LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

PERTEMUAN 5



OLEH:

IRGI FATIHUL IHSAN

NIM: 2311533010

DOSEN PENGAMPU:

NURFIAH, S.ST, M.KOM

FAKULTAS TEKNOLOGI INFORMASI

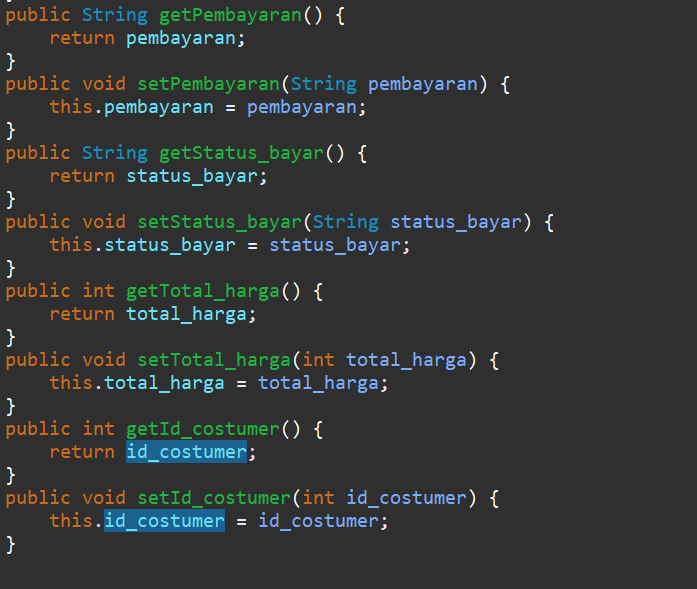
DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, SEPTEMBER 2024

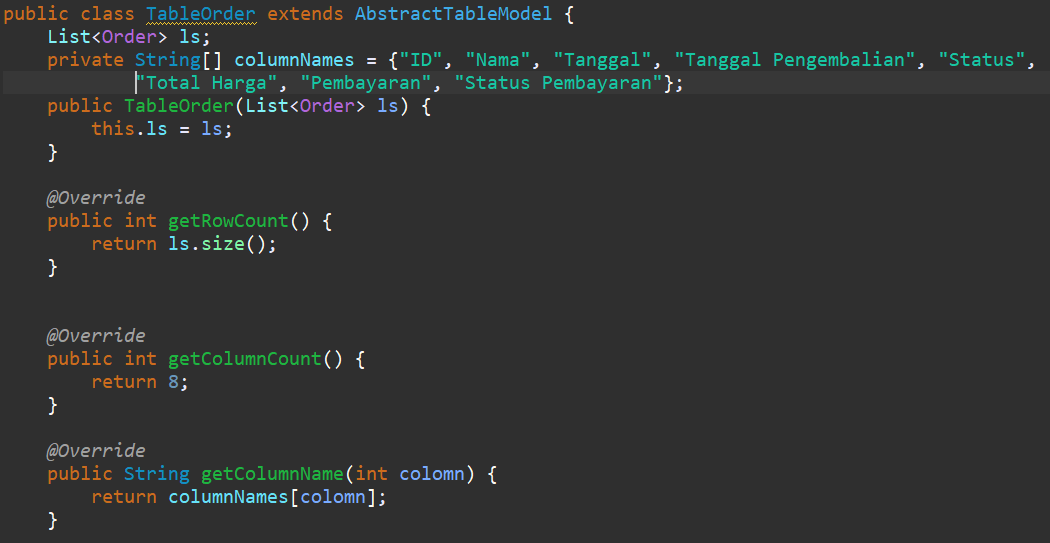
**Membuat model order**

****

****

Terdapat 9 entity yang akan digunakan pada order, yaitu id\_order, nama, tanggal, tanggal\_kembali, status, pembayaran, status\_pembayaran, total\_harga, dan id\_costumer. Semua entity dibuat setter dan getternya dalam file model order tersebut.

