**LAPORAN PRAKTIKUM PEMOGRAMAN BERORIENTASI OBJEK (PBO)**

***MEMBUAT FUNGSI CRUD (Create, Read, Update, Delete)***

***DENGAN DATABASE MYSQL***

***(service dan customer)***



OLEH :

RIDHO RADYA PRATAMA

(2311533020)

DOSEN PENGUMPU : Nurfiah, S.ST, M.Kom.

**FAKULTAS TEKNOLOGI INFORMASI**

**DEPARTEMEN INFORMATIKA**

**UNIVERSITAS ANDALAS**

# Pendahuluan

Pada praktikum ini, saya diharapkan mampu mengimplementasikan fungsi CRUD (Create, Read, Update, Delete) untuk data pengguna menggunakan database MySQL. Praktikum ini bertujuan untuk melatih kemampuan saya dalam membuat tabel pengguna di MySQL, menghubungkan aplikasi Java dengan database, membuat tampilan GUI (Graphical User Interface) untuk CRUD pengguna, serta mengaplikasikan konsep Pemrograman Berorientasi Objek (PBO). Selain itu, saya juga akan mempelajari pembuatan dan penggunaan interface serta fungsi DAO (Data Access Object) yang dapat memisahkan logika akses data dari logika bisnis, meningkatkan modularitas, dan memungkinkan perubahan dalam pengelolaan data tanpa mempengaruhi bagian lain dari aplikasi. Alat-alat yang dibutuhkan untuk praktikum ini antara lain komputer yang telah terinstal JDK, Eclipse, MySQL atau XAMPP, serta MySQL Connector untuk menghubungkan Java dengan MySQL.

# Tujuan

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

* + - Mahasiswa mampu membuat table user pada database MySQL
    - Mahasiswa mampu membuat koneksi Java dengan database MySQL
    - Mahasiswa mampu membuat tampilan GUI CRUD user
    - Mahasiswa mampu membuat dan mengimplementasikan interface
    - Mahasiswa mampu membuat fungsi DAO (Data Access Object) dan mengimplementasikannya.
    - Mahasiswa mampu membuat fungsi CRUD dengan menggunakan konsep Pemrograman Berorientasi Objek

# Alat

* + - Computer / laptop yang telah terinstall JDK dan Eclipse
    - MySQL / XAMPP
    - MySQL connector atau Connector/J

# Teori

**XAMPP** yaitu paket software yang terdiri dari Apache HTTP Server, MySQL,PHP dan Perl yang bersifat open source, xampp biasanya digunakan sebagai deleveopment environment dalam pengembangan aplikasi berbasis web secara localhost.

Apache berfungsi sebagai web server yang digunakan untuk menjalankan halaman web, MySQL digunakan untuk manajemen basis data dalam melakukan manipulasi data, PHP digunakan sebagai Bahasa pemrograman untuk membuat aplikasi berbasis web.

**MySQL** adalah sebuah relational database management system (RDBMS) open-source yang digunakan dalam pengelolaan database suatu aplikasi, MySQL ini dapat digunakan untuk menyimpan, mengelola dan mengambil data dalam format table.



Logo MySQL

**MySQL Connection/j** adalah driver yang digunakan untuk menghubungkan aplikasi berbasis java dengan database MySQL sehingga dapat berinteraksi seperti menyimpan, mengubah, mengambil dan menghapus data. Beberapa fungsi MySQL connector yaitu :

* + - Membuka koneksi ke database MySQL
    - Mengirimkan permintaan SQL ke server MySQL
    - Menerima hasil dari permintaan SQL
    - Menutup koneksi ke database MySQL

**DAO (Data Access Object)** merupakan object yang menyediakan abstract interface terhadap beberapa method yang berhubungan dengan database seperti mengambil data (read), menyimpan data(create), menghapus data (delete), mengubah data(update). Tujuan penggunaan DAO yaitu :

* + - Meningkatkan modularitas yaitu memisahkan logika akses data dengan logika bisnis sehingga memudahkan untuk dikelola
    - Meningkatkan reusabilitas yaitu DAO dapat digunakan Kembali
    - Perubahan pada logika akses data dapat dilakukan tanpa mempengaruhi logika bisnis.

**Interface** dalam Bahasa java yaitu mendefinisikan beberapa method abstrak yang harus diimplementasikan oleh class yang akan menggunakannya.

**CRUD** (Create, Read, Update, Delete) merupakan fungsi dasar atau umum yang ada pada sebuah aplikasi yang mana fungsi ini dapat membuat, membaca, mengubah dan menghapus suatu data pada database aplikasi.

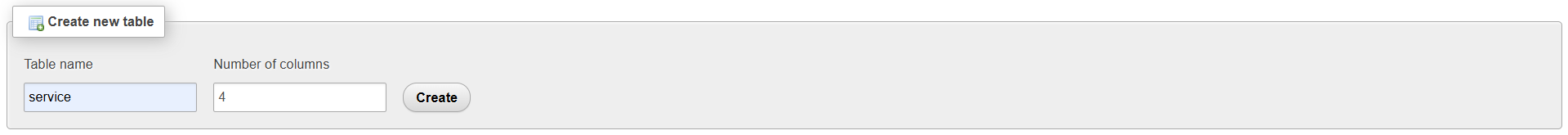
# Langkah-langkah

## XAMPP

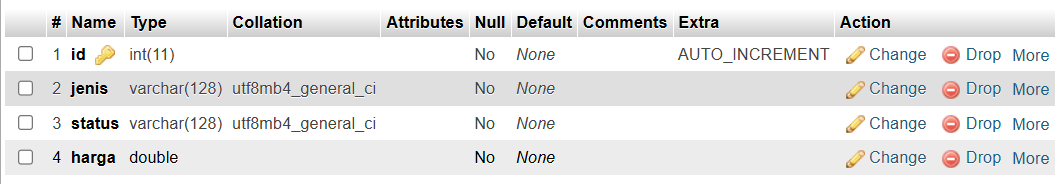
* + - Jalankan xampp dan aktifkan apache dan mysql 

