

Pemrograman Berbasis Objek

Pembuatan Aplikasi Laundry Menggunakan Netbeans

(Disusun Untuk Memenuhi Tugas Akhir Mata Kuliah Pemrograman Berbasis Objek)



DOSEN

Dewi Andryani

DISUSUN OLEH

- | | |
|------------------------------------|------------|
| 1. Dimas Ajie Putra | (50421382) |
| 2. Eymir Nabil Makarim | (50421426) |
| 3. Ferdinand Andhika Widhiyan | (50421513) |
| 4. Fernando Michael Hebert Siregar | (50421515) |
| 5. Firman Nurcahyo | (50421524) |
| 6. Gede Putu Adyatma Putra | (50421545) |

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS GUNADARMA

2023

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan penulisan tugas akhir yang berjudul “Pembuatan Aplikasi Laundry Dengan Aplikasi Netbeans ” ini tepat pada waktunya. Adapun tujuan dari penulisan ini untuk memenuhi tugas akhir pada mata kuliah Pemrograman Berbasis Objek.

Kami mengucapkan terima kasih kepada Ibu Dewi Andryani selaku dosen mata kuliah Pemrograman Berbasis Objek yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan sesuai dengan bidang studi yang kami tekuni. Semoga penulisan ini dapat bermanfaat bagi kami dan para pembaca. Mohon maaf bila terdapat kekurangan dalam penyusunan laporan tugas akhir ini.

Jakarta, 11 Juni 2023

Kelompok 2

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI.....	3
BAB I PENDAHULUAN.....	5
1.1 Latar Belakang	5
1.2 Rumusan Masalah.....	5
1.3 Tujuan	6
1.4 Ruang Lingkup.....	6
BAB II TINJAUAN PUSTAKA	7
1. UML (Unified Modelling Language)	7
1.1. Definisi UML.....	7
1.2. Diagram UML.....	7
2. PHP	11
3. Database.....	11
4. XAMPP.....	12
BAB III PEMBAHASAN.....	13
1. Gambaran Aplikasi	13
1.1. Halaman Login.....	13
1.2. Halaman Home	13
1.3. Halaman Pemesanan	14
1.4. Halaman Pengambilan	14
1.5. Tampilan Selesai Pembayaran	15
1.6. Halaman Report (Laporan)	15
2. Analisis Kebutuhan.....	16
2.1. Data Yang Dibutuhkan	16
2.2. Kebutuhan Fungsional	16
2.3 Kebutuhan Non Fungsional.....	16
3. Rancangan Aplikasi	17

3.1. Use Case Diagram.....	17
3.2. Activity Diagram	18
4. Rancangan Database	22
5. Pembuatan Aplikasi	24
5.1. Listing Home	24
5.2. Listing Login.....	27
5.3. Listing Pembayaran	31
5.4. Listing Pemesanan	39
5.5. Output Halaman Login	48
5.6. Output Halaman Home	49
5.7. Output Halaman Pemesanan	49
5.8. Output Halaman Pengembalian	51
5.9. Output Halaman Laporan (Report)	51
BAB IV PENUTUP	52
1. KESIMPULAN.....	52
2. SARAN.....	52
DAFTAR PUSTAKA	54

BAB I

PENDAHULUAN

1.1 Latar Belakang

Industri laundry merupakan salah satu industri yang terus berkembang seiring dengan perkembangan gaya hidup modern dan peningkatan kesibukan masyarakat. Namun, proses tradisional laundry yang melibatkan pencucian manual dan penjemuran pakaian dapat menjadi waktu yang memakan dan kurang efisien bagi banyak orang. Untuk mengatasi tantangan ini, penggunaan aplikasi mobile untuk layanan laundry telah menjadi tren populer dalam beberapa tahun terakhir.

Pengembangan aplikasi laundry modern bertujuan untuk meningkatkan efisiensi dalam layanan laundry, memungkinkan pengguna untuk melakukan pemesanan secara online, melacak status cucian, dan mengoptimalkan proses pengantaran dan pengambilan pakaian. Aplikasi ini juga memberikan pengalaman pengguna yang lebih nyaman dan praktis dengan menyediakan fitur-fitur seperti pembayaran digital, pilihan jadwal pengantaran yang fleksibel, dan kemampuan untuk memberikan umpan balik terhadap layanan yang diberikan.

Kelompok kami membuat aplikasi Laundry ini di aplikasi Netbeans dengan bahasa pemrograman java. Bahasa java ini digunakan untuk mengontrol beberapa fungsi button pada aplikasi kami nantinya. Lalu kami juga menghubungkannya dengan database (backend) dan membuat server aplikasi. Kelompok kami membuat halaman web untuk untuk menginputkan data mahasiswa ke database. Di halaman web tersebut juga telah tersedia perhitungan total nilai otomatis ketika nilai diinputkan.

1.2 Rumusan Masalah

1. Bagaimana cara membuat aplikasi dengan bahasa pemrograman java di Netbeans ?
2. Apa saja kebutuhan dari aplikasi Laundry Boyz ?
3. Apa saja fitur dalam aplikasi laundry Boyz menggunakan NetBeans?
4. Apa saja rancangan database aplikasi Laundry Boyz?

1.3 Tujuan

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan tugas ini adalah membuat suatu aplikasi untuk menginput data, meyimpan data dan menghitung biaya laundry pelanggan yang menggunakan aplikasi laundry secara otomatis. Adapun tujuan dari pembuatan aplikasi ini, yaitu:

1. Memberikan efisiensi waktu untuk kami dalam mendata jumlah pelanggan yang menggunakan aplikasi laundry.
2. Penyimpanan data pelanggan lebih aman karena disimpan pada database.