**Membuat Tabel Order**

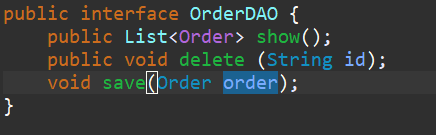
****

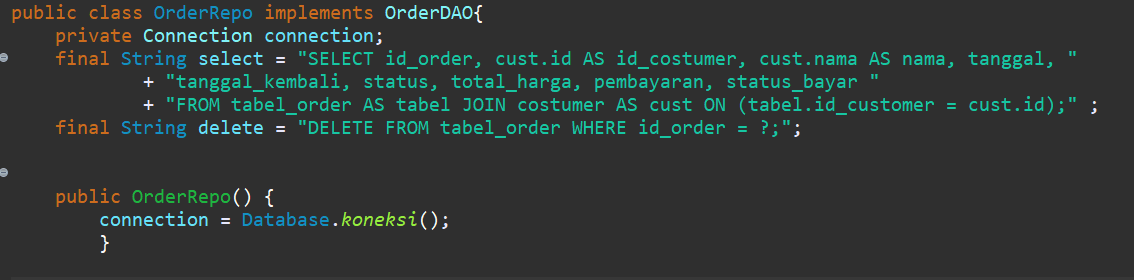
****

Terdapat beberapa Method pada class ini, yaitu:

1. **Konstruktor TableOrder(List<Order> ls)**:
   * Konstruktor ini digunakan untuk menginisialisasi objek TableOrder dengan daftar pesanan (ls) yang diberikan sebagai parameter. Daftar ini akan digunakan untuk mengisi data dalam tabel.
2. **@Override public int getRowCount()**:
   * Method ini mengembalikan jumlah baris yang ada pada tabel. Dalam hal ini, jumlah baris sama dengan ukuran daftar ls, yaitu jumlah objek Order yang ada.
3. **@Override public int getColumnCount()**:
   * Method ini mengembalikan jumlah kolom yang ada pada tabel. Dalam contoh ini, jumlah kolom yang ditentukan adalah 8, sesuai dengan jumlah elemen dalam array columnNames.
4. **@Override public String getColumnName(int column)**:
   * Method ini mengembalikan nama kolom untuk kolom tertentu berdasarkan indeks yang diberikan. Nama kolom diambil dari array columnNames, yang berisi nama-nama kolom yang akan ditampilkan di header tabel.
5. **@Override public Object getValueAt(int rowIndex, int columnIndex)**:
   * Method ini mengembalikan nilai yang sesuai dengan baris dan kolom tertentu dalam tabel. Dengan menggunakan switch, method ini memeriksa columnIndex dan mengembalikan nilai dari objek Order yang sesuai dengan rowIndex dan kolom yang diminta. Jika columnIndex tidak valid, method ini akan mengembalikan null.
6. **public Order getCostumerAt(int rowIndex)**:
   * Method ini mengembalikan objek Order yang sesuai dengan rowIndex yang diberikan. Ini berguna untuk mendapatkan akses langsung ke objek Order yang mendasari data tabel, sehingga pengguna dapat melakukan operasi lebih lanjut pada objek tersebut jika diperlukan.

**Membuat OrderDAO dan OrderRepo**



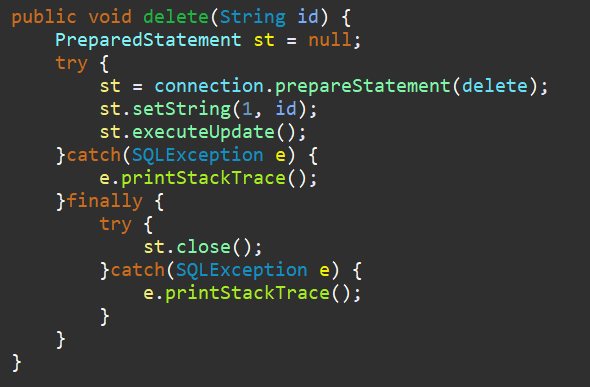


Ini merupakan sebuah query untuk menampilkan tabel order dan customer yang dijoin-kan menjadi satu query select. Fungsi dari join adalah menggabungkan tabel costumer dengan digabungkannya id\_customer dari tabel order dengan id dari tabel customer sehingga dapat menampilkan nama dari tabel customer di tabel\_order.