## Membuat Database dan Table Service dan Table Costumer

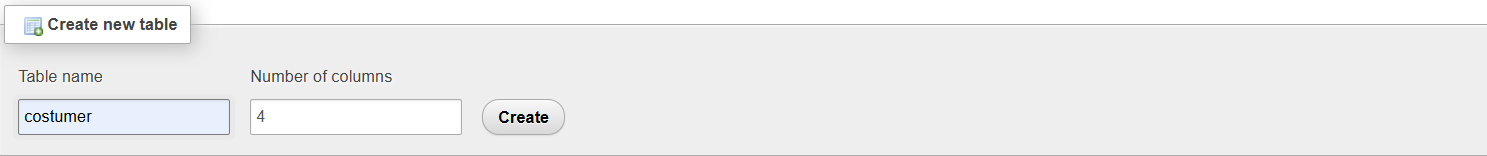
* + - Buka <http://localhost/phpmyadmin>
    - Buat table service dengan cara klik database laundry\_apps dan buat table dengan nama service



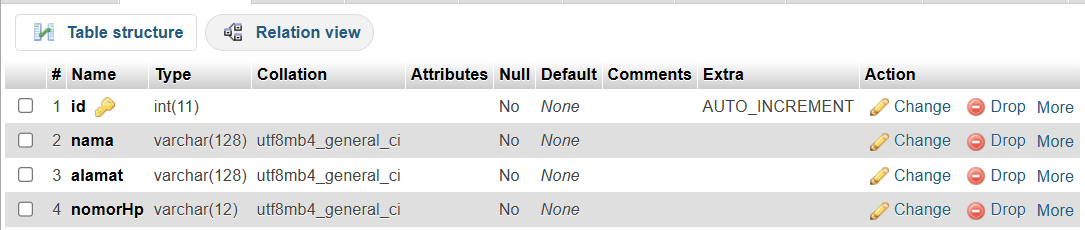
Klik create, kemudian isi tabel dengan ‘id’, ‘jenis’, ‘status’, dan ‘harga’.Maka akan muncul seperti gambar dibawah ini.



* + - Buat table service dengan cara klik database laundry\_apps dan buat table dengan nama Costumer

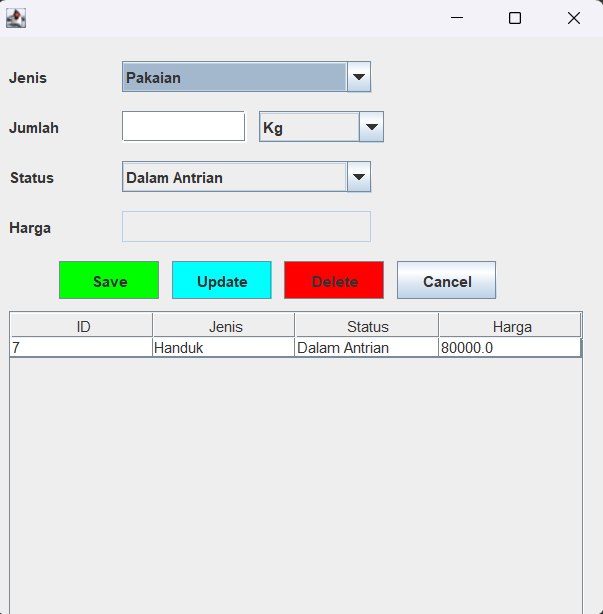


Klik create, kemudian isi tabel dengan ‘id’, ‘jenis’, ‘status’, dan ‘harga’.Maka akan muncul seperti gambar dibawah ini.



## Membuat Tampilan CRUD Service dan Customer

* + - Buat file baru menggunakan JFrame pada package ui dengan nama ServiceFrame seperti gambar berikut ini.

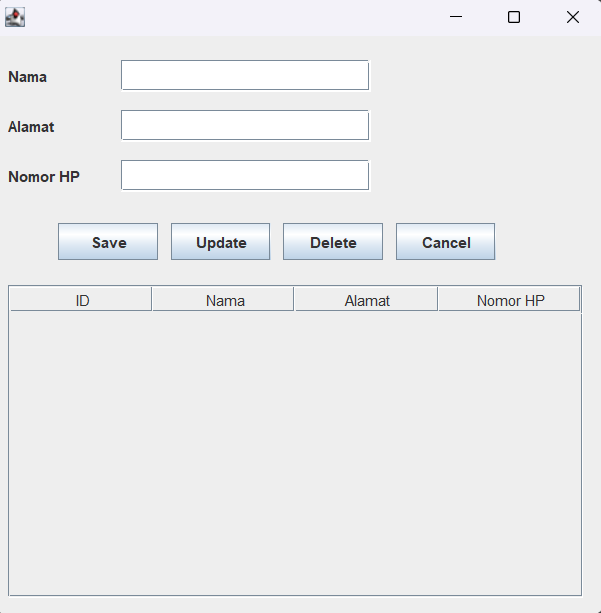


Keterangan :

|  |  |  |
| --- | --- | --- |
| **Component** | **Variable** | **Keterangan** |
| JComboBox | comboJenis | Jenis |
| JTextField | txtQuantity | Nilai Satuan |
| JComboBox | comboSatuan | Satuan |
| JComboBox | comboStatus | Status |

|  |  |  |
| --- | --- | --- |
| JTextField | txtHarga | Harga |
| JButton | btnSave | Save |
| JButton | btnUpdate | Update |
| JButton | btnDelete | Delete |
| JButton | btnCancel | Cancel |
| JTable | tableService | Table Service |

* + - Buat file baru menggunakan JFrame pada package ui dengan nama CostumerFrame seperti gambar berikut ini.



