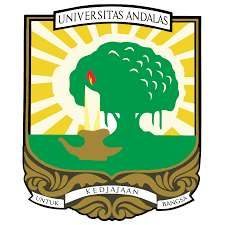
**LAPORAN PRAKTIKUM 4 & 5 PBO**

**MEMBUAT FUNGSI ORDER LAUNDRY**

****

**Disusun Oleh:**

**INDAH SYAHFITRI**

**2311532016**

**Dosen Pembimbing: Nurfiah, S. ST., M. Kom.**

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

1. **PENDAHULUAN**

XAMPP yaitu paket software yang terdiri dari Apache HTTP Server, MySQL,PHP dan Perl yang bersifat open source, XAMPP biasanya digunakan sebagai *development 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.

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:

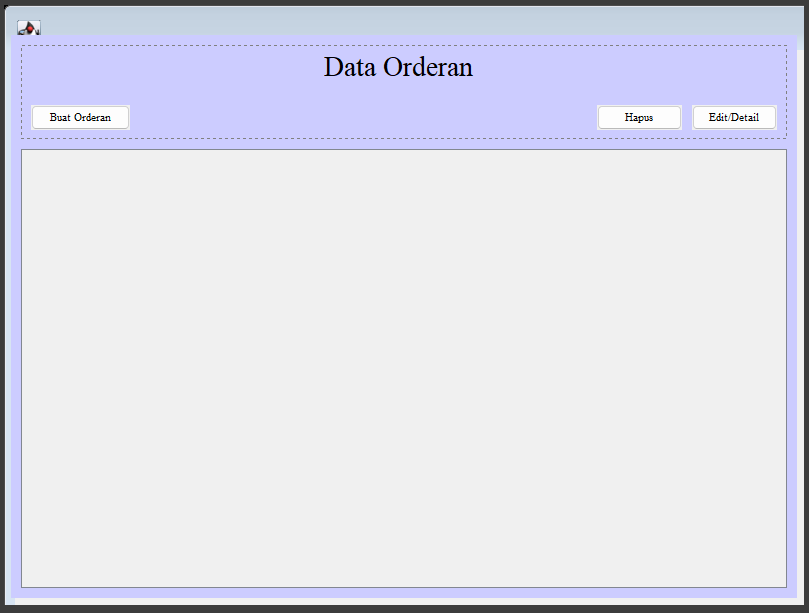
• Meningkatkan modularitas yaitu memisahkan logika akses data dengan logika bisnis sehingga memudahkan untuk dikelola.

• Meningkatkan reusabilitas yaitu DAO dapat digunakan kembali.

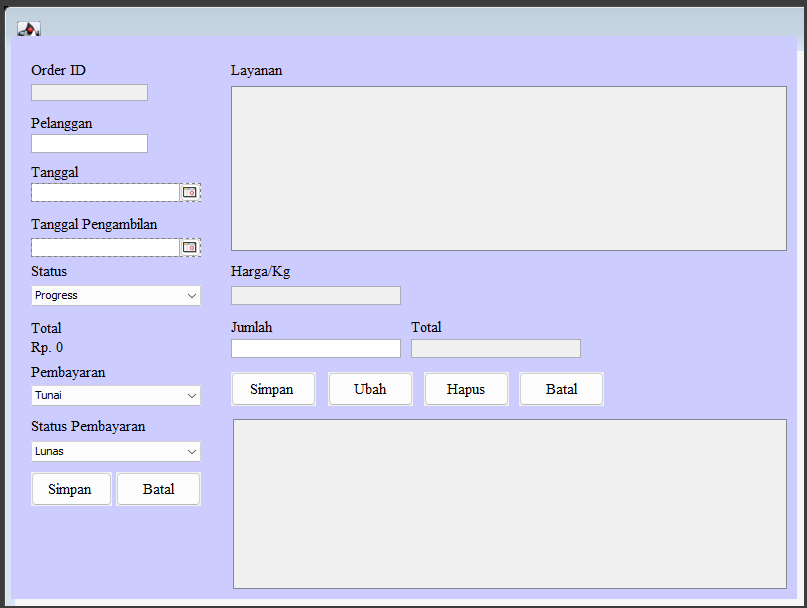
• Perubahan pada logika akses data dapat dilakukan tanpa mempengaruhi logika bisnis.

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

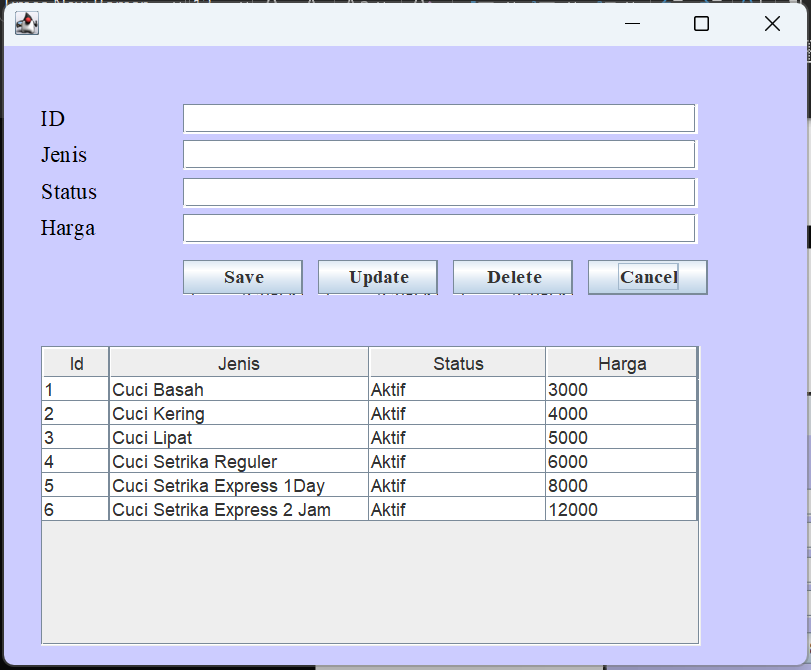
1. **TUJUAN PRAKTIKUM**
2. Mahasiswa mampu membuat UI untuk melakukan order laundry.
3. Mahasiswa mampu membuat fungsi pemesanan multi layanan laundry baik menambahkan, mengubah dan menghapusnya.
4. Mahasiswa mampu membuat fungsi pemesanan laundry.
5. **LANGKAH-LANGKAH**
6. Pertama-tama, buatlah kelas windowbuilder dengan nama OrderFrame lalu desain tampilannya hingga seperti berikut.