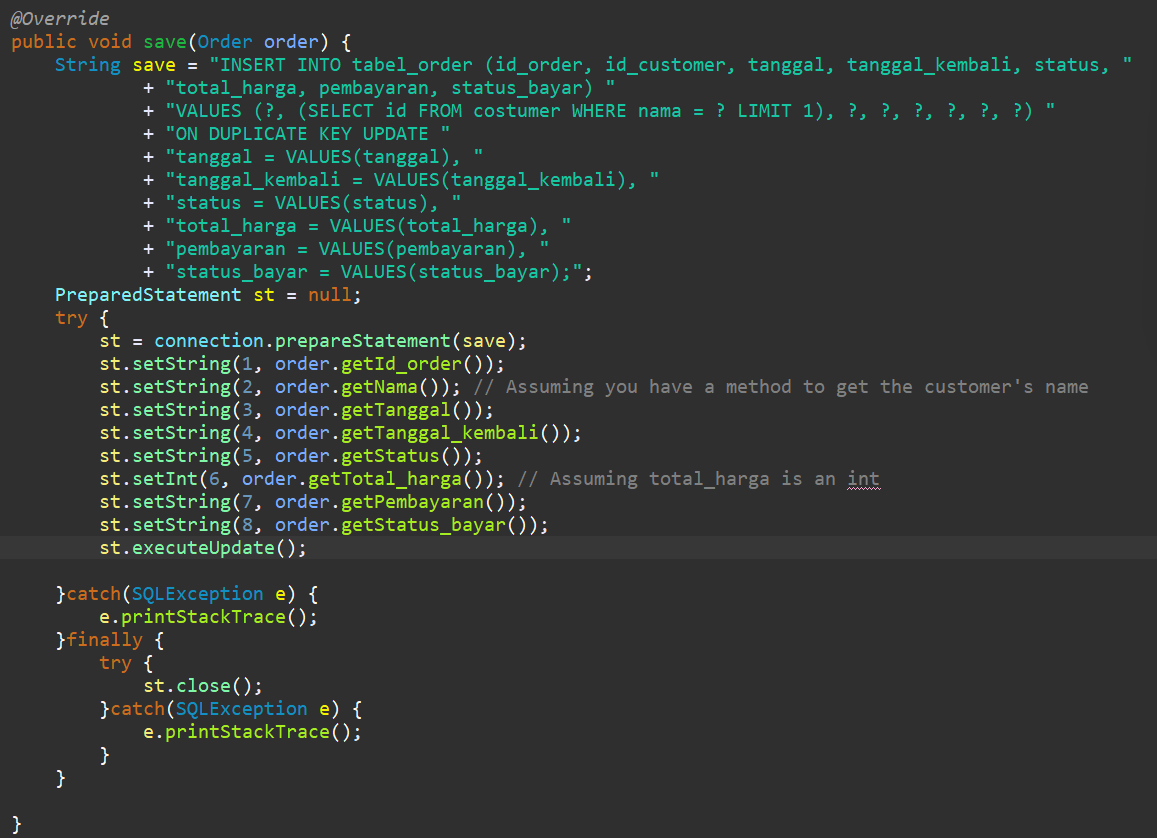
Untuk query delete, fungsinya yaitu menghapus row.



Metode show() dalam kelas OrderRepo berfungsi untuk mengambil semua data pesanan dari database dan mengembalikannya dalam bentuk daftar objek Order. Pertama, metode ini menginisialisasi sebuah daftar kosong (ArrayList<Order>) untuk menyimpan objek Order. Kemudian, dengan menggunakan Statement, metode ini menjalankan query SQL yang telah didefinisikan untuk memilih data dari tabel tabel\_order dan costumer. Hasil dari query disimpan dalam ResultSet, yang memungkinkan iterasi melalui setiap baris hasil. Dalam loop, untuk setiap baris yang ditemukan, objek Order baru dibuat dan atribut-atributnya diisi dengan data yang diambil dari kolom yang sesuai dalam ResultSet. Setelah semua atribut diatur, objek Order tersebut ditambahkan ke dalam daftar. Jika terjadi kesalahan saat menjalankan query atau memproses hasil, metode ini menangkap SQLException dan mencatat kesalahan tersebut menggunakan Logger. Setelah seluruh proses selesai, metode ini mengembalikan daftar yang berisi semua objek Order, memungkinkan pengguna untuk mengakses semua data pesanan yang ada di database.