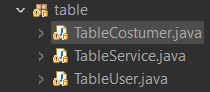
Keterangan :

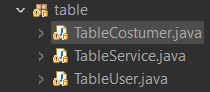
|  |  |  |
| --- | --- | --- |
| **Component** | **Variable** | **Keterangan** |
| JTextField | txtNama | Nama |
| JTextField | txtAlamat | Alamat |

|  |  |  |
| --- | --- | --- |
| JTextField | txtNoHp | Nomor HP |
| JButton | btnSave | Save |
| JButton | btnUpdate | Update |
| JButton | btnDelete | Delete |
| JButton | btnCancel | Cancel |
| JTable | tableService | Table Service |

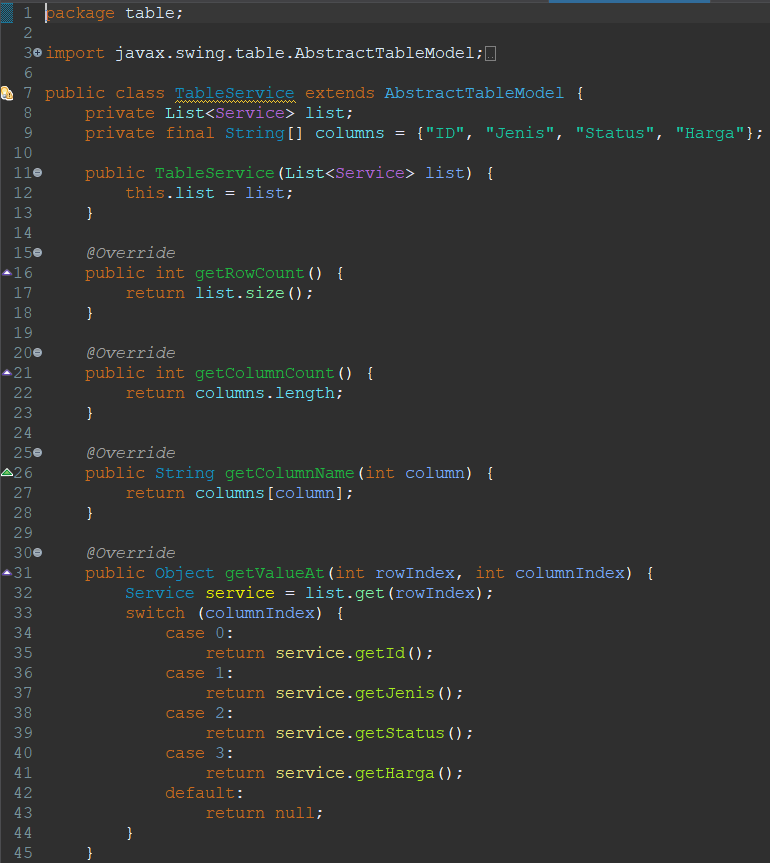
## Membuat Table Model

Table model user ini berguna untuk mengambil data dari database dan ditampilkan kedalam table.

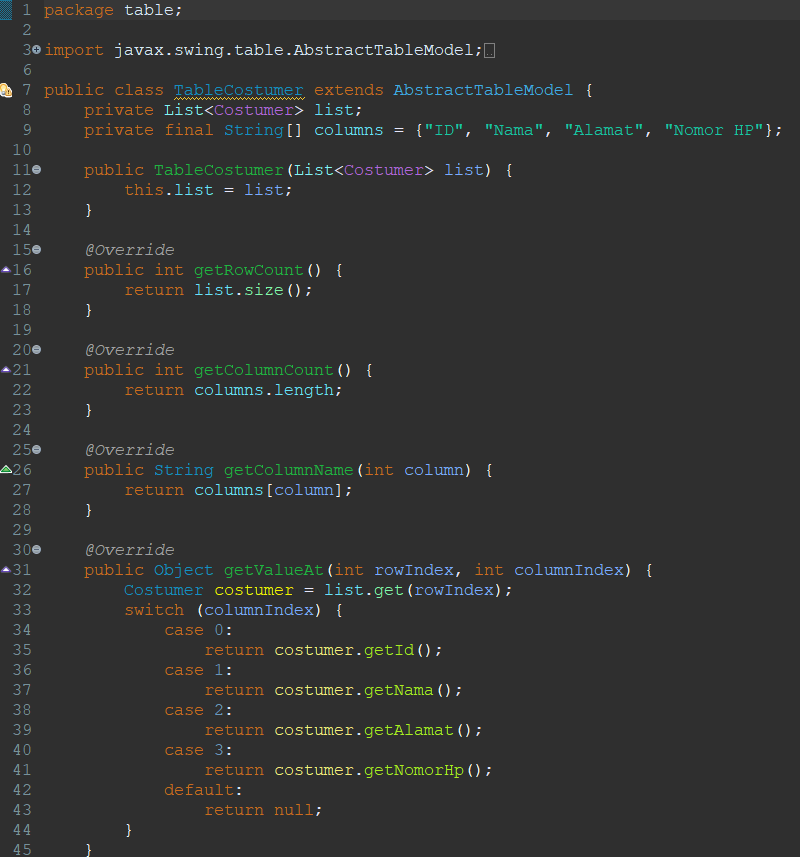
* + - Buat file baru didalam package table dengan nama **TableService** ****
    - Buat file baru didalam package table dengan nama **TableCustomer**



kemudian isikan TableService dengan kode program berikut.



kemudian isikan TableCustomer dengan kode program berikut.

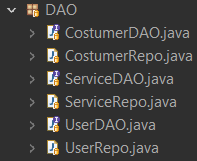


## Membuat Fungsi DAO

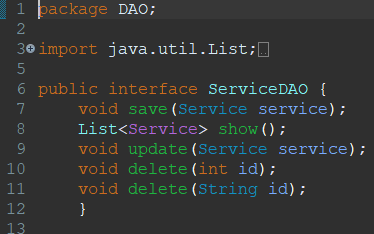
* + - Pada package DAO



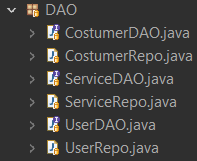
* + - Buat class Interface baru dengan nama **ServiceDAO**,



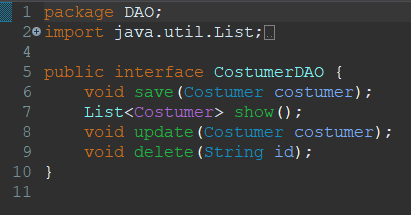
kemudian isikan dengan kode program berikut.