1.4 Ruang Lingkup

Pada makalah ini berfokus kepada :

1. Perancangan sistem aplikasi laundry agar meningkatkan efisiensi petugas dalam hal administrasi dan dokumentasi
2. Mengaplikasikan perancangan aplikasi Laundry yang telah dibuat dengan menggunakan bahasa pemrograman Java, Database dan PHP.

BAB II

TINJAUAN PUSTAKA

1. UML (*Unified Modelling Language*)

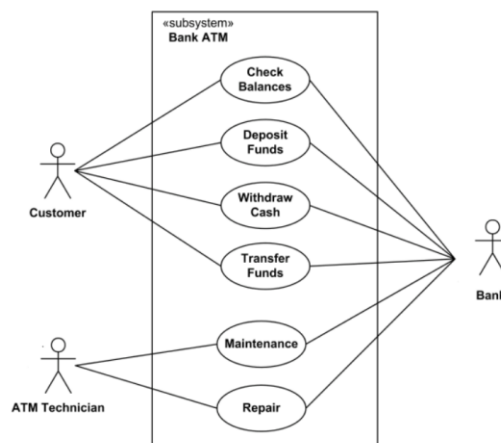
1.1. Definisi UML

UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. (Munawar, 2005:17). Dengan kata lain UML adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek.

UML juga dapat digunakan sebagai alat transfer ilmu tentang sistem aplikasi yang akan dikembangkan dari developer satu ke developer lainnya. UML sangat penting bagi sebagian orang karena UML berfungsi sebagai bridge atau jembatan penerjemah antara pengembang sistem dengan pengguna. Di sinilah pengguna dapat memahami sistem yang nantinya akan dikembangkan.

1.2. Diagram UML

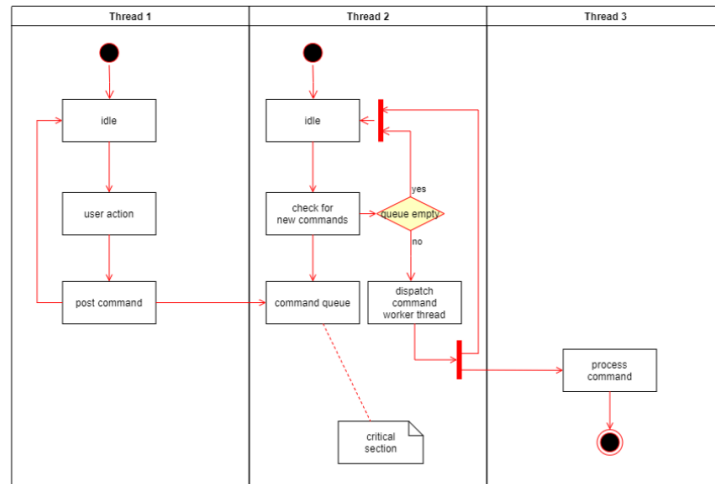
- Use Case Diagram



Use Case Diagram adalah satu jenis dari diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. Use Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. Use Case merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu

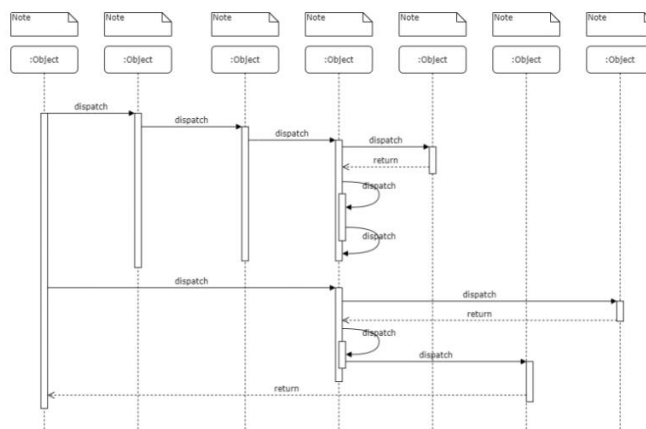
menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada Use Case.

- Activity Diagram



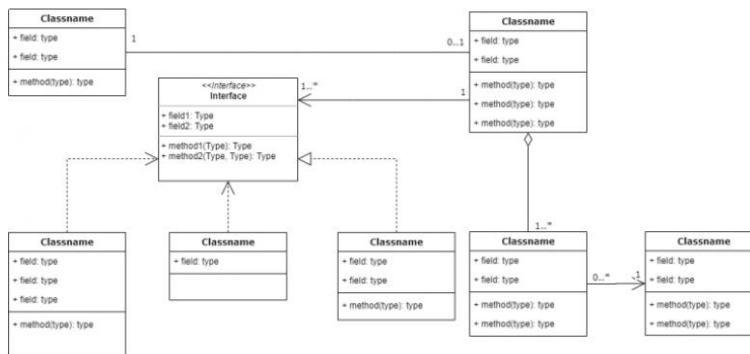
Activity diagram atau dalam bahasa Indonesia berarti diagram aktivitas, merupakan sebuah diagram yang dapat memodelkan berbagai proses yang terjadi pada sistem. Seperti layaknya runtutan proses berjalannya suatu sistem dan digambarkan secara vertikal. Activity diagram adalah salah satu contoh diagram dari UML dalam pengembangan dari Use Case.

- Sequence Diagram



Sequence diagram merupakan diagram yang menjelaskan interaksi objek berdasarkan urutan waktu. Sequence dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu, seperti yang tertera pada Use Case diagram.