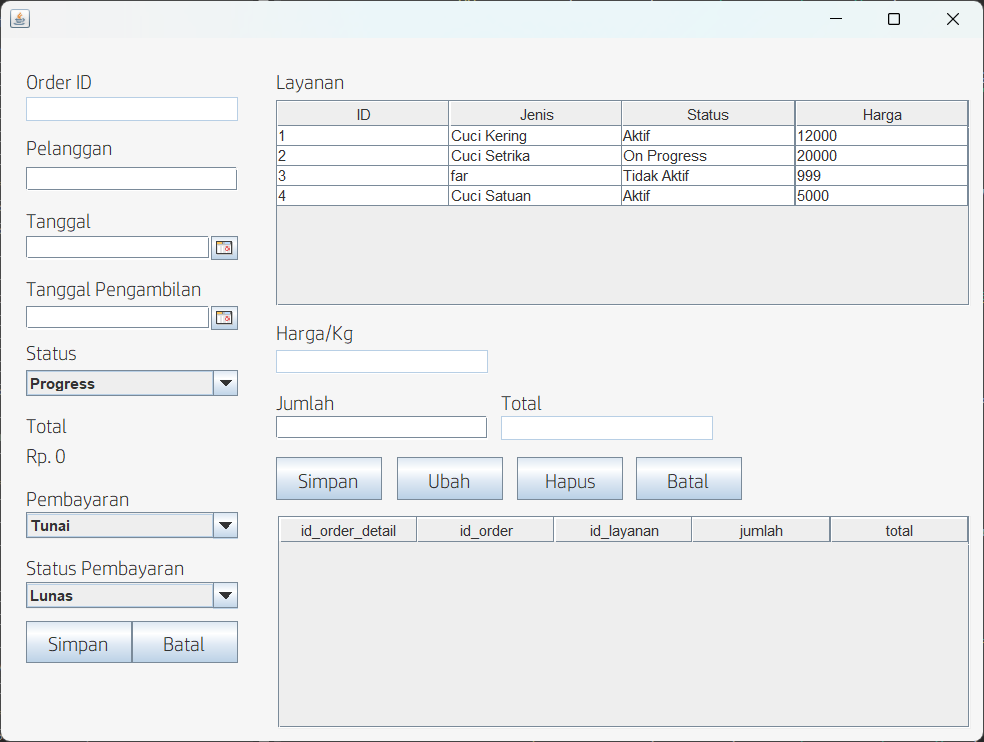
Metode delete(String id) dalam kelas OrderRepo bertujuan untuk menghapus entri pesanan dari database berdasarkan id\_order yang diberikan. Pertama, metode ini mendeklarasikan sebuah variabel PreparedStatement yang akan digunakan untuk mengeksekusi query SQL yang aman dari SQL injection. Di dalam blok try, objek PreparedStatement dibuat dengan menggunakan query DELETE yang telah didefinisikan sebelumnya, di mana id\_order yang ingin dihapus akan diisi sebagai parameter. Metode ini kemudian menggunakan st.setString(1, id); untuk menetapkan nilai parameter pertama dari query dengan id yang diterima oleh metode. Setelah itu, st.executeUpdate(); dipanggil untuk menjalankan query dan menghapus data yang sesuai dari tabel tabel\_order. Jika terjadi kesalahan selama proses ini, pengecualian SQLException ditangkap dan dicetak ke konsol untuk membantu dalam debugging. Di bagian finally, metode ini memastikan bahwa PreparedStatement ditutup dengan memanggil st.close();, sehingga sumber daya dibebaskan dengan baik, terlepas dari apakah operasi berhasil atau gagal. Dengan demikian, metode ini memberikan cara yang efisien dan aman untuk menghapus data pesanan dari database berdasarkan id\_order yang spesifik.