1. Selanjutnya, buatlah kelas windowbuilder dengan nama OrderDetail lalu desain tampilannya hingga seperti berikut.

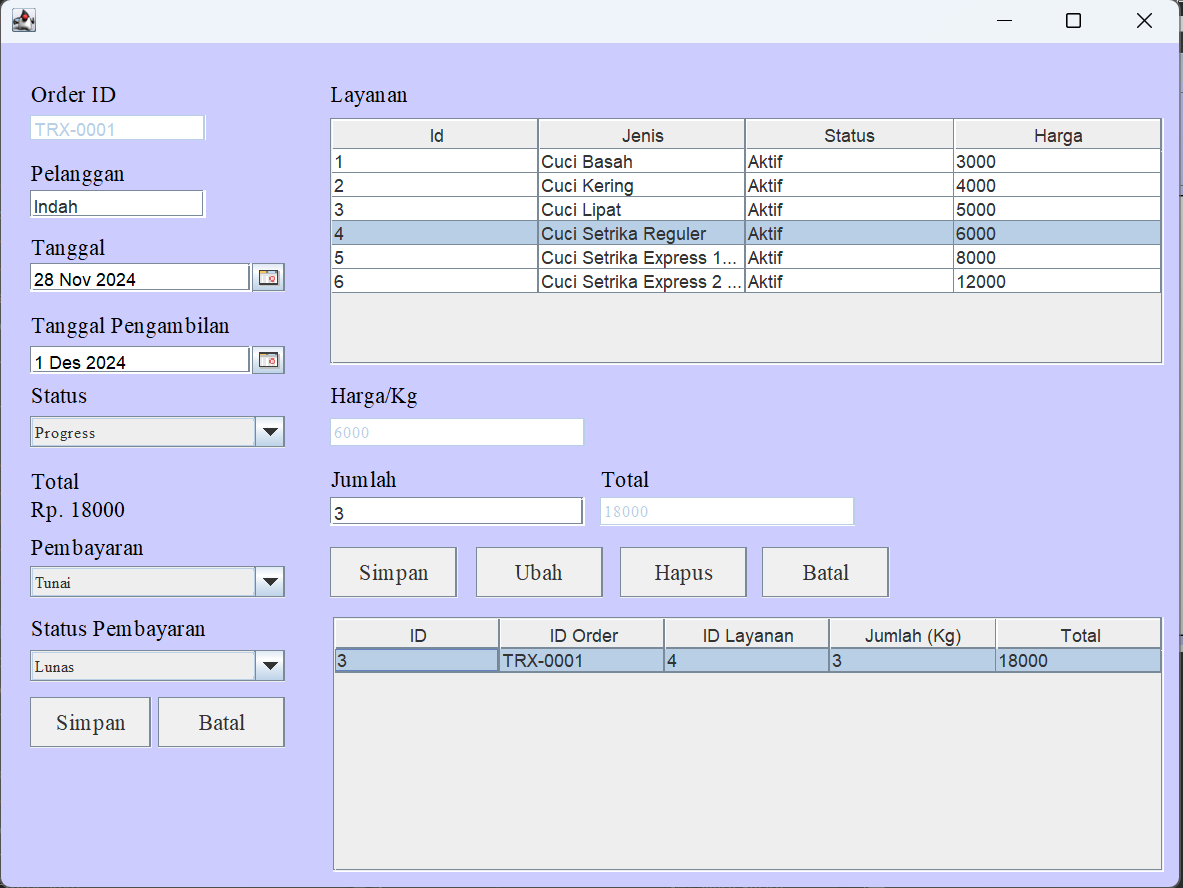


1. Cara kerjanya yakni akan tampil UI OrderFrame yang mana pada tampilan ini akan ditampilkan data orderan laundry, tombol untuk menambah orderan, menghapus dan melakukan perubahan data order. Jika pengguna menekan tombol buat orderan, maka UI OrderDetail akan muncul untuk diisi. Pada UI Order Detail akan ditampilkan layanan-layanan yang tersedia pada aplikasi laundry misalkan cuci, cuci + setrika, setrika dll yang telah dibuat di ServiceFrame sebelumnya.



1. Operator akan memilih jenis layanan kemudian secara otomatis pada form harga/kg akan terisi dengan harga sesuai dengan layanan yang dipilih. Operator akan menginputkan jumlah dari laundry yang dipesan maka secara otomatis total akan terisi dengan hasil perkalian harga dengan jumlah.

Selanjutnya operator akan klik tombol simpan untuk menyimpan pesanan, jika pelanggan akan memesan layanan lain maka dapat dilakukan dengan cara yang sama di atas. Jika ada perubahan terhadap pesanan maka operator dapat memilik pesanan maka otomatis terisi kolom harga, jumlah dan total. Untuk melakukan perubahan klik ubah, untuk menghapus klik hapus dan jika ingin membatalkan klik batal. Selanjutnya jika pesanan sudah selesai, operator akan memilih nama pelanggan, tanggal pemesanan, tanggal pengambilan, status, jenis pembayaran dan status pembayaran kemudian klik simpan untuk menyimpan pemesanan laundry.

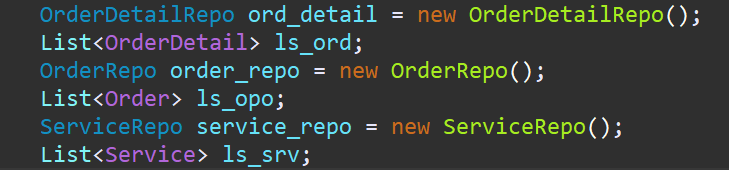


1. Untuk menyimpan detail order, buat table baru di database dengan nama table\_order dan struktur seperti berikut.

Sebuah gambar berisi teks, Font, nomor, cuplikan layar

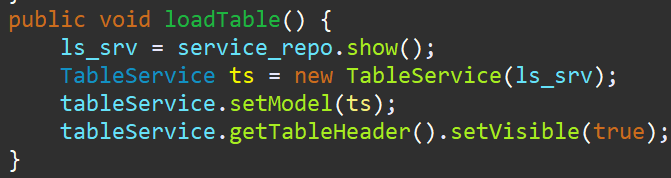
Deskripsi dibuat secara otomatis

1. Buat kode seperti ini di dalam kelas OrderDetailFrame.



Kode tersebut membuat beberapa objek repository dan list untuk mengelola data dalam aplikasi. Objek seperti OrderDetailRepo, OrderRepo, dan ServiceRepo digunakan untuk mengakses dan memproses data terkait detail pesanan, pesanan, dan layanan. Sementara itu, list seperti ls\_ord, ls\_opo, dan ls\_srv berfungsi sebagai wadah untuk menyimpan kumpulan data yang diambil atau diproses dari repository tersebut.

1. Buat method untuk menampilkan data layanan di kelas yang sama.

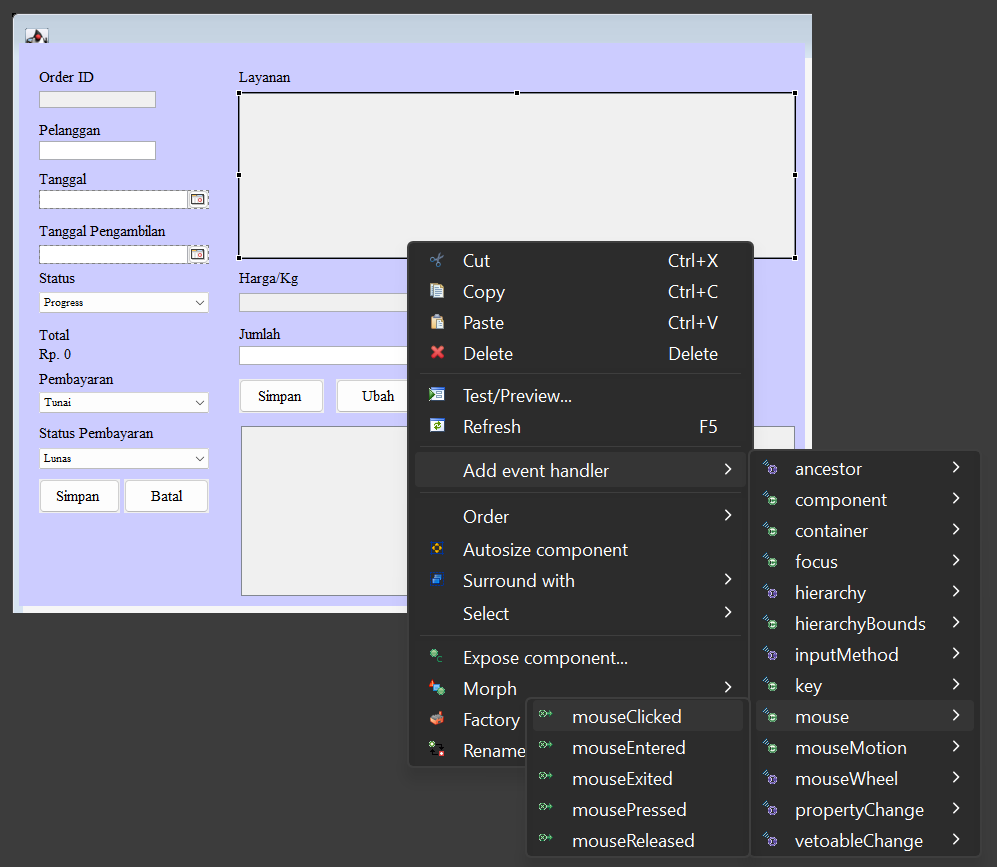


Metode loadTable() bertugas memuat data layanan ke dalam tabel. Pertama, data layanan diambil dari repository menggunakan metode show() dan disimpan dalam list ls\_srv. Data tersebut kemudian digunakan untuk membuat objek TableService, yang berfungsi sebagai model tabel. Model ini diterapkan ke komponen tabel tableService, dan bagian header tabel diatur agar terlihat dengan metode getTableHeader().setVisible(true). Hal ini memastikan tabel ditampilkan dengan data yang terstruktur dan header yang jelas.

1. Selanjutnya, panggil method loadTable() ke dalam main Frame agar data layanan muncul ke dalam table ketika frame diload.



1. Lalu, agar form harga terisi otomatis ketika layanan dipilih, klik kanan pada table service → Add event handler → mouse → mouseClicked.