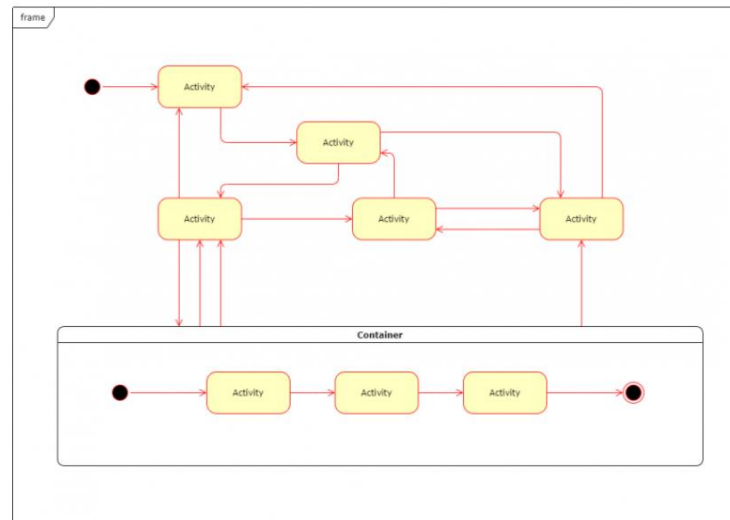
- Class Diagram



Class diagram atau diagram kelas merupakan suatu diagram yang digunakan untuk menampilkan kelas-kelas berupa pake-paket untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya.

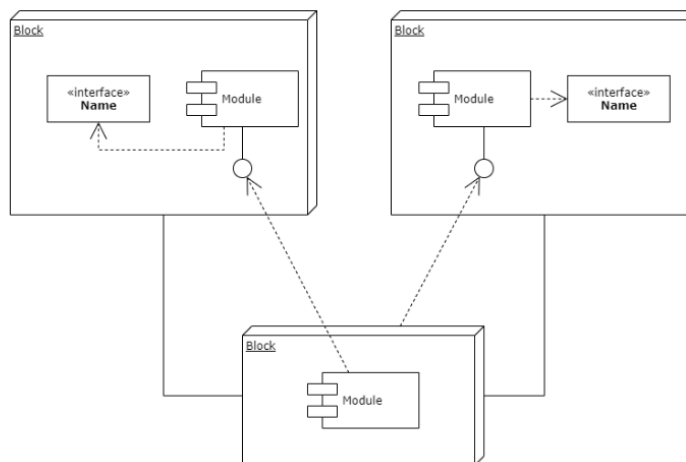
Namun, pada Class diagram desain modelnya dibagi menjadi 2 bagian. Class diagram yang pertama merupakan penjabaran dari domain model yang merupakan abstraksi dari basis data. Class diagram yang kedua merupakan bagian dari modul program MVC pattern (Model View Controller), di mana terdapat class boundary sebagai class interface, class control sebagai tempat ditemukannya algoritma, dan class entity sebagai tabel dalam basis data dan query program.

- Stateachine Diagram



Statemachine yaitu salah satu jenis diagram pada UML yang berfungsi untuk menggambarkan transisi serta perubahan pada suatu objek pada sistem.

- Component Diagram



Component diagram yang berfungsi untuk menggambarkan software pada suatu sistem. Component diagram merupakan penerapan pada piranti lunak atau software dari satu class maupun lebih, dan biasanya berupa file data, source code,.exe, table, dokumen, atau yang lainnya.

2. PHP

PHP merupakan singkatan dari Hypertext Preprocessor. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnya yang dikirim ke klien, tempat pemakai menggunakan browser. Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi tool yang disebut “Personal Home Page”.

Paket inilah yang menjadi cikal bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI versi 2. Pada versi inilah pemrograman dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan database dan melakukan perhitungan-perhitungan yang kompleks sambil jalan (Kadir, 2008:2). PHP adalah bahasa pemrograman berbentuk skrip yang mengandung fungsi-fungsi dalam membentuk sistem yang sistematis.

3. Database

Database atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (query) basis data disebut sistem manajemen basis data (database management system, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan objek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data.

Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan di mana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel.

Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. Istilah basis data mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (database management system/DBMS).

4. XAMPP

XAMPP adalah software web server apache yang di dalamnya tertanam server MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat website yang dinamis. XAMPP sendiri mendukung dua system operasi yaitu windows dan Linux. Untuk linux dalam proses penginstalanny menggunakan command line sedangkan untuk windows dalam proses penginstalannya menggunakan interface grafis sehingga lebih mudah dalam penggunaan XAMPP di Windows di banding dengan Linux. XAMPP yang merupakan aplikasi vital bagi berjalannya XAMPP dengan baik.

BAB III

PEMBAHASAN

1. Gambaran Aplikasi

1.1. Halaman Login



1.2. Halaman Home



1.3. Halaman Pemesanan

LAUN-DRY BOYZ
Pesan Disini

Nama: Ferdinand Jenis Pewangi: Cair
Nomor Telepon: 0899696969 Jumlah Pakaian: 6 /KG
Alamat: Alamanda Barcelona Harga: 65000
No Antrian:

Nomor A...	Nama	Nomor T...	Alamat	Jenis_p...	Jumlah	Harga	Tanggal
------------	------	------------	--------	------------	--------	-------	---------

1.4. Halaman Pengambilan

LAUN-DRY BOYZ
MEMBUAT PAKAIAN ANDA BERSINAR SEPERTI BARU!

No Antr...	Nama	No Tel...	Alamat	Jenis P...	Jumlah	Harga	Tanggal
6	Ferdina...	089969...	Alaman...	Cair	6	65000	2023-0...