Metode save(Order order) dalam kelas OrderRepo berfungsi untuk menyimpan objek Order baru ke dalam database. Proses ini dimulai dengan mendeklarasikan sebuah PreparedStatement di dalam blok try, yang memungkinkan eksekusi query SQL dengan aman dan efisien. Query SQL yang digunakan adalah perintah INSERT yang dirancang untuk menambahkan data ke dalam tabel tabel\_order, di mana parameter-parameter untuk kolom yang relevan diisi menggunakan nilai-nilai dari objek Order. Setiap nilai diisi dengan metode seperti setString() dan setInt(), yang sesuai dengan tipe data kolom dalam tabel. Setelah semua parameter diatur, metode ini menjalankan query dengan memanggil executeUpdate(), yang mengeksekusi perintah dan menambahkan baris baru ke dalam tabel. Jika terjadi kesalahan, pengecualian SQLException akan ditangkap dan dicetak untuk membantu dalam debugging. Di bagian finally, PreparedStatement ditutup untuk membebaskan sumber daya. Dengan demikian, metode ini tidak hanya menyimpan data pesanan dengan benar, tetapi juga memastikan keamanan dan efisiensi dalam proses penyimpanan ke database.

**Update Frame Order Detail**

Tampilan:

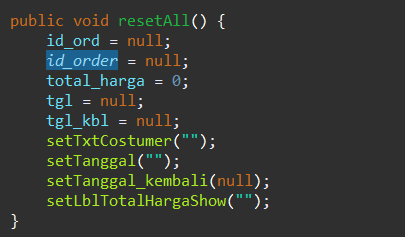


Kode:

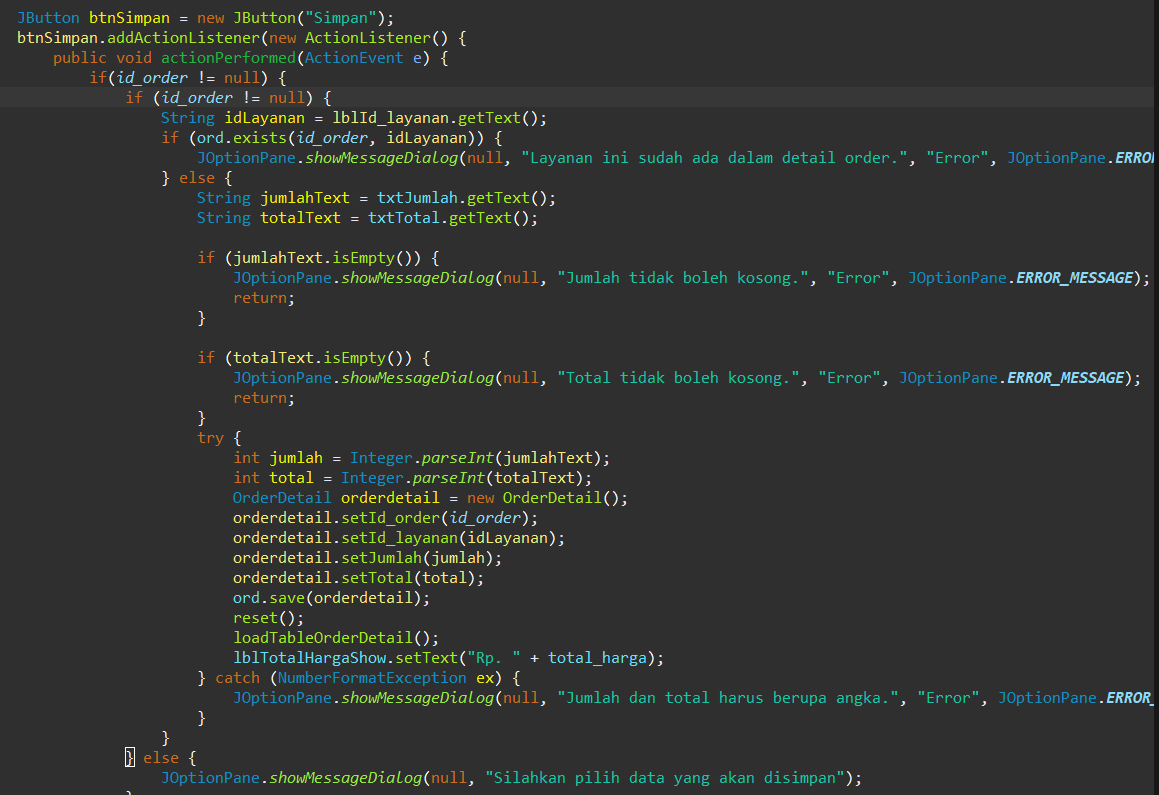


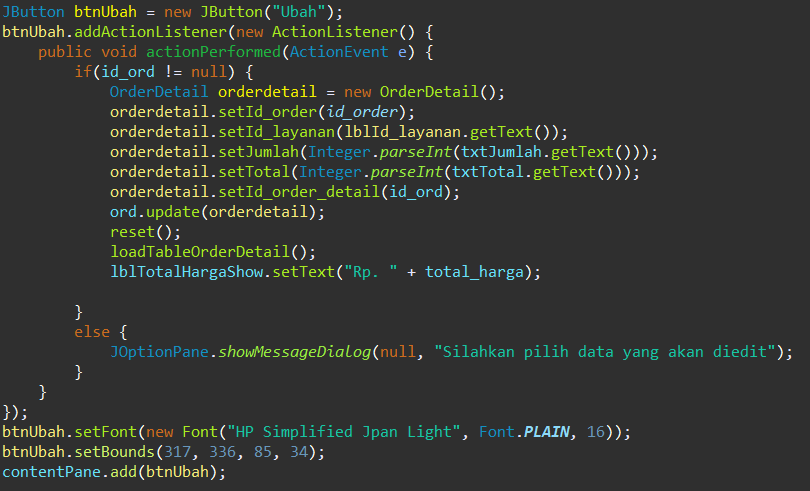


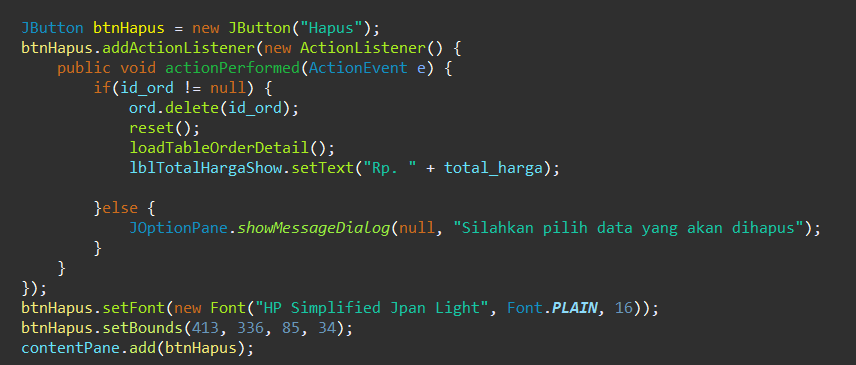