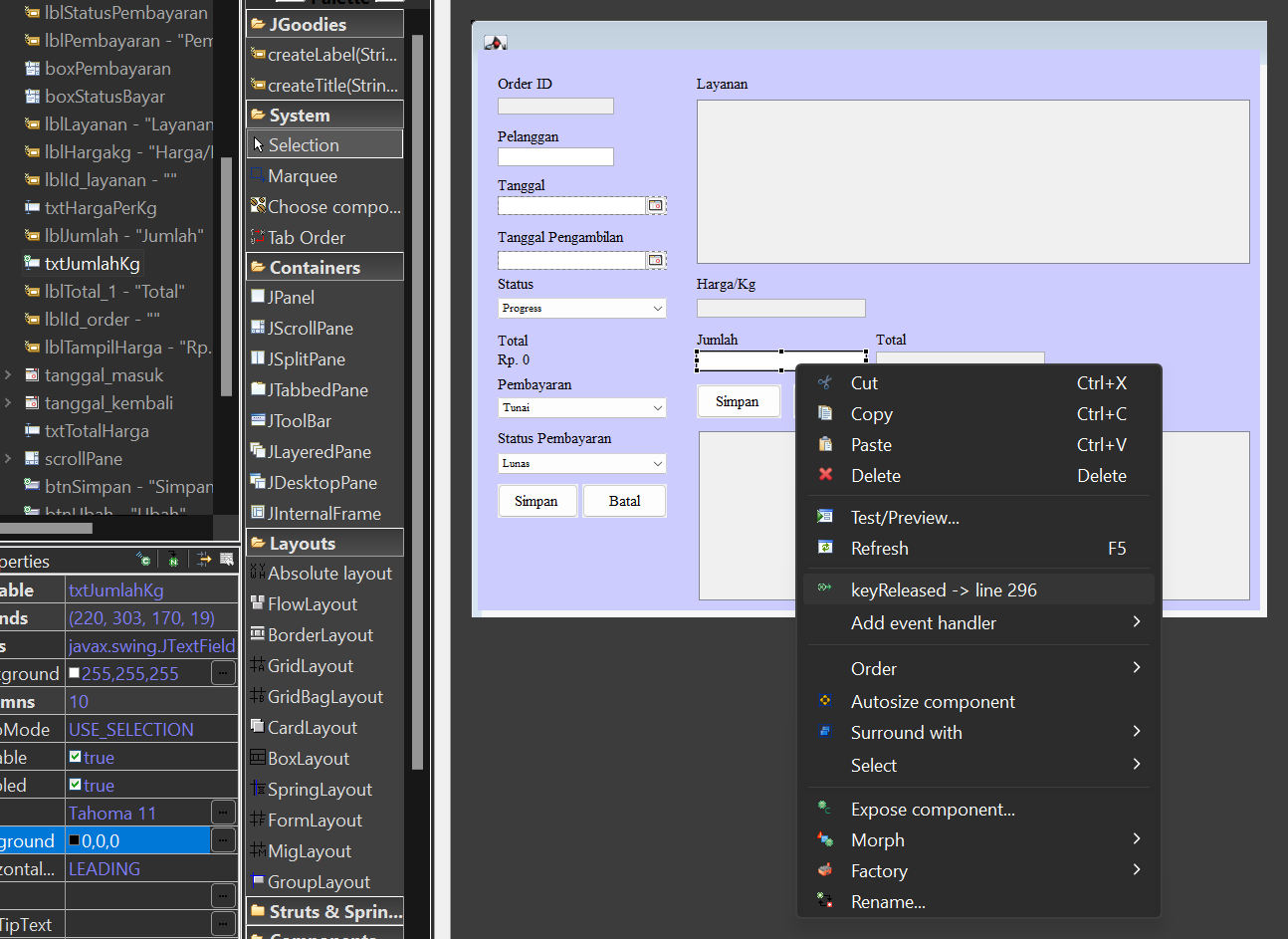


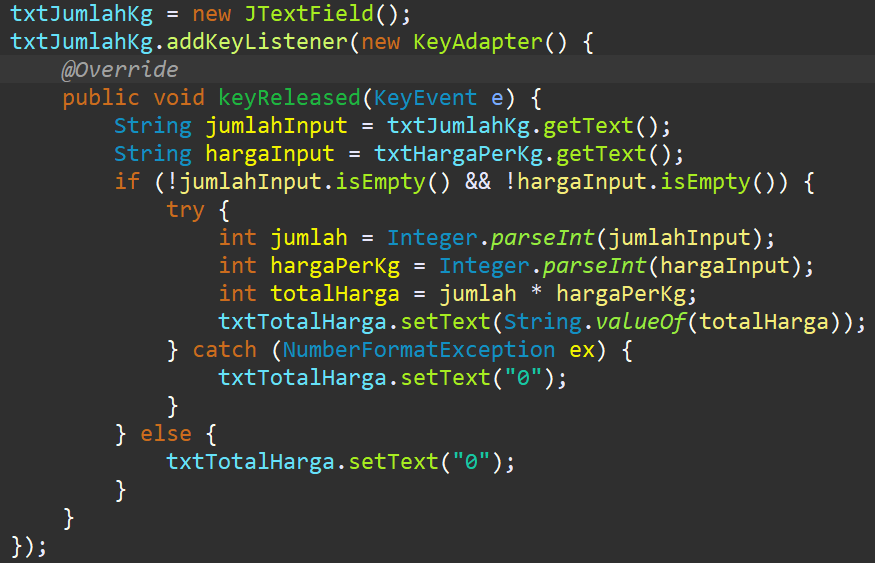
Tambahkan program berikut.



Kode tersebut menangani aksi klik pada tabel tableService. Ketika pengguna mengklik salah satu baris, program akan mendapatkan baris yang dipilih (selectedRow) dan mengambil nilai ID layanan dari kolom pertama pada baris tersebut. Nilai harga per kilogram dari kolom keempat akan ditampilkan di txtHargaPerKg, dan ID layanan akan ditampilkan di label lblId\_layanan. Selanjutnya, program memeriksa apakah layanan dengan ID tersebut sudah ada di detail pesanan (order detail) menggunakan metode cekId\_Order. Jika layanan sudah ada, akan muncul pesan peringatan melalui kotak dialog. Jika tidak, program tidak melakukan tindakan tambahan.

1. Ketika form harga sudah terisi, maka saat operator memasukkan jumlah maka form total akan terisi secara otomatis. Untuk membuat sistem seperti ini, klik kanan JTextField jumlah → Add event handler → key → keyReleased lalu tambahkan kode berikut.





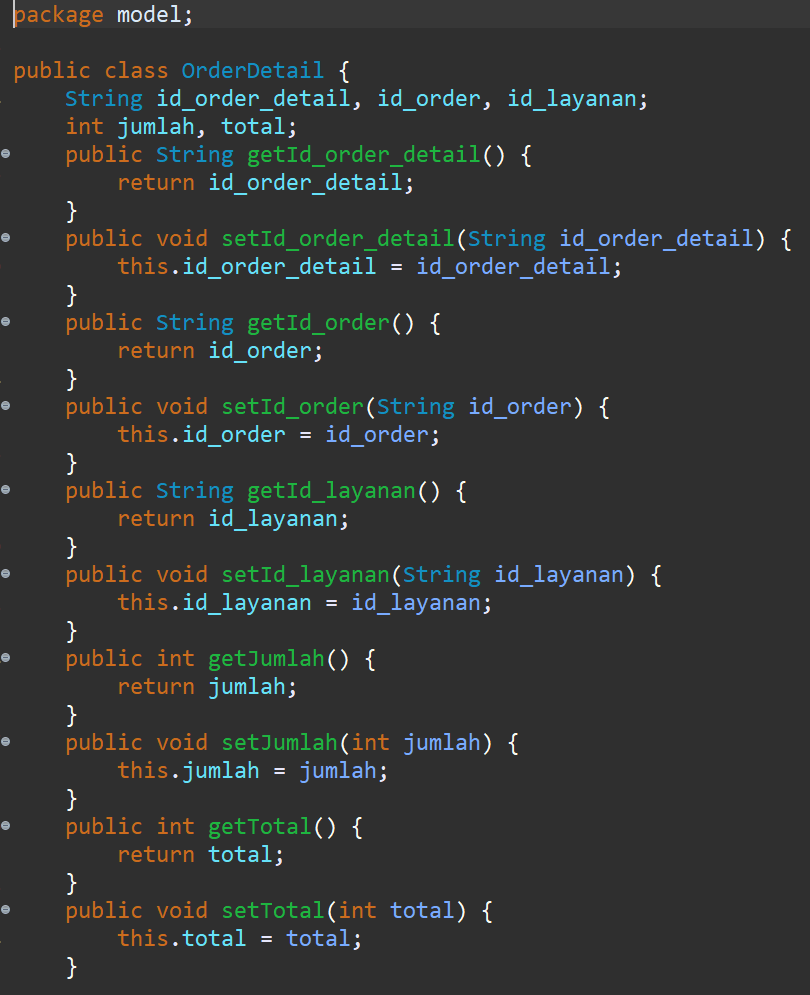
Kode ini berfungsi untuk menghitung total harga secara otomatis berdasarkan input jumlah kilogram dan harga per kilogram di sebuah form. Ketika pengguna mengetikkan jumlah kilogram di txtJumlahKg, metode keyReleased akan dijalankan. Kode ini mengambil nilai dari txtJumlahKg dan txtHargaPerKg, lalu mengalikan keduanya untuk mendapatkan total harga. Jika input valid, hasil perhitungan akan ditampilkan di txtTotalHarga. Jika terjadi kesalahan input, seperti memasukkan nilai non-angka, maka txtTotalHarga akan diatur menjadi "0".