* + - Buat class Interface baru dengan nama **CostumerDAO**,

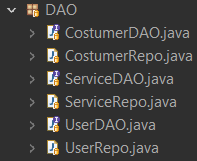
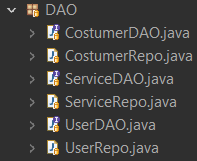


kemudian isikan dengan kode program berikut.



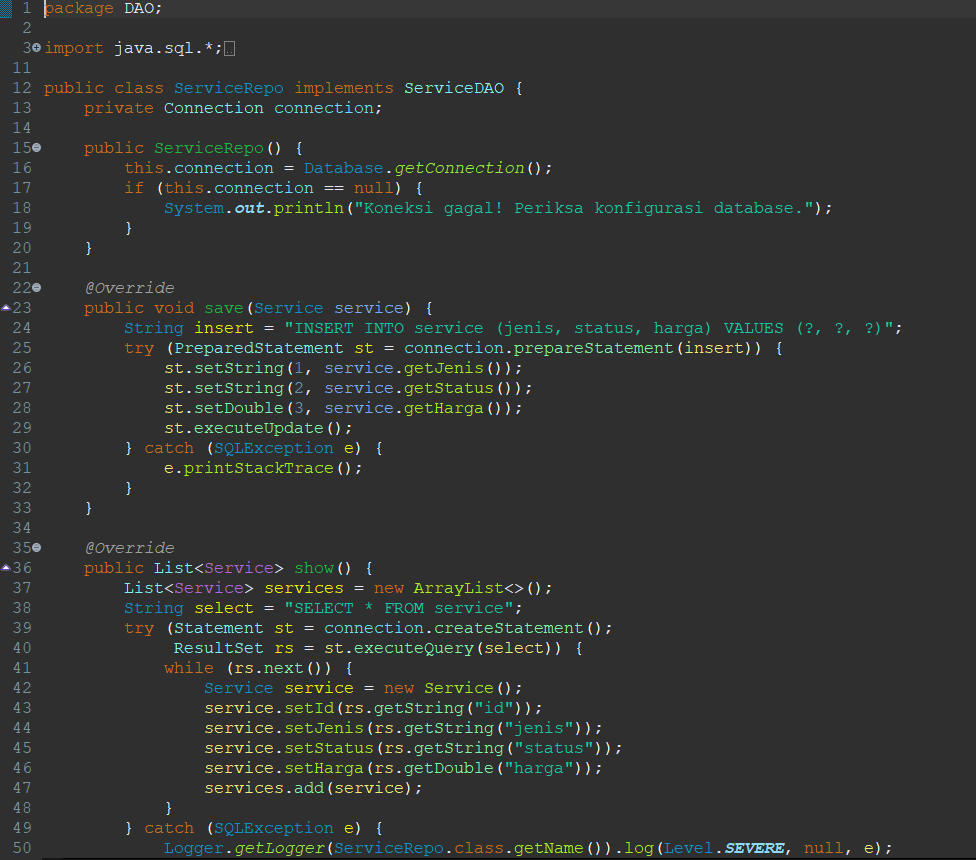
Terdapat method **save, show, delete dan update**. Method pada class interface digunakan sebagai method utama yang wajib diimplementasikan pada class yang menggunakannya.

## Menggunakan Fungsi DAO

* + - Buat class baru pada package DAO dengan nama **ServiceRepo** yang mana akan digunakan untuk mengimplementasikan DAO yang telah dibuat.  
      
    - Buat class baru pada package DAO dengan nama **CostumerRepo** yang mana akan digunakan untuk mengimplementasikan DAO yang telah dibuat.  
      
    - Implementasikan ServiceDao dengan kata kunci **implements**

****

* + - Masukkan kode di bawah ini



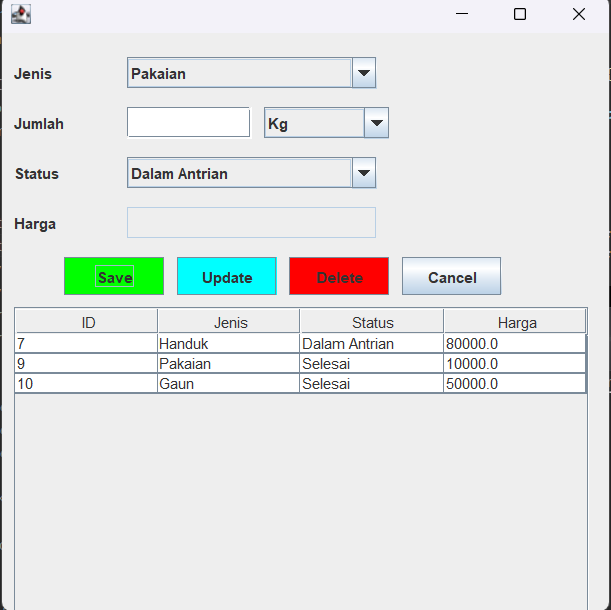
* + - Implementasikan CostumerDao dengan kata kunci **implements**

****

* + - Masukkan kode di bawah ini



**Service**

****

Pada service, terdapat form **Jenis**, Quantity, Status, Harga. Untuk Jenis digunakan comboBox denga nisi jenis-jenis pakaian, yaitu “pakaian, Seprei, Handuk, Jas, Gaun” .

Selanjutnya adalah **Quantity** untuk nilai satuan dan satuan nya, di mana jika Pakaian maka otomatis terisi Kg dan selain itu dengan satuan Helai.