Nomor Antrian: 6 Total Harga: 65000
Nama: Ferdinand Jumlah: 6 Harga Bayar: 100000
Nomor Telepon: 0899696969 Harga Total: 65000 Kembali: 35000
Alamat: anda Barcelona Jenis Pewangi: Cair
Tanggal: 2023-06-11

No...	Nama	No ...	Ala...	Jeni...	Jum...	Harga	Tan...	Total	Bayar	Ke...
3	mila	087...	cicur...	Bub...	1	130...	201...	130...	20000	7000

11/06/2023

1.5. Tampilan Selesai Pembayaran

**LAUN-DRY BOYZ**
Jl. Kartawijaya, Kota Bekasi







Nama : Ferdinand
Tanggal : Rp 2023-06-11
Total : Rp 65000
Bayar : Rp 100000
Kembalian : Rp 35000


Selamat Datang Kembali Boyz

© 2023 LAUN-DRY BOYZ

1.6. Halaman Laporan (Report)

← → ↺ ⓘ localhost/Laporanjava/laporan.php

 (4) WhatsApp  Gmail  YouTube  Vclass  Maps  Edit Page < ravenbo...

 Laporan Laun-Dry Boyz
Jl. Kartawijaya, Kota Bekasi

No Antrian	Nama	No Telepon	Alamat	Jenis Pewangi	Jumlah	Harga	Tanggal	Total	Bayar	Kembalian
3	mila	087770432050	cicurug	Bubuk	1	13000	2018-05-26	13000	20000	7000
6	Ferdinand	0899696969	Alamanda Barcelona	Cair	6	65000	2023-06-11	65000	100000	35000

© 2023 LAUN-DRY BOYZ

2. Analisis Kebutuhan

2.1. Data yang dibutuhkan

Data yang dibutuhkan untuk pengembangan aplikasi Laundry :

- Data Pemesanan : Nomor antrian, nama pemesan, nomor telepon pemesan, alamat pemesan, jumlah pakaian, jenis pewangin yang digunakan, harga yang harus dibayar, tanggal pesanan dilakukan
- Data Pengembalian : Nomor antrian, nama pemesan, nomor telepon pemesan, alamat pemesan, jumlah pakaian, jenis pewangin yang digunakan, harga yang harus dibayar, tanggal pesanan dilakukan, Uang yang dibayar pemesan, uang kembalian.

2.2. Kebutuhan fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses yang dilakukan oleh sistem dan informasi yang dihasilkan sistem. Adapun proses dan informasi yang dihasilkan adalah:

- Proses login yang bisa dilakukan oleh petugas Laundry
- Proses pengelolaan data pemesanan, meliputi input, dan delete yang dilakukan oleh petugas Laundry
- Proses penyimpanan pelaporan mengenai pemesanan dan pengambilan pakaian pemesan yang dilakukan oleh sistem.
- Proses pembayaran Laundry akan dicetak oleh sistem saat pemesan sudah melakukan pembayaran dilakukan oleh petugas
- Sistem dapat mencetak transaksi yang ada pada Laundry.
- Sistem dapat menyimpan data histori atau dokumentasi atas transaksi Laundry.

2.3. Kebutuhan Non Fungsional

2.3.1. Perangkat

- Java sebagai bahasa pemrograman pembuatan aplikasi
- PHP sebagai bahasa pemrograman penghubung Aplikasi dengan Database
- MYSQL sebagai program DBMS (*Database Management System*)

2.3.2. Keamanan

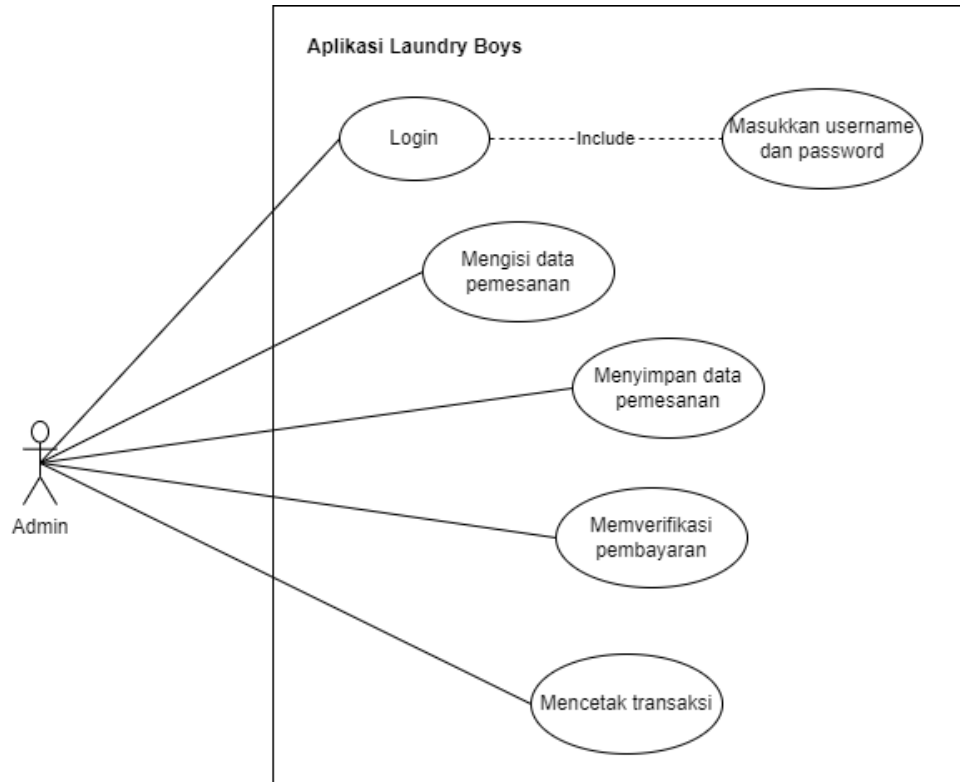
Penggunaan username dan password dalam form login agar hanya petugas Laundry atau admin yang dapat mengaksesnya saja.