1. Selanjutnya, tambahkan file baru pada package DAO dengan nama OrderDetailDAO kemudian tambahkan kode program berikut.

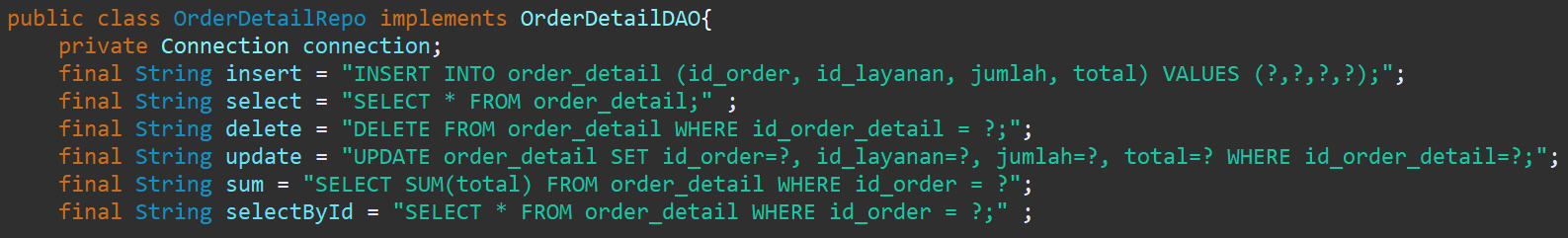


Terdapat 5 buah method yaitu Save digunakan untuk menyimpan data order detail dengan Argument Model OrderDetail, Show digunakan untuk menampilkan data OrderDetail dengan argument/parameter order\_id, Delete digunakan untuk menghapus data, Update digunakan untuk melakukan perubahan terhadap data, Total digunakan untuk menghitung jumlah total order detail berdasarkan order\_id.

1. Tambahkan file baru pada package model dengan nama OrderDetail kemudian tambahkan kode program berikut untuk inisialisasi dan memasang getter setter atribut yang akan digunakan.



1. Tambahkan file baru pada package DAO dengan nama OrderDetailRepo, pada file ini akan dilakukan proses manipulasi data. File ini akan mengimplements file OrderDetailDAO.



1. Untuk menyimpan Order Detail, buat kode berikut.

Sebuah gambar berisi teks, cuplikan layar, Font

Deskripsi dibuat secara otomatis

Kode ini berfungsi untuk menyimpan data OrderDetail ke dalam database. Pertama, kode ini mempersiapkan perintah SQL menggunakan PreparedStatement dengan query insert. Kemudian, nilai-nilai dari objek OrderDetail (seperti id\_order, id\_layanan, jumlah, dan total) dimasukkan ke dalam perintah SQL dengan metode setString dan setInt. Setelah itu, perintah SQL dieksekusi dengan executeUpdate untuk memasukkan data ke dalam database. Jika terjadi kesalahan (SQLException), pesan kesalahan akan dicetak. Di bagian akhir, objek PreparedStatement ditutup untuk melepaskan sumber daya yang digunakan, meskipun jika ada kesalahan saat penutupan, itu juga akan ditangani.

1. Untuk menampilkan Order Detail, buat kode berikut.



Kode ini adalah metode show() yang digunakan untuk mengambil data dari tabel OrderDetail dalam basis data. Di dalam metode ini, pertama-tama dibuat sebuah list kosong ls untuk menyimpan objek OrderDetail. Kemudian, kode ini menjalankan query SQL dengan menggunakan Statement untuk mengambil data dari basis data. Setiap baris hasil query (ResultSet) diambil, dan data dari setiap kolom seperti id\_order\_detail, id\_order, id\_layanan, jumlah, dan total dimasukkan ke dalam objek OrderDetail. Setelah itu, objek tersebut ditambahkan ke dalam list ls. Jika terjadi kesalahan saat proses pengambilan data, maka error akan dicatat menggunakan logger. Terakhir, list yang berisi objek OrderDetail dikembalikan sebagai hasil.

1. Untuk menghapus Order Detail, buat kode berikut.



Kode ini adalah metode delete yang digunakan untuk menghapus data dari database berdasarkan id yang diberikan. Metode ini mempersiapkan query SQL untuk menghapus data menggunakan PreparedStatement. id yang diberikan sebagai parameter diatur ke dalam query menggunakan st.setString(1, id). Setelah itu, query dieksekusi dengan st.executeUpdate(). Jika terjadi kesalahan selama proses penghapusan, maka pengecualian SQLException akan ditangani dan pesan kesalahan akan dicatat dengan menggunakan Logger. Setelah eksekusi selesai, koneksi dengan statement (st) akan ditutup di dalam blok finally untuk memastikan tidak ada kebocoran sumber daya.

1. Untuk mengupdate Order Detail, buat kode berikut.

Sebuah gambar berisi teks, cuplikan layar

Deskripsi dibuat secara otomatis

Kode di atas berfungsi untuk memperbarui data dalam tabel Order\_Detail di database. Metode update menerima objek OrderDetail sebagai parameter, yang berisi informasi yang akan diupdate. Kode ini menggunakan PreparedStatement untuk menyiapkan dan mengeksekusi query SQL yang memperbarui kolom-kolom pada tabel berdasarkan nilai-nilai yang ada dalam objek OrderDetail. Parameter yang diatur meliputi id\_order, id\_layanan, jumlah, total, dan id\_order\_detail. Jika terjadi kesalahan saat eksekusi, kesalahan tersebut akan dicatat dengan log. Setelah operasi selesai, statement st ditutup untuk menghindari kebocoran memori.

1. Untuk menghitung total Order Detail, buat kode berikut.

Sebuah gambar berisi teks, cuplikan layar

Deskripsi dibuat secara otomatis

Kode ini adalah metode yang digunakan untuk menghitung total pesanan berdasarkan id\_order. Pertama, kode ini menyiapkan statement SQL untuk menjalankan query yang menghitung total, kemudian menetapkan parameter id\_order pada query tersebut. Setelah query dijalankan, hasilnya diambil menggunakan ResultSet, dan jika ada data, nilai total diperoleh dari kolom pertama hasil query. Jika terjadi kesalahan saat eksekusi, pesan kesalahan akan dicatat menggunakan logger. Terakhir, kode ini menutup PreparedStatement dan ResultSet untuk menghindari kebocoran sumber daya, sebelum mengembalikan nilai total.

1. Fungsi menampilkan Order Detail, tambahkan kode program berikut.

Sebuah gambar berisi teks, cuplikan layar, Font

Deskripsi dibuat secara otomatis

Selanjutnya panggil method pada class main seperti kode berikut.