Kemudian **Status** untuk mengetahui status laundry, menggunakan comboBox, denga nisi “Dalam antrian, Sedang di cuci, Selesai”

Terakhir **Harga** yang otomatis terisi sesuai dengan harga x satuan. Dimana masing-masing harga yang di tetapkan adalah

private final double PAKAIAN\_PRICE\_PER\_KG = 5000;

private final double SEPREI\_PRICE\_PER\_HELAI = 35000;

private final double JAS\_PRICE\_PER\_HELAI = 20000;

private final double GAUN\_PRICE\_PER\_HELAI = 50000;

private final double HANDUK\_PRICE\_PER\_HELAI = 10000;

dan untuk ‘x quantity’ menggunakan

case "Pakaian":

if (satuan.equals("Kg")) {

harga = quantity \* PAKAIAN\_PRICE\_PER\_KG;

}

break;

case "Seprei":

if (satuan.equals("Helai")) {

harga = quantity \* SEPREI\_PRICE\_PER\_HELAI;

}

break;

case "Handuk":

if (satuan.equals("Helai")) {

harga = quantity \* HANDUK\_PRICE\_PER\_HELAI;

}

break;

case "Jas":

if (satuan.equals("Helai")) {

harga = quantity \* JAS\_PRICE\_PER\_HELAI;

}

break;

case "Gaun":

if (satuan.equals("Helai")) {

harga = quantity \* GAUN\_PRICE\_PER\_HELAI;

}

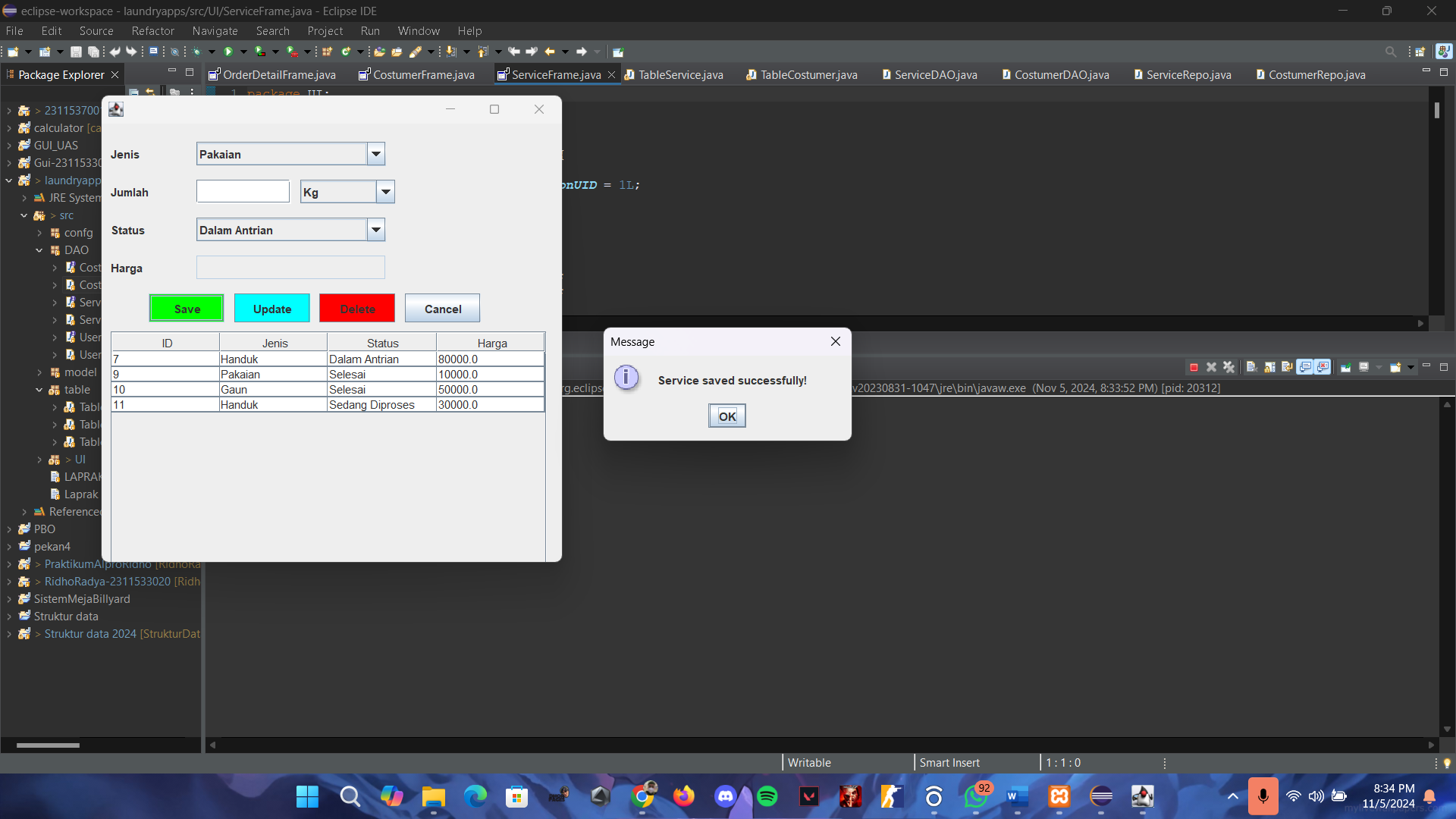
break;

default:

break;

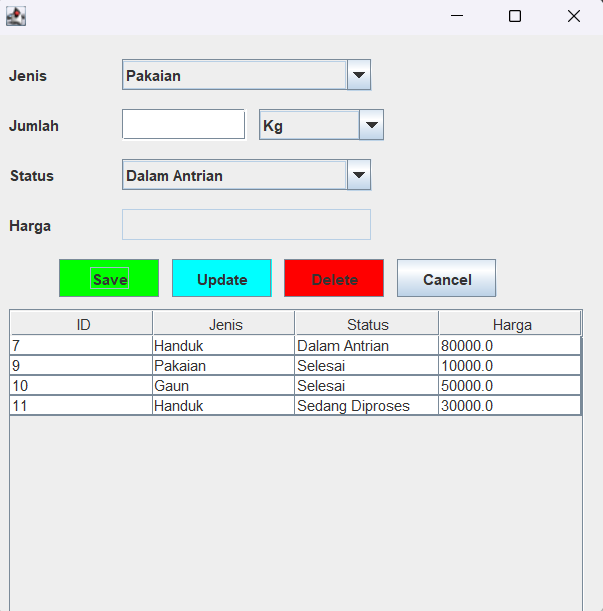
}

Tombol **Save** pada service



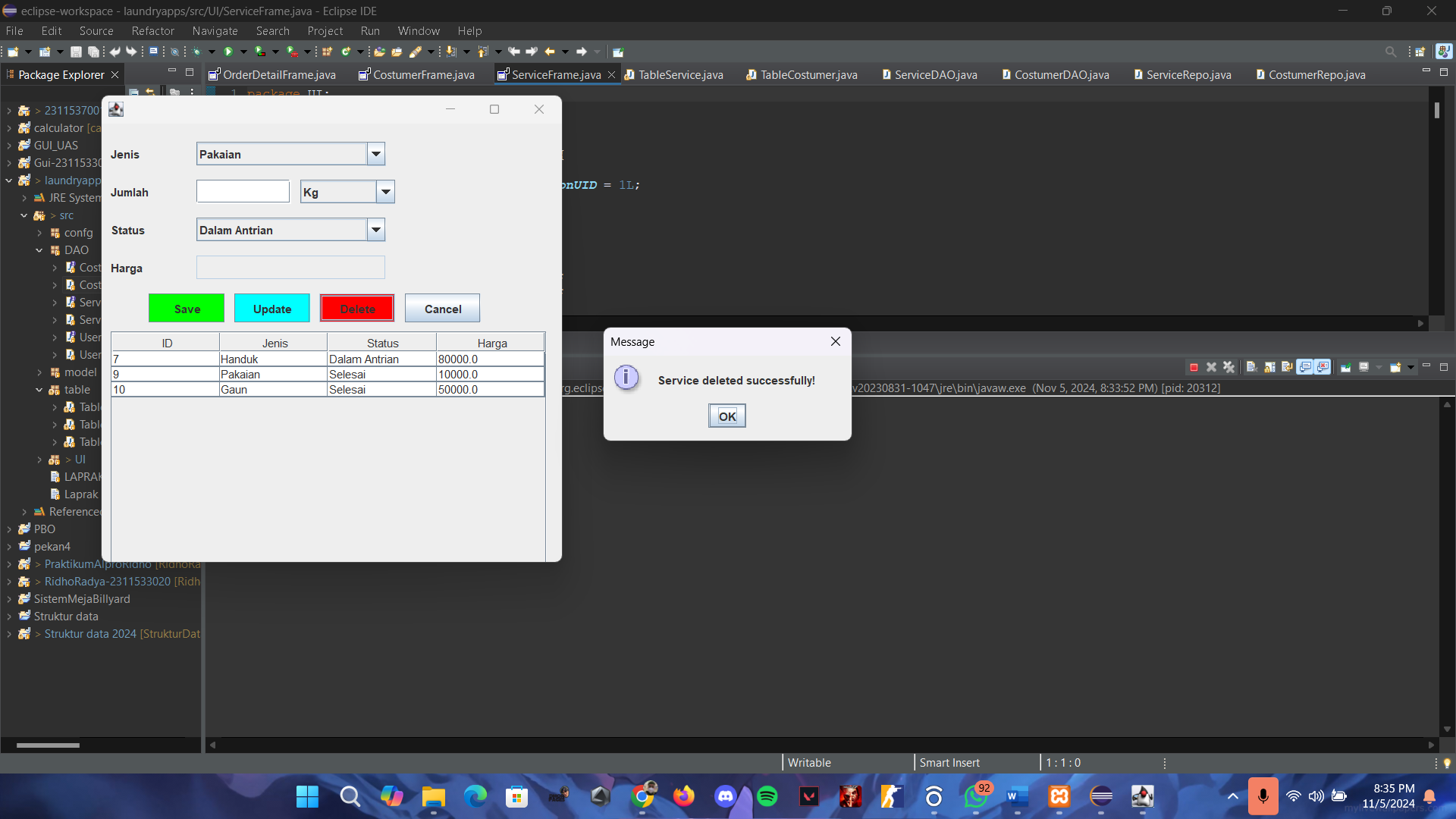
Berhasil menyimpan data yang telah di masukkan

Tombol **Update**

****

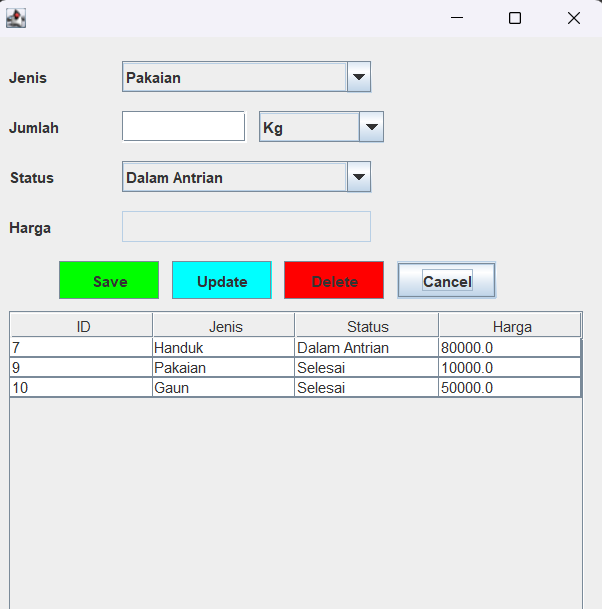
Berhasil merubah data yang telah di masukkan, semula Handuk menjadi Pakaian di nomor 10

tombol **Delete**



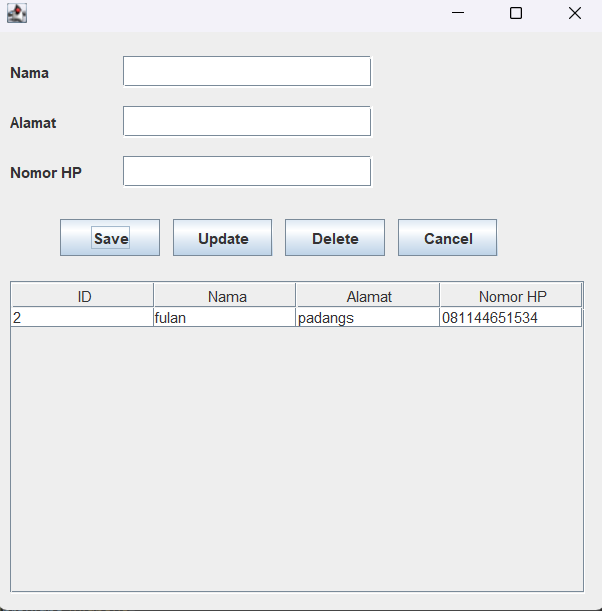
Berhasil menghapus data

Tombol cancel

****

Untuk men cancel isi form yang telah di isi sebelumnya

**Customer**

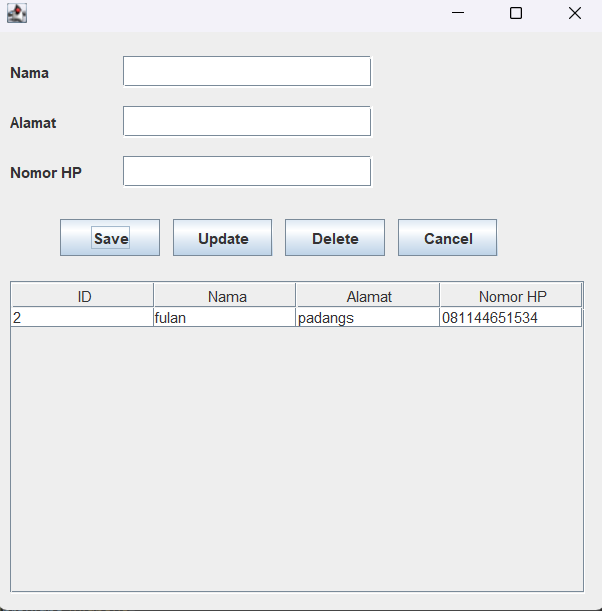
****

Pada service, terdapat form **Nama**, untuk mengisi nama customer

Selanjutnya adalah **Alamat** untuk mengisi alamat customer

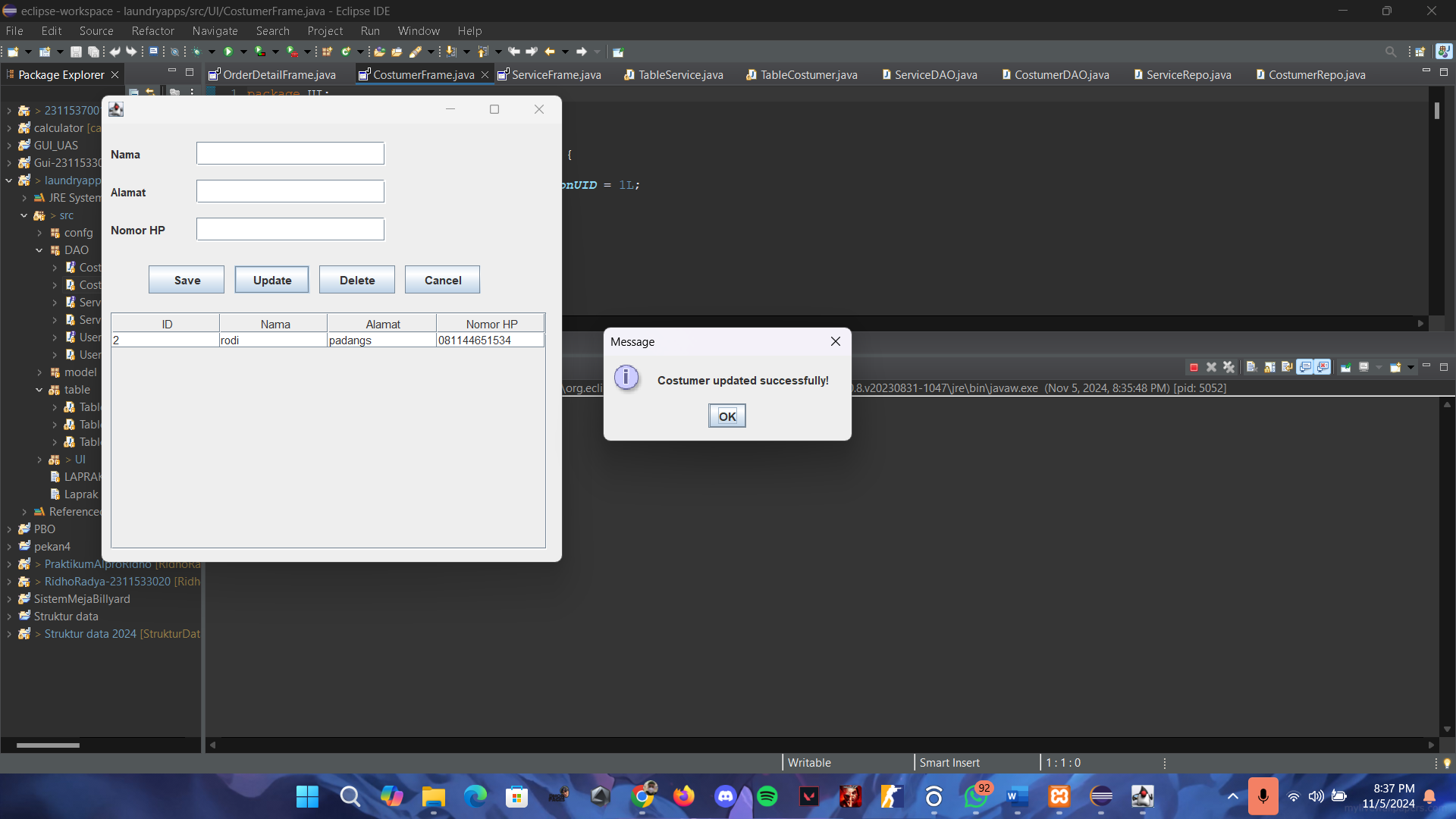
Kemudian **no hp** untuk mengisi nomor hp customer

Tombol **Save** pada costumer

****

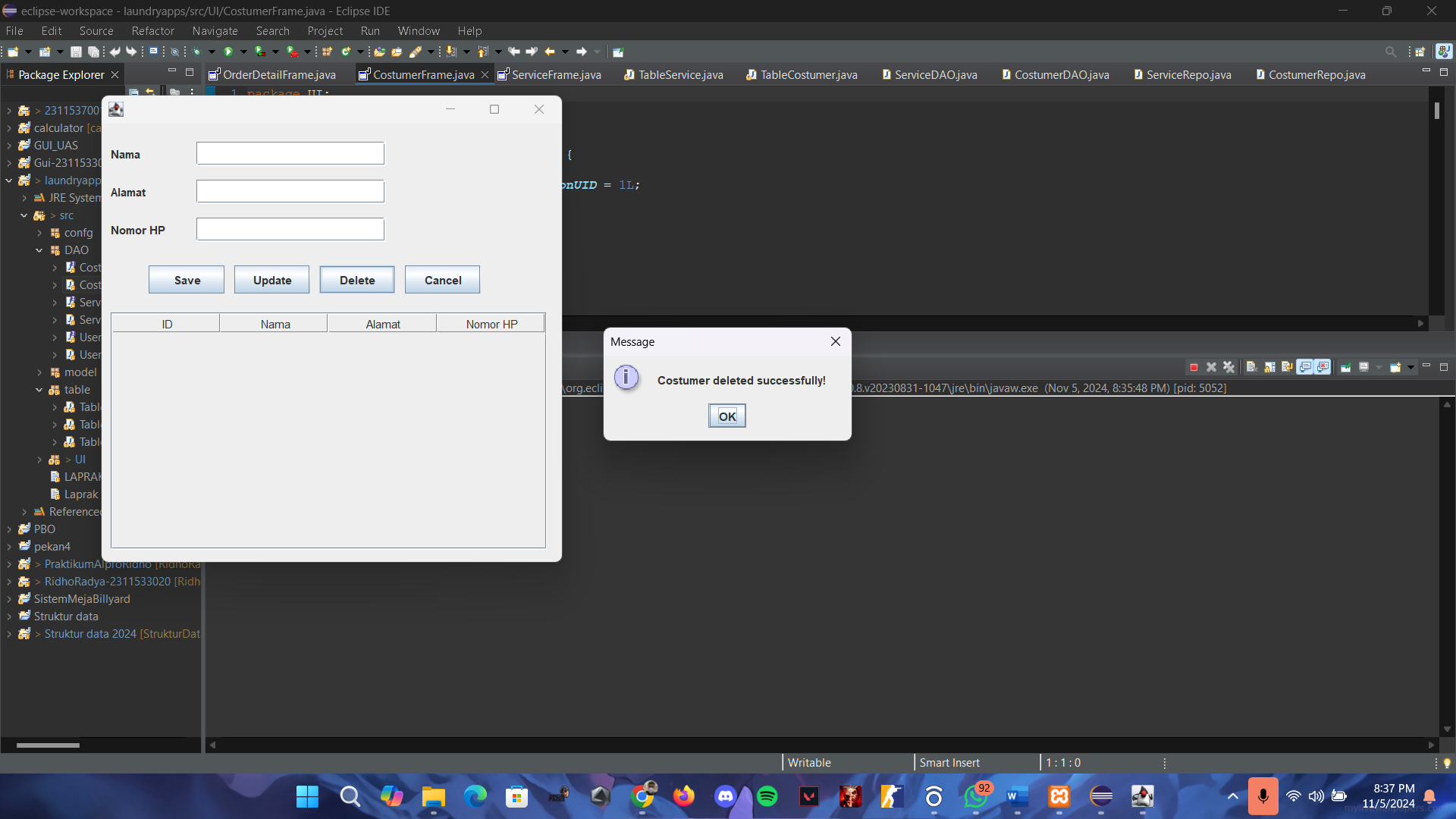
Berhasil menyimpan data yang telah di masukkan

Tombol **Update**



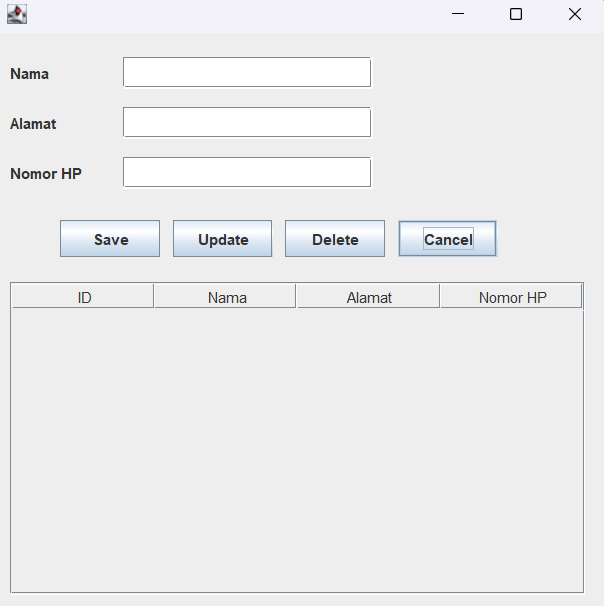
Berhasil merubah data yang telah di masukkan, semula fulan menjadi fulanah di nomor 4

tombol **Delete**



Berhasil menghapus data

Tombol cancel

****

Untuk men cancel isi form yang telah di isi sebelumnya

# Penutup

Melalui praktikum ini, saya telah mempelajari dan mempraktikkan pembuatan fungsi CRUD menggunakan Java dan MySQL dengan menerapkan konsep Pemrograman Berorientasi Objek. Saya juga telah berhasil membuat GUI untuk mengelola data pengguna serta memanfaatkan fungsi DAO untuk modularitas dan pemisahan logika. Dengan pemahaman yang didapatkan, saya diharapkan mampu mengimplementasikan konsep-konsep ini pada proyek-proyek pengembangan perangkat lunak di masa mendatang, terutama yang melibatkan pengelolaan data berbasis databas