2.3.3.Interface

Tampilan dari aplikasi sangat jelas dan memudahkan petugas Laundry agar bisa menggunakan aplikasi ini dengan baik.

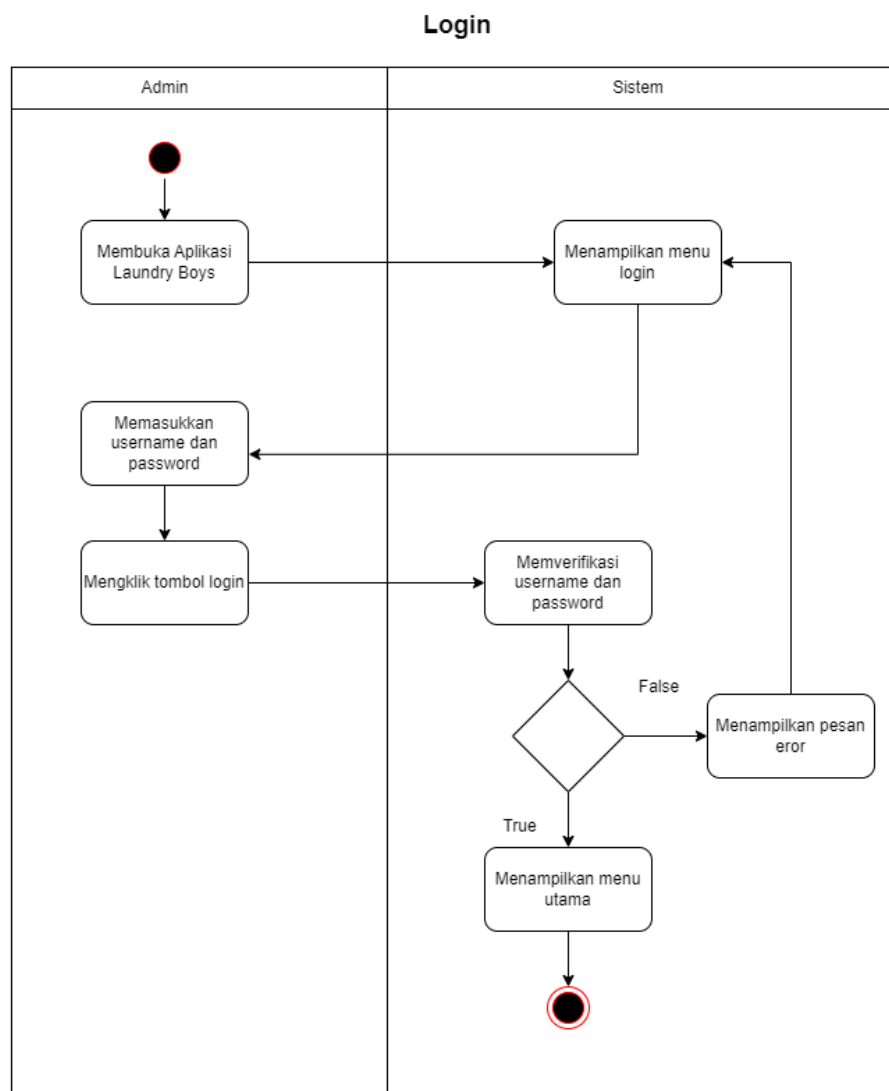
3. Rancangan Aplikasi

3.1. Use Case Diagram

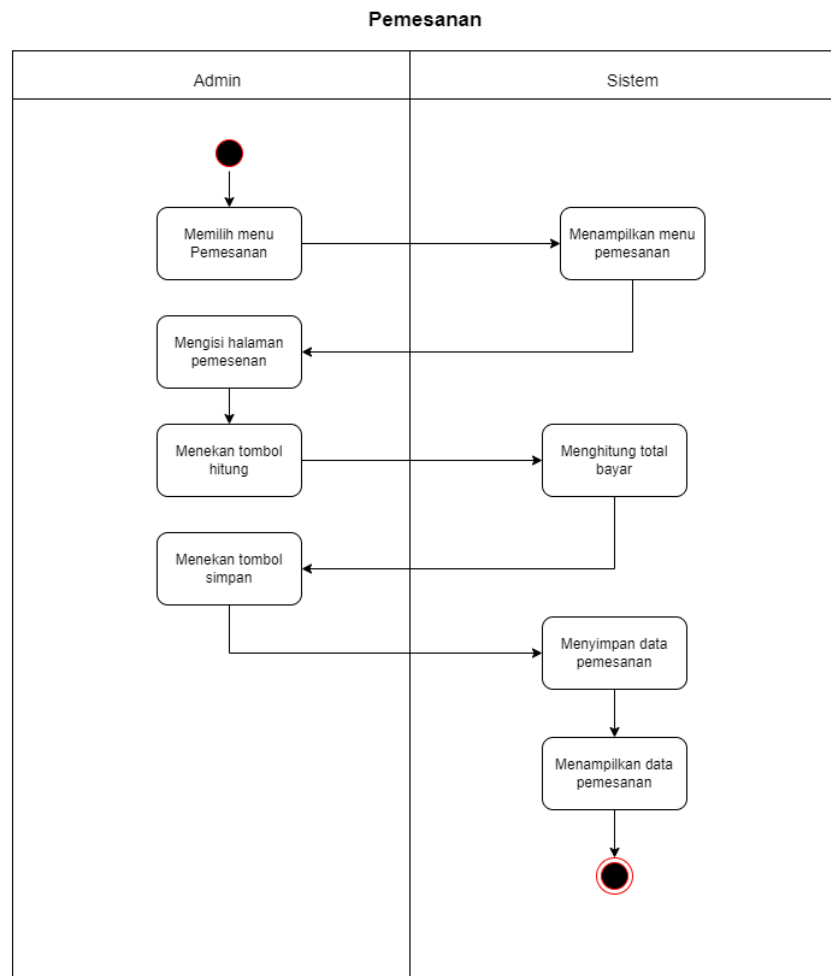


3.2. Activity Diagram