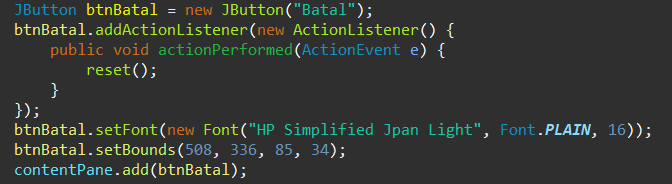


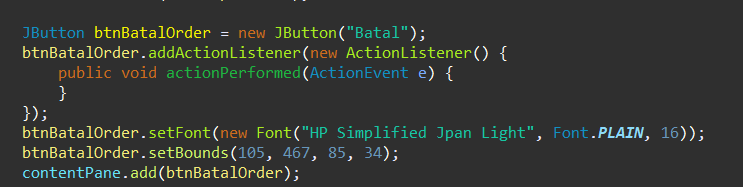
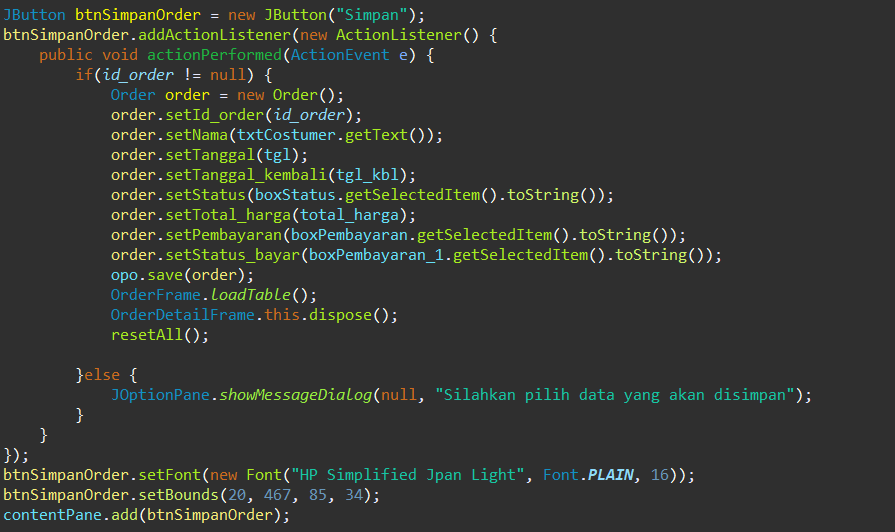
  