Sebuah gambar berisi teks, cuplikan layar, Font

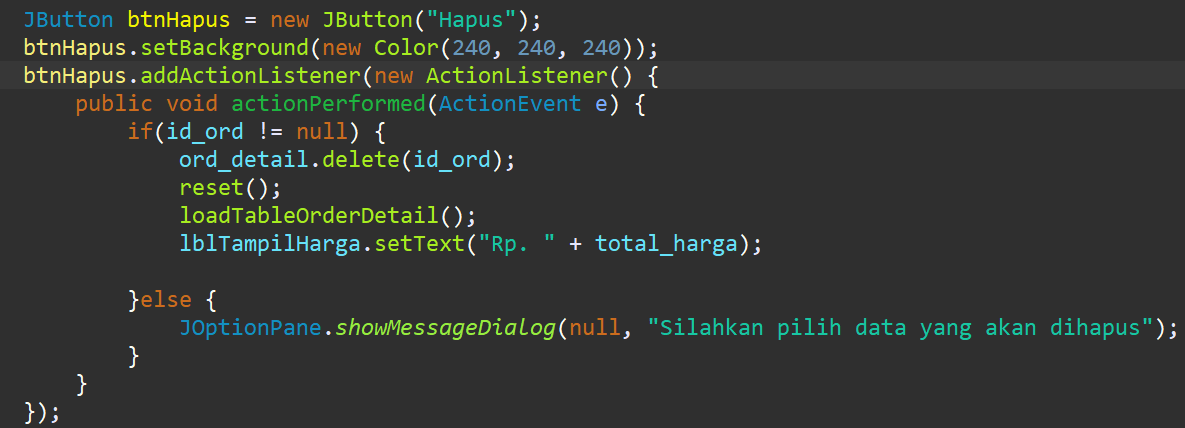
Deskripsi dibuat secara otomatis

1. Tambahkan event handler pada btnUbah seperti kode program berikut.Sebuah gambar berisi teks, cuplikan layar, Font

   Deskripsi dibuat secara otomatis

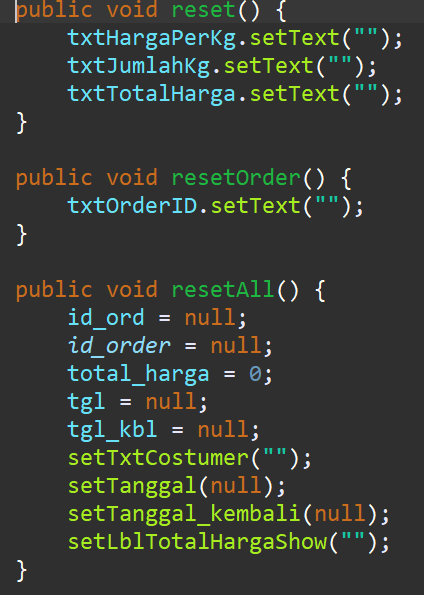
Kode ini mendefinisikan sebuah tombol btnUbah yang, ketika diklik, akan memeriksa apakah id\_ord tidak null. Jika tidak null, kode akan membuat objek OrderDetail, mengisi data yang diperlukan (seperti id\_order, id\_layanan, jumlah, total harga, dan id\_order\_detail) menggunakan nilai dari komponen antarmuka pengguna (UI). Kemudian, data tersebut akan diperbarui melalui metode update pada objek ord\_detail, diikuti dengan pemanggilan fungsi reset() untuk membersihkan form, dan memuat ulang data pada tabel OrderDetail serta menampilkan total harga terbaru pada label lblTampilHarga. Jika id\_ord null, maka akan muncul pesan peringatan yang meminta pengguna untuk memilih data yang akan diedit.

1. Tambahkan kode program berikut pada event Handler btnHapus.



Kode ini membuat sebuah tombol JButton dengan label "Hapus", yang diberi warna latar belakang abu-abu muda. Ketika tombol ini diklik, sebuah ActionListener akan dijalankan. Di dalamnya, jika variabel id\_ord tidak bernilai null, maka data dengan id\_ord tersebut akan dihapus melalui method delete dari objek ord\_detail, kemudian tampilan akan direset dengan memanggil method reset(), tabel detail pesanan akan dimuat ulang menggunakan loadTableOrderDetail(), dan label harga total akan diperbarui dengan nilai total\_harga. Jika id\_ord bernilai null, sebuah pesan peringatan akan muncul, memberitahukan pengguna untuk memilih data yang akan dihapus.

1. Tambahkan kode program berikut pada event handler btnBatal.



Kode ini berisi tiga metode untuk mereset nilai-nilai dalam form atau variabel yang ada. reset() mengosongkan input teks untuk harga per kilogram, jumlah kilogram, dan total harga. resetOrder() mengosongkan ID pesanan (txtOrderID). Sedangkan resetAll() mereset beberapa variabel seperti ID pesanan, total harga, dan tanggal, serta mengatur ulang teks yang ditampilkan pada beberapa elemen seperti label total harga, nama pelanggan, dan tanggal. Tujuan dari metode-metode ini adalah untuk mengatur ulang nilai-nilai form dan variabel agar siap digunakan untuk data baru.

1. Sampai pada tahapan ini, kita sudah dapat melakukan proses menambah detail order, mengubah, menghapus dan membatalkannya berdasarkan dengan order id, berikut ini tampilan aplikasi saat melakukan proses CRUD data. Tampilannya yakni sebagai berikut.

Sebuah gambar berisi teks, elektronik, cuplikan layar, tampilan

Deskripsi dibuat secara otomatis

Ketika pertama kali OrderDetailFrame dijalankan maka form Order ID akan terisi secara otomatis, ini kedepannya akan dibuatkan generate Order ID secara otomatis, akan tetapi pada praktikum kali ini masih manual, selanjutnya akan tampilan daftar layanan yang tersedia jika dipilih salah satu layanan maka form harga akan terisi otomatis. Pada gambar di atas kita memilih layanan cuci setrika express 1 hari, maka secara otomatis form harga terisi dengan harga layanan yaitu Rp8.000/kg, selanjutnya operator akan menginputkan jumlah cuciannya ke dalam form jumlah dan secara otomatis akan menghitung totalnya yaitu harga x jumlah dengan memanggil method total yang sudah dibuat. Diinputkan jumlah kgnya yaitu 2 maka secara otomatis totalnya menjadi 16.000, selanjutnya klik tombol simpan untuk menyimpan. Maka, data akan tersimpan di tabel orderdetail di bawah. Jika ingin mengubah atau menghapus data dapat dilakukan dengan cara klik data order kemudian secara otomatis form harga, jumlah dan total akan terisi dengan data-data sesuai dengan yang dipilih, selanjutnya klik tombol ubah untuk mengubah data atau klik tombol hapus untuk menghapus data.

1. Langkah selanjutnya setelah berhasil menambahkan data order detail maka selanjutnya membuat pesanan, Langkah pertama adalah memilih pelanggan, menginputkan tanggal, tanggal pengambilan, memilih status, metode pembayaran dan status pembayaran menggunakan JDialog. JDialog berfungsi untuk menampilkan nama-nama pelanggan yang ada pada aplikasi, Ketika operator klik form pelanggan maka akan ditampilkan daftar pelanggan, selanjutnya operator memilih pelanggan maka secara otomatis fomr pelanggan akan terisi oleh data pelanggan tersebut. Ada 2 buah parameter yang dikirimkan oleh JDialog saat memilih pelanggan yaitu id dan nama, id merupakan id pelanggan yang nantinya akan disimpan sementara pada variable id\_pelanggan, kemudian nama akan ditampilkan pada form nama, berikut ini langkah-langkahnya.
2. Tambahkan JDialog pada package ui dengan nama DialogPelanggan.java, buat desain UI seperti gambar berikut.

Sebuah gambar berisi teks, cuplikan layar, tampilan, software

Deskripsi dibuat secara otomatis

1. Tambahkan kode program berikut pada DialogPelanggan.java.

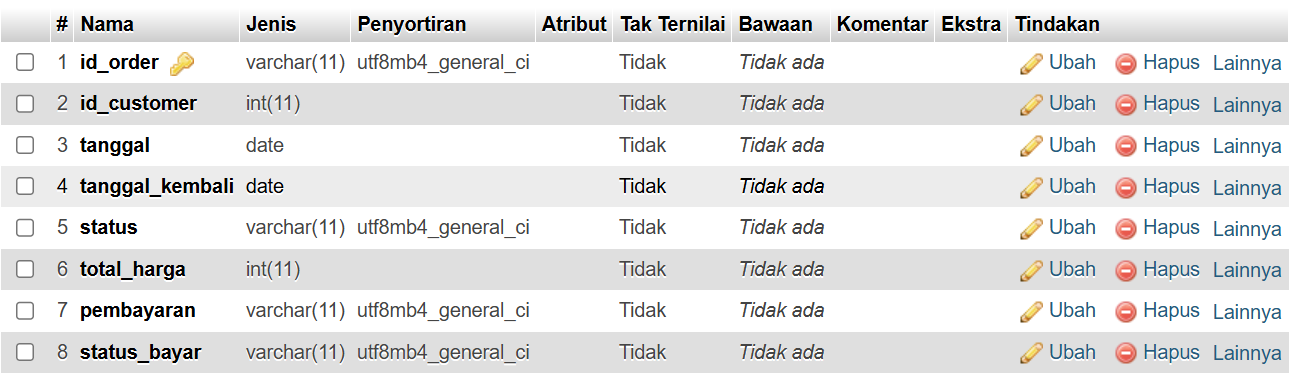
Sebuah gambar berisi teks, elektronik, cuplikan layar, tampilan