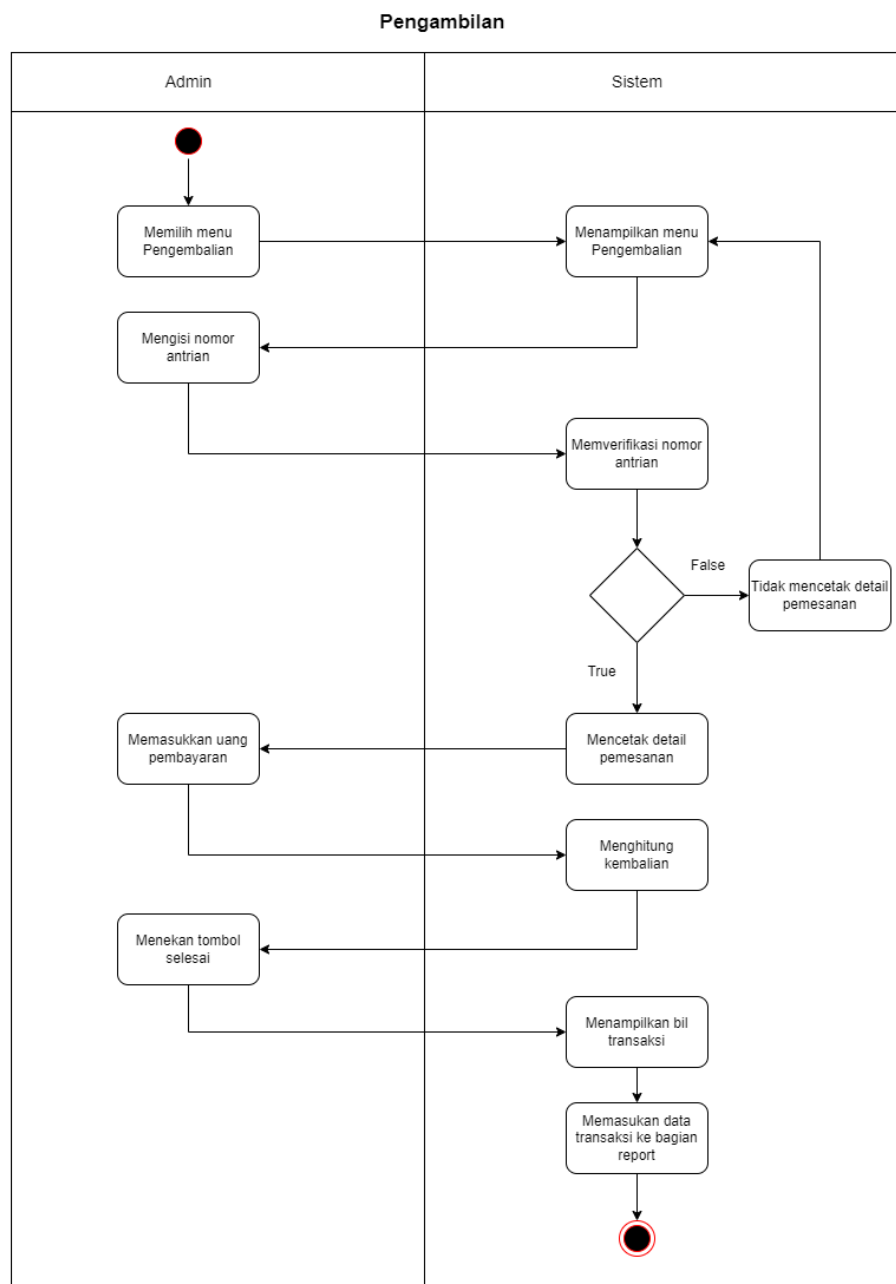
3.2.1.Login



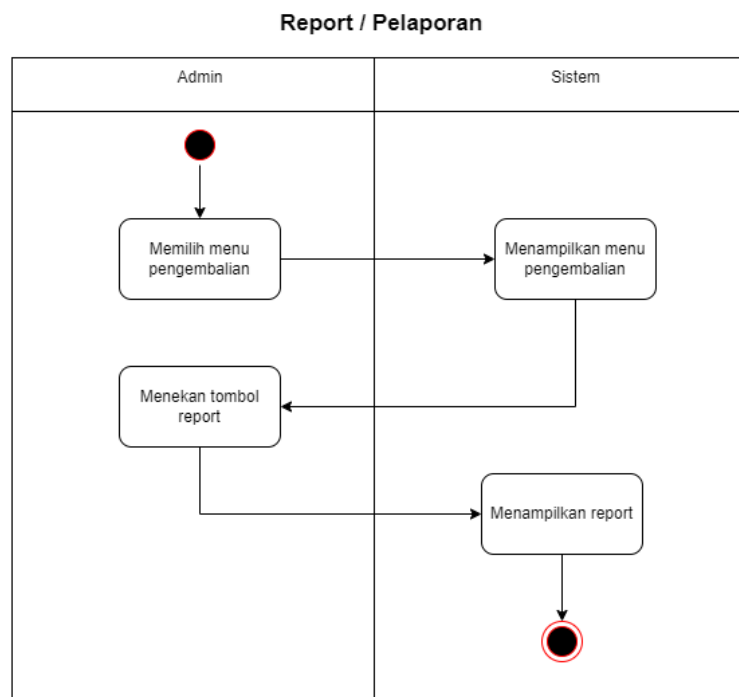
3.2.2.Pemesanan



3.2.3. Pengembalian



3.2.4.Report / Pelaporan



4. Rancangan Database

tbl_login	
	Nama Kolom
	user
	pass

tbl_ambil	
	Nama Kolom
	no_antrian
	nama
	no_telp
	alamat
	jumlah
	harga
	tanggal
	bayar
	kembali