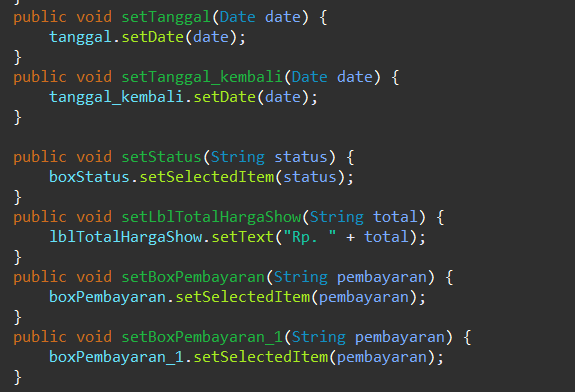









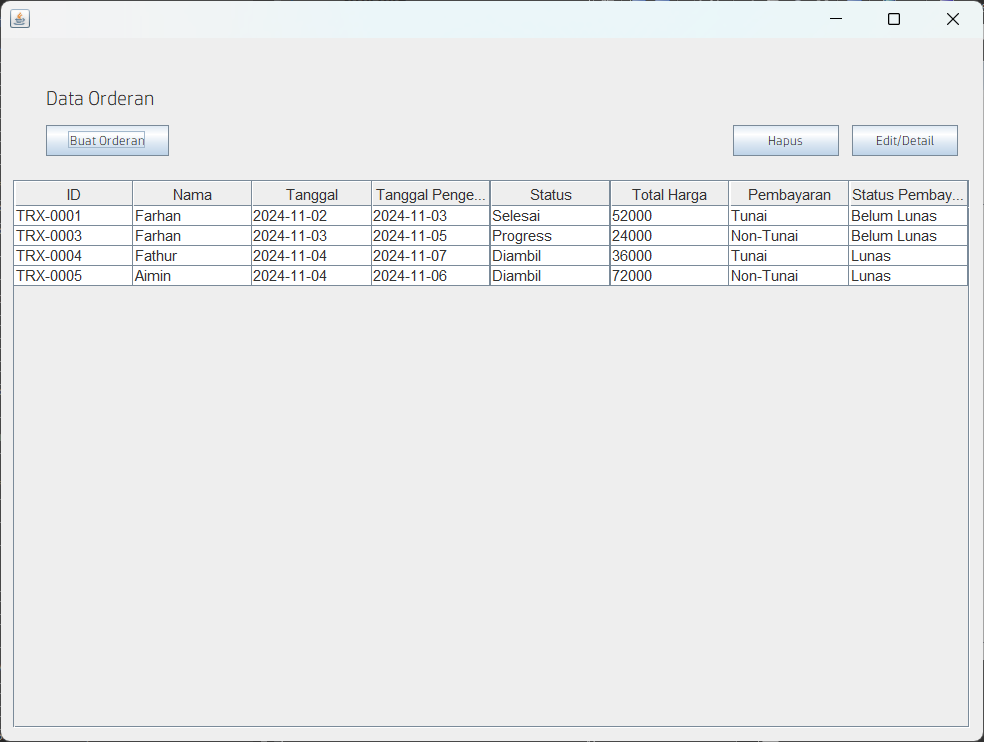




Kelas OrderDetailFrame merupakan implementasi antarmuka pengguna berbasis Java Swing yang dirancang untuk mengelola detail pesanan dalam aplikasi manajemen. Dalam konstruktor kelas ini, berbagai komponen GUI seperti JPanel, JTextField, JComboBox, JLabel, dan JTable diinisialisasi dan diatur untuk menciptakan tampilan yang terstruktur. Kelas ini menggunakan beberapa repositori, termasuk OrderRepo, ServiceRepo, dan OrderDetailRepo, untuk berinteraksi dengan data pesanan, layanan, dan detail pesanan. Metode utama seperti loadTable() dan loadTableOrderDetail() bertugas untuk memuat dan menampilkan data layanan serta detail pesanan berdasarkan id\_order. Selain itu, terdapat mekanisme event handling yang menangani interaksi pengguna, seperti memilih layanan dari tabel dan menghitung total harga secara otomatis saat jumlah diubah. Kelas ini juga menyediakan fungsi CRUD (Create, Read, Update, Delete) melalui tombol untuk menyimpan, mengubah, dan menghapus detail pesanan, serta membatalkan pengisian. Penggunaan JDateChooser memungkinkan pengguna untuk memilih tanggal dengan mudah, dan terdapat dialog untuk memilih pelanggan yang akan diisi di txtCostumer. Secara keseluruhan, OrderDetailFrame menyajikan antarmuka yang komprehensif untuk pengelolaan detail pesanan, memungkinkan pengguna untuk melakukan berbagai operasi dengan efisien.

**Membuat Frame Order**

Tampilan:

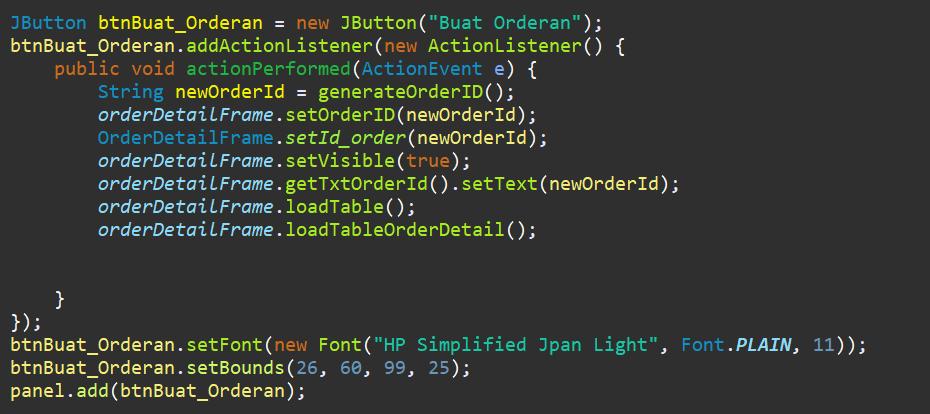


Kode:

Inisiasi variable



Tombol “Buat Orderan”



Tombol hapus



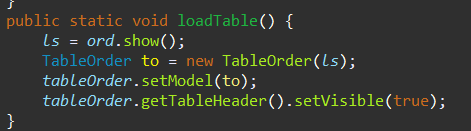
Tombol Edit



Method Tabel Order untuk setiap row yang di klik



Method Load Table



Method Generate Custom ID untuk id\_order



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.