Deskripsi dibuat secara otomatis

1. Untuk menampilkan JDialog Ketika form pelanggan diklik, tambahkan event handler berikut pada txtPelanggan.



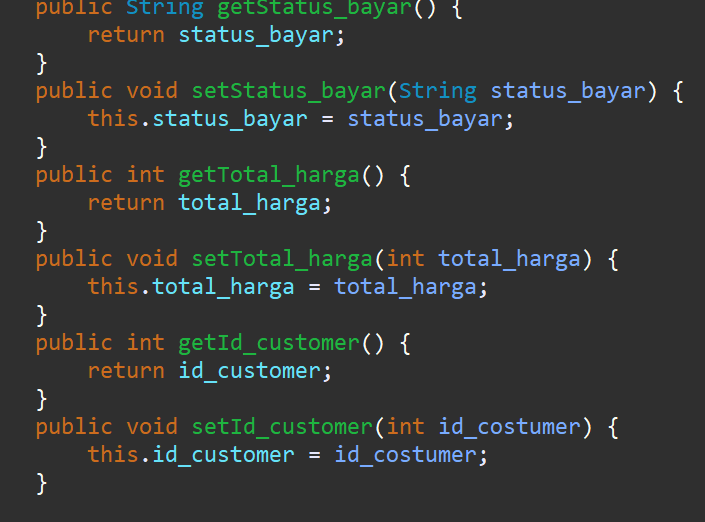
1. Setelah proses CRUD order detail selesai, selanjutnya menyimpan data order dengan tahapan berikut.
2. Buat table baru dengan nama tabel\_order dengan atribut seperti berikut.



1. Tambahkan model baru pada package model dengan nama Order dan tambahkan kode program berikut.

Sebuah gambar berisi teks, cuplikan layar, software

Deskripsi dibuat secara otomatis



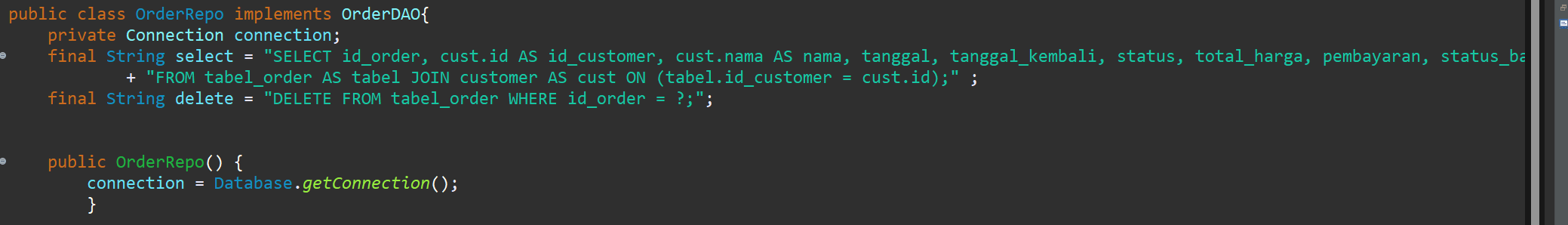
1. Tambahkan DAO pada package dao dengan nama OrderDao dan tambahkan kode program berikut.

Sebuah gambar berisi teks, cuplikan layar, Font, nomor

Deskripsi dibuat secara otomatis

Kode ini adalah definisi dari sebuah interface OrderDAO yang berfungsi sebagai kontrak untuk mengelola data pesanan (Order). Interface ini memiliki tiga metode: show() untuk menampilkan daftar pesanan, delete(String id) untuk menghapus pesanan berdasarkan ID, dan save(Order order) untuk menyimpan data pesanan baru. Setiap metode ini akan diimplementasikan di kelas yang mengimplementasikan OrderDAO untuk melakukan operasi yang sesuai terhadap data pesanan.

1. Tambahkan OrderRepo pada package DAO dan tambahkan kode program berikut ini.



1. Untuk Menambah Data Order, tambahkan kode program berikut ini.

Sebuah gambar berisi teks, cuplikan layar, software

Deskripsi dibuat secara otomatis

Kode ini berfungsi untuk menyimpan atau memperbarui data pesanan (Order) ke dalam database. Kode menggunakan SQL query dengan perintah INSERT INTO untuk menambah data baru pada tabel tabel\_order, dan menggunakan ON DUPLICATE KEY UPDATE untuk memperbarui data yang sudah ada jika id\_order sudah ada di database. Pertama, objek PreparedStatement digunakan untuk menyiapkan query yang akan dieksekusi. Kemudian, data dari objek Order dimasukkan ke dalam query melalui metode setString dan setInt. Setelah query disiapkan, perintah st.executeUpdate() digunakan untuk mengeksekusi query tersebut. Jika terjadi error, pengecualian (SQLException) akan ditangani, dan setelah eksekusi selesai, koneksi PreparedStatement ditutup di bagian finally untuk memastikan sumber daya dibebaskan.

1. Untuk menampilkan Data Order, tambahkan kode program berikut ini.

Sebuah gambar berisi teks, cuplikan layar, software, Font

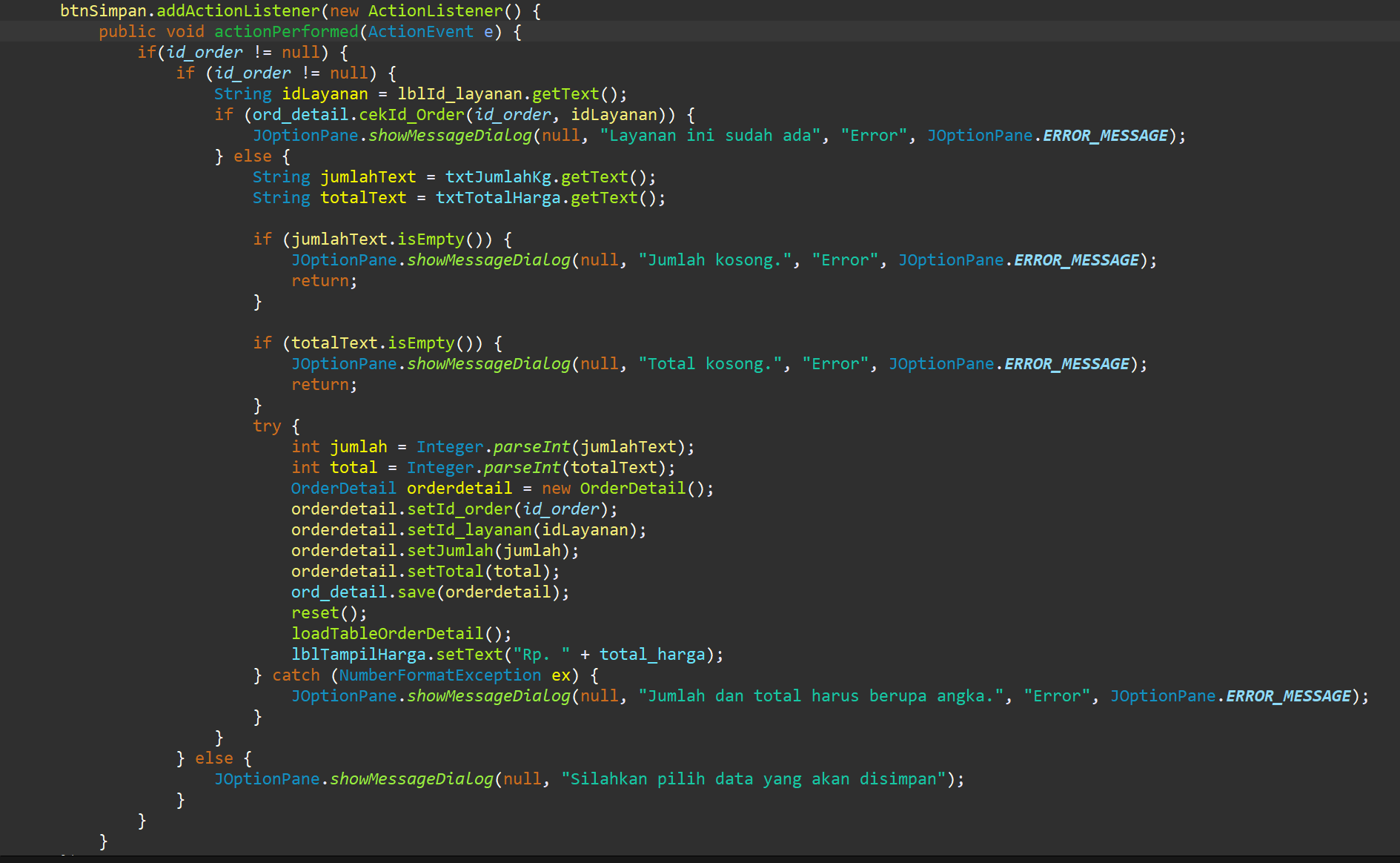
Deskripsi dibuat secara otomatis

1. Untuk menghapus Data Order, tambahkan kode program berikut ini.

Sebuah gambar berisi teks, cuplikan layar, Font

Deskripsi dibuat secara otomatis

1. Sebelum menyimpan data Order maka harus dipastikan terlebih dahulu data pelanggan, tanggal, tanggal pengambilan, status, pembayaran dan status pembayaran baru diklik tombol simpan. Tambahkan kode program berikut ini pada tombol simpan.

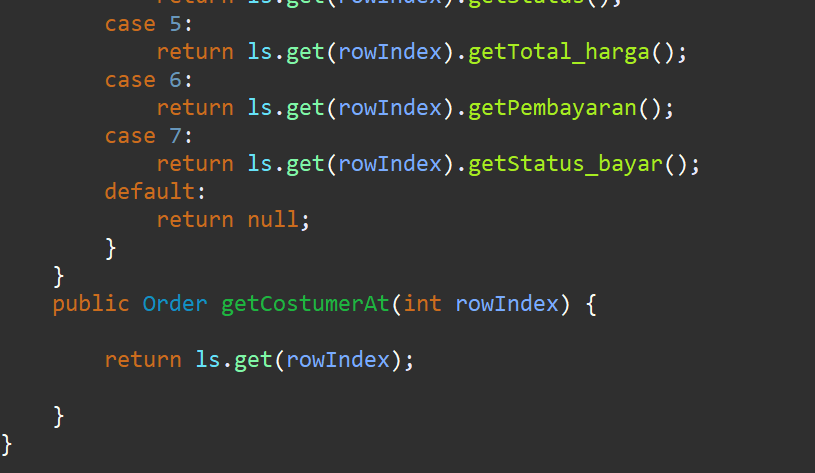


Kode tersebut adalah bagian dari event handler untuk tombol "Simpan". Ketika tombol diklik, pertama-tama kode memeriksa apakah id\_order tidak kosong. Jika ada, kode akan memeriksa apakah layanan dengan id\_order dan idLayanan sudah ada di database menggunakan metode cekId\_Order. Jika sudah ada, akan muncul pesan error. Jika belum ada, kode akan memeriksa apakah input jumlah dan total harga sudah diisi. Jika ada yang kosong, akan muncul pesan error. Setelah itu, kode mencoba mengonversi input jumlah dan total harga menjadi angka dan membuat objek OrderDetail untuk disimpan ke dalam database. Setelah berhasil menyimpan, form akan direset, tabel detail pesanan diperbarui, dan harga total ditampilkan. Jika terjadi kesalahan dalam konversi angka, pesan error akan muncul.

1. Setelah data order berhasil disimpan, selanjutnya tampilkan data order dengan buatkan terlebih dahulu class untuk table order, tambahkan class baru pada package table dengan nama TableOrder tambahkan kode program berikut.

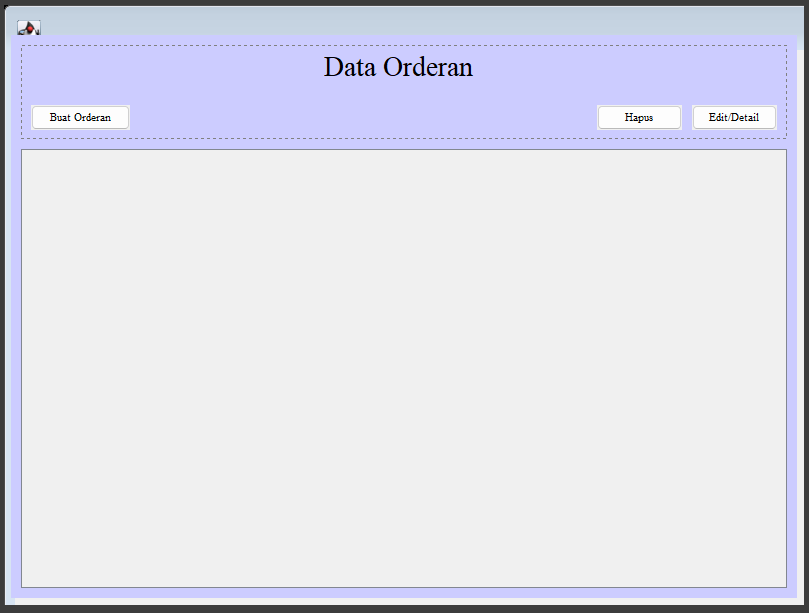
Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Deskripsi dibuat secara otomatis

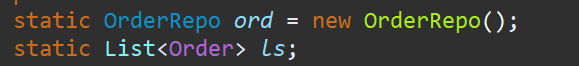


Kode ini adalah implementasi dari kelas TableOrder yang mengextends AbstractTableModel untuk menampilkan data pesanan (order) dalam bentuk tabel di aplikasi berbasis Java. Kelas ini menerima daftar objek Order dan mengatur bagaimana data tersebut ditampilkan dalam tabel. Metode getRowCount() mengembalikan jumlah baris berdasarkan banyaknya data dalam daftar ls, sementara getColumnCount() menentukan jumlah kolom dalam tabel. Metode getColumnName() memberikan nama kolom sesuai dengan urutan pada array columnNames. Untuk mendapatkan nilai pada suatu sel, metode getValueAt() digunakan, yang akan mengembalikan nilai sesuai dengan indeks baris dan kolom. Terakhir, metode getCostumerAt() memungkinkan akses ke objek Order berdasarkan baris yang dipilih dalam tabel.

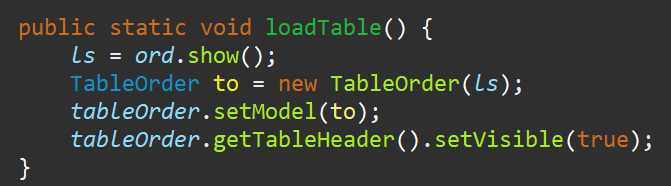
1. Kemudian, tambahkan JFrame baru dengan nama OrderFrame, buat desainnya seperti gambar berikut.



1. Selanjutnya tambahkan kode program berikut pada OrderFrame.



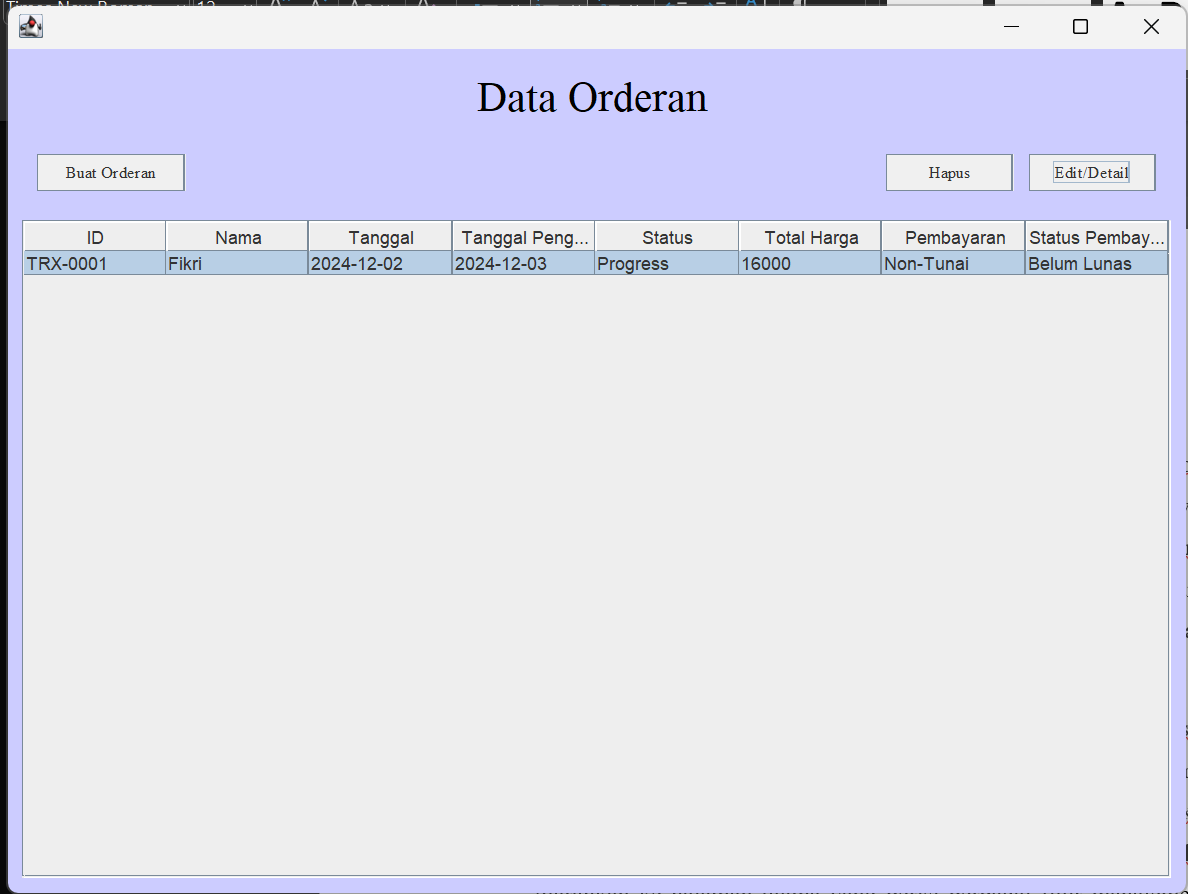
1. Buat method untuk menampilkan data order dan Panggil method loadTableOrder() pada main frame.



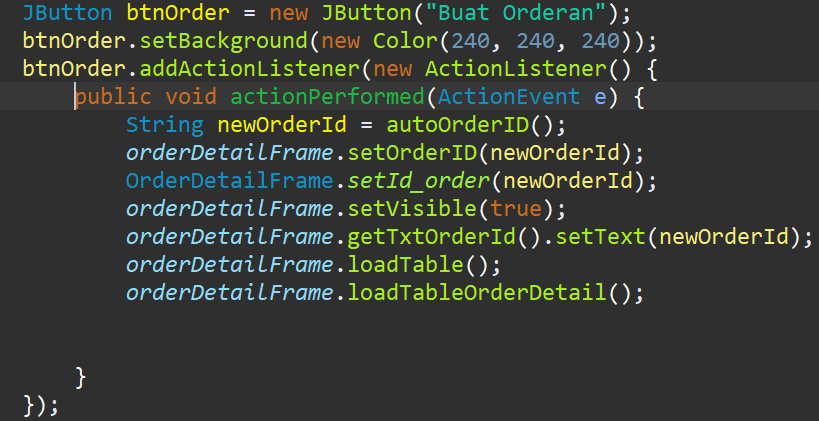
Sebuah gambar berisi teks, cuplikan layar, Font

Deskripsi dibuat secara otomatis

1. Tampilan data order akan seperti berikut jika program dieksekusi.



1. Tambahkan kode program berikut pada event handler Buat Orderan sehingga akan tampil form OrderDetailFrame dan dapat menambahkan data order.



1. Tambahkan kode program berikut pada event handler pada tombol hapus.

Sebuah gambar berisi teks, cuplikan layar

Deskripsi dibuat secara otomatis

Untuk menghapus data, pilih data order dan klik tombol hapus.

1. Membuat Method Tabel Order untuk setiap row yang diklik.

Sebuah gambar berisi teks, cuplikan layar, Font

Deskripsi dibuat secara otomatis

1. Method Generate Custom ID untuk id\_order.

Sebuah gambar berisi teks, cuplikan layar, Font

Deskripsi dibuat secara otomatis  
Kelas OrderFrame adalah bagian dari antarmuka pengguna aplikasi berbasis Java Swing yang bertujuan untuk mengelola dan menampilkan data pesanan. Kelas ini menginisialisasi berbagai komponen GUI, termasuk panel, tabel, dan tombol untuk melakukan berbagai operasi terkait pesanan. Dalam konstruktor OrderFrame(), jendela diatur dengan ukuran dan tata letak yang sesuai, dan panel untuk menampilkan informasi pesanan serta tombol untuk membuat, mengedit, dan menghapus pesanan ditambahkan. Fungsi utama dari kelas ini termasuk memuat data pesanan dari repositori menggunakan metode loadTable(), yang mengambil daftar pesanan dari OrderRepo dan menampilkan hasilnya dalam JTable. Kelas ini juga memiliki metode generateOrderID(), yang menghasilkan ID pesanan baru dengan format tertentu berdasarkan ID pesanan yang sudah ada. Pengguna dapat berinteraksi dengan tabel pesanan; ketika baris dipilih, informasi terkait pesanan tersebut disimpan dalam variabel untuk digunakan saat mengedit atau menghapus pesanan. Tombol "Buat Orderan" membuka jendela OrderDetailFrame untuk membuat pesanan baru, sedangkan tombol "Edit/Detail" dan "Hapus" memungkinkan pengguna untuk mengedit atau menghapus pesanan yang dipilih. Secara keseluruhan, OrderFrame menyediakan antarmuka yang intuitif untuk mengelola data pesanan dalam aplikasi.

1. Pada order frame, jika diklik edit orderan, tampilan GUI yang awalnya begini:

Sebuah gambar berisi teks, cuplikan layar, tampilan, software

Deskripsi dibuat secara otomatis

Jadi seperti ini, menampikan data-data orderan seseuai dengan data yang dipilih dan dapat dilakukan fungsi untuk edit orderan.

Sebuah gambar berisi teks, elektronik, cuplikan layar, tampilan

Deskripsi dibuat secara otomatis

1. Ini merupakan program penerapan JCalendar di tanggal pemesanan dan tanggal kembalian.

Sebuah gambar berisi teks, cuplikan layar, software, Font

Deskripsi dibuat secara otomatis

1. **SIMPULAN**

Praktikum ini mempelajari pembuatan aplikasi pemesanan laundry dengan menggunakan konsep Object-Oriented Programming (OOP) dan teknologi berbasis Java, khususnya menggunakan XAMPP sebagai server lokal dan MySQL untuk pengelolaan basis data. Praktikum ini berhasil mencapai beberapa tujuan, seperti pembuatan tampilan antarmuka pengguna (UI) untuk melakukan pemesanan laundry, implementasi fungsi untuk menambah, mengubah, dan menghapus pesanan, serta pengelolaan data pemesanan laundry dengan menggunakan CRUD (Create, Read, Update, Delete).

Proses pembuatan aplikasi dimulai dengan desain tampilan menggunakan WindowBuilder untuk menciptakan antarmuka yang memungkinkan pengguna melakukan pemesanan laundry secara mudah. Fungsi pemesanan termasuk pengisian data layanan laundry, penghitungan harga per kilogram secara otomatis, dan kalkulasi total harga berdasarkan jumlah yang dipesan. Data yang dimasukkan kemudian disimpan dalam basis data MySQL menggunakan DAO (Data Access Object) yang memisahkan logika akses data dengan logika bisnis aplikasi, sehingga mempermudah pengelolaan dan pemeliharaan aplikasi.

Selain itu, praktikum ini juga melibatkan pengelolaan pemesanan laundry dengan menggunakan JDialog untuk memilih pelanggan, serta proses penyimpanan data pemesanan dalam tabel yang sudah terstruktur dengan baik di basis data. Keseluruhan proses CRUD ini berhasil diterapkan dengan baik, di mana mahasiswa dapat menambah, mengubah, menghapus, dan membatalkan pesanan berdasarkan ID yang sudah ada.

Secara keseluruhan, praktikum ini memberikan pemahaman yang lebih dalam tentang pembuatan aplikasi berbasis web dengan menggunakan Java dan MySQL, serta penerapan konsep-konsep penting dalam pemrograman berorientasi objek., pengelolaan data yang terstruktur dalam basis data dan penerapan desain antarmuka yang user-friendly untuk mempermudah operasional aplikasi.