tbl_pesanan	
	Nama Kolom
PK	no_antrian
	nama
	no_telp
	alamat
	jumlah
	harga
	tanggal

Struktur Pada Tabel Login

<div> Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Pelacakan Trigger </div>									
Struktur tabel		Tampilan hubungan							
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	user	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	2	pass	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<div> Pilih Semua Dengan pilihan: Jelajahi Ubah Hapus Utama Unik Indeks Spasial Teks penuh Add to central columns Remove from central columns </div>									

Struktur Pada Tabel Pemesanan

<div> Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Pelacakan Trigger </div>									
Struktur tabel		Tampilan hubungan							
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	no_antrian	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2	nama	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3	no_telp	varchar(13)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4	alamat	text	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	5	j_pewangi	varchar(5)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6	jumlah	int(11)		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	7	harga	int(11)		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	8	tanggal	date		Tidak	Tidak ada			Ubah Hapus Lainnya

Struktur Pada Tabel Pengambilan

Struktur tabel		Tampilan hubungan								
	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	no_antrian	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	2	nama	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3	no_telp	varchar(13)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4	alamat	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	5	j_pewangi	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6	jumlah	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	7	harga	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	8	tanggal	date			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	9	total	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	10	bayar	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	11	kembali	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya

5. Pembuatan Aplikasi

5.1. Listing Home

```
package laundry;

import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author LENOVO
 */
public class Home extends javax.swing.JFrame {

    /**
     * Creates new form home
     */
    public Home() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jPanel3 = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        pesan = new javax.swing.JButton();
        ambil = new javax.swing.JButton();
        jPanel2 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    }
}
```

Program Java di atas merupakan implementasi GUI (Graphical User Interface) untuk aplikasi laundry. Kode tersebut terdiri dari kelas "Home" yang merupakan turunan dari kelas "javax.swing.JFrame" dan berada dalam package "laundry".

Kelas "Home" memiliki konstruktor "Home()" yang digunakan untuk menginisialisasi tampilan GUI. Dalam konstruktor ini, terdapat pemanggilan metode "initComponents()" yang bertujuan untuk menginisialisasi komponen-komponen GUI yang ada.

Program ini mengandalkan Swing, sebuah toolkit GUI untuk Java, yang memungkinkan penggunaan elemen-elemen GUI seperti JPanel, JLabel, JButton, dan sebagainya. Dalam tampilan GUI, terdapat dua panel yaitu jPanel1 dan jPanel2. Pada jPanel1, terdapat label jLabel3 dan jLabel1, serta dua tombol pesan dan ambil. Sedangkan pada jPanel2, terdapat label jLabel2.


```

private void pesanActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_pesanActionPerformed
    // TODO add your handling code here:
    Pemesanan a2 = new Pemesanan();
    a2.show();
    this.setVisible(false);
} //GEN-LAST:event_pesanActionPerformed

private void ambilActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_ambilActionPerformed
    // TODO add your handling code here:
    Pembayaran a3 = null;
    try {
        a3 = new Pembayaran();
    } catch (SQLException ex) {
        Logger.getLogger(Home.class.getName()).log(Level.SEVERE, null, ex);
    }
    a3.show();
    this.setVisible(false);
} //GEN-LAST:event_ambilActionPerformed

```

Pada event handler "pesanActionPerformed()", ketika tombol "pesan" ditekan, akan dibuat objek baru dari kelas "Pemesanan" dengan menggunakan konstruktor default. Kemudian, tampilan dari objek tersebut ditampilkan menggunakan method "show()". Selanjutnya, tampilan frame utama "Home" diatur sebagai tidak terlihat dengan menggunakan method "setVisible(false)".

Pada event handler "ambilActionPerformed()", ketika tombol "ambil" ditekan, akan dibuat objek baru dari kelas "Pembayaran" dengan menggunakan konstruktor yang dapat melempar SQLException. Kemudian, tampilan dari objek tersebut ditampilkan menggunakan method "show()". Selanjutnya, tampilan frame utama "Home" diatur sebagai tidak terlihat dengan menggunakan method "setVisible(false)".

Kode ini mengatur perilaku ketika tombol "pesan" dan "ambil" ditekan. Ketika tombol "pesan" ditekan, akan membuka tampilan "Pemesanan" dan menutup tampilan "Home". Sedangkan, ketika tombol "ambil" ditekan, akan membuka tampilan "Pembayaran" dan menutup tampilan "Home". Dengan demikian, event handler ini mengatur transisi antar tampilan dalam aplikasi laundry.

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Home().setVisible(true);
        }
    });
}

```

Pada metode "main", terdapat beberapa langkah yang dilakukan sebelum menampilkan tampilan GUI utama. Pertama, dilakukan pengaturan tampilan dan gaya menggunakan "Nimbus look and feel" dengan menggunakan loop untuk mencari dan mengatur gaya yang sesuai. Jika gaya "Nimbus" ditemukan, maka akan diatur sebagai gaya tampilan yang digunakan.

Selanjutnya, dalam blok "java.awt.EventQueue.invokeLater", dibuat objek dari kelas "Home" dan tampilan frame utamanya diatur sebagai terlihat dengan menggunakan method "setVisible(true)". Ini akan menampilkan tampilan GUI utama aplikasi laundry kepada pengguna. Dengan demikian, kode ini bertanggung jawab untuk memulai dan menampilkan tampilan GUI utama aplikasi laundry saat program dijalankan.

```

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton ambil;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JButton pesan;
// End of variables declaration//GEN-END:variables
}

```

Komponen-komponen GUI yang dideklarasikan meliputi tombol "ambil" dan "pesan", serta label "jLabel1", "jLabel2", dan "jLabel3". Selain itu, terdapat juga panel-panel "jPanel1", "jPanel2", dan "jPanel3" yang digunakan untuk mengatur tata letak dan penempatan komponen-komponen GUI.

Deklarasi variabel dilakukan untuk menyimpan referensi ke objek-objek yang akan digunakan dalam tampilan GUI. Variabel-variabel ini akan digunakan untuk mengakses dan mengatur properti serta perilaku dari komponen GUI tersebut, seperti mengubah teks pada label, menambahkan aksi pada tombol, dan lain sebagainya.

5.2. Listing Login

```
package laundry;

import java.sql.Connection;;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import koneksi.koneksiDB.con;
import laundry.Home;

/**
 *
 * @author LENOVO
 */
public class Login extends javax.swing.JFrame {

    public Statement st;
    public ResultSet rs;
    public DefaultTableModel tabmodel;

    Connection con = (Connection) koneksi.koneksiDB.getkoneksi();
    /**
     * Creates new form login
     */
    public Login() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents
    private void initComponents() {

        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        user = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        pass = new javax.swing.JPasswordField();
        blogin = new javax.swing.JButton();
        blogin1 = new javax.swing.JButton();
        jSeparator2 = new javax.swing.JSeparator();
        jSeparator3 = new javax.swing.JSeparator();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

Kelas "Login" mengimport beberapa kelas dan paket yang diperlukan, seperti kelas Connection, ResultSet, Statement, dan DefaultTableModel. Kelas ini juga mengimport paket koneksiDB dan Home untuk memanfaatkan fungsi-fungsi yang telah didefinisikan di dalamnya.

Dalam kelas "Login", terdapat beberapa variabel yang dideklarasikan, antara lain st (Statement), rs (ResultSet), dan tabmodel (DefaultTableModel). Variabel-variabel ini akan digunakan untuk melakukan operasi-operasi database, seperti eksekusi query dan memanipulasi data.

Selain itu, kelas "Login" juga menginisialisasi beberapa komponen GUI, seperti label-label ("jLabel1", "jLabel2", dan "jLabel3"), field input ("user" dan "pass"), serta tombol-tombol ("blogin" dan "blogin1"). Komponen-komponen ini akan digunakan untuk menerima input dari pengguna dan memprosesnya. Pada metode konstruktor "Login()", komponen-komponen GUI diinisialisasi melalui pemanggilan metode " initComponents()" yang merupakan bagian dari metode konstruktor.

Metode ini akan memuat tampilan GUI login yang telah dibuat di dalam NetBeans. Dengan adanya implementasi kelas "Login" ini, pengguna dapat memasukkan informasi login, seperti username dan password, untuk mengakses sistem aplikasi laundry.

```
private void userActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_userActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_userActionPerformed

private void bloginActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_bloginActionPerformed
    // TODO add your handling code here:
    try {
        st = con.createStatement();
        rs = st.executeQuery("select*from tbl_login where "+user.getText()+"");

        if(rs.next()){
            if (pass.getText().equals(rs.getString("pass"))) {
                JOptionPane.showMessageDialog(rootPane, "Bisa");
                setVisible(false);
                new Home().setVisible(true);
            } else {
                JOptionPane.showMessageDialog(rootPane, "Password Salah");
                user.setText("");
                pass.setText("");
                pass.requestFocus();
            }
        } else {
            JOptionPane.showMessageDialog(rootPane, "Username Salah");
            user.setText("");
            pass.setText("");
            user.requestFocus();
        }
    } catch (Exception e){
        e.printStackTrace();
    }
} //GEN-LAST:event_bloginActionPerformed

private void blogin1ActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_blogin1ActionPerformed
    user.setText("");
    pass.setText("");
} //GEN-LAST:event_blogin1ActionPerformed

private void passActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_passActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_passActionPerformed
```

Pada tampilan GUI login, terdapat field "USERNAME" dan "PASSWORD" yang digunakan oleh pengguna untuk menginputkan informasi login. Selain itu, terdapat tombol "LOGIN" dan "RESET" yang memberikan aksi kepada pengguna untuk melakukan login atau mereset field input.

Ketika pengguna melakukan aksi pada tombol "LOGIN", dilakukan pengecekan pada database untuk mencari data pengguna dengan username yang sesuai. Jika data pengguna ditemukan dan password yang diinputkan cocok dengan password yang ada di database, maka tampilan GUI login akan ditutup dan GUI utama (Home) akan ditampilkan. Jika password tidak cocok, pesan kesalahan akan ditampilkan. Jika username tidak ditemukan, pesan kesalahan juga akan ditampilkan.

Selain itu, terdapat tombol "RESET" yang digunakan untuk mereset field input username dan password. Jadi, jika pengguna ingin mengosongkan field input atau mengulang proses login, mereka dