordd.setId(id);
        odr.update(ordd);
        reset();
        loadTable();
    }
});

```

```

JButton btnHapus2 = new JButton("Hapus");
btnHapus2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(id != null) {
            odr.delete(id);
            reset();
            loadTable();
        } else {
            JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di hapus");
        }
    }
});

```

8. Membuat suatu perintah agar menampilkan seluruh total OrderDetail pada total Order

The screenshot shows a Java Swing window with a light gray background. On the left side, there are three text input fields: 'Total' containing 'Rp. 77000', 'Pembayaran' containing 'Cash', and 'Status Pembayaran' containing 'Lunas'. Each field has a small downward arrow on its right side. Below these fields are two buttons: 'Simpan' and 'Batal'. On the right side, there is a table with four columns: 'ID', 'Jenis', 'Quantity', and 'Total'. The table contains three rows of data. Above the table, there are four buttons: 'Simpan', 'Ubah', 'Hapus', and 'Batal'. The table has a vertical scrollbar on its right side.

ID	Jenis	Quantity	Total
1	Cuci	8	32000
2	Setrika	5	15000
3	Cuci Kilat	3	30000

9. Menambahkan tanggal saat pemesanan

```
LocalDate today = LocalDate.now();
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd/MM/yyyy");
String formattedDate = today.format(formatter);
```

```
txtTanggal = new JTextField();
txtTanggal.setText(formattedDate);
txtTanggal.setEditable(false);
txtTanggal.setFont(new Font("SansSerif", Font.BOLD, 12));
txtTanggal.setColumns(10);
txtTanggal.setBounds(10, 193, 221, 25);
panel_1.add(txtTanggal);
```

The screenshot shows a Java Swing window with a light gray background. It contains two text input fields. The first field is labeled 'Tanggal' and contains the date '27/10/2024'. The second field is labeled 'Tanggal Pengambilan' and is currently empty.

HASIL PRAKTIKUM

UI OrderFrame

Back

DATA ORDERAN

Order Delete Edit/Detail

UI OrderDetail

Service List

ID	Jenis	Harga	Status
1	Cuci	4000	Tersedia
2	Setrika	3000	Tersedia
3	Cuci Kilat	10000	Tersedia

Order ID: TRX-00001

Pelanggan: Pilih

Tanggal:

Tanggal Pengambilan:

Status: Proses

Total: Rp. 10.000

Pembayaran: Cash

Status Pembayaran: Lunas

Simpan Ubah Hapus Batal

ID	Jenis	Quantity	Total
1	Cuci	8	32000
2	Setrika	5	15000
3	Cuci Kilat	3	30000

Simpan Batal

HASIL PRAKTIKUM

Order ID

TRX-00001

Pelanggan

Pilih

Tanggal

Tanggal Pengambilan

Status

Proses

Total

Rp. 10.000

Pembayaran

Cash

Status Pembayaran

Lunas

Simpan

Batal

Service List

ID	Jenis	Harga	Status
1	Cuci	4000	Tersedia
2	Setrika	3000	Tersedia
3	Cuci Kilat	10000	Tersedia

Harga/Kg

3000

Jenis Layanan

Setrika

Jumlah

Total

Simpan

Ubah

Hapus

Batal

ID	Jenis	Quantity	Total
1	Cuci	8	32000
2	Setrika	5	15000
3	Cuci Kilat	3	30000

Order ID

TRX-00001

Pelanggan

Pilih

Tanggal

Tanggal Pengambilan

Status

Proses

Total

Rp. 10.000

Pembayaran

Cash

Status Pembayaran

Lunas

Simpan

Batal

Service List

ID	Jenis	Harga	Status
1	Cuci	4000	Tersedia
2	Setrika	3000	Tersedia
3	Cuci Kilat	10000	Tersedia

Harga/Kg

3000.0

Jenis Layanan

Setrika

Jumlah

4

Total

15000.0

Simpan

Ubah

Hapus

Batal

ID	Jenis	Quantity	Total
1	Cuci	8	32000
2	Setrika	5	15000
3	Cuci Kilat	3	30000

HASIL PRAKTIKUM

Order ID

TRX-00001

Pelanggan

Pilih

Tanggal

Tanggal Pengambilan

Status

Proses

Total

Rp. 77000

Pembayaran

Cash

Status Pembayaran

Lunas

Simpan

Batal

Service List

ID	Jenis	Harga	Status
1	Cuci	4000	Tersedia
2	Setrika	3000	Tersedia
3	Cuci Kilat	10000	Tersedia

Harga/Kg

Jenis Layanan

Jumlah

Total

Simpan

Ubah

Hapus

Batal

ID	Jenis	Quantity	Total
1	Cuci	8	32000
2	Setrika	5	15000
3	Cuci Kilat	3	30000

Order ID

TRX-00001

Pelanggan

Kevin Andhika

Tanggal

27/10/2024

Tanggal Pengambilan

Status

Proses

Total

Rp. 77000

Pembayaran

Cash

Status Pembayaran

Lunas

Simpan

Batal

Service List

ID	Jenis	Harga	Status
1	Cuci	4000	Tersedia
2	Setrika	3000	Tersedia
3	Cuci Kilat	10000	Tersedia

Harga/Kg

Jenis Layanan

Jumlah

Total

Simpan

Ubah

Hapus

Batal

ID	Jenis	Quantity	Total
1	Cuci	8	32000
2	Setrika	5	15000
3	Cuci Kilat	3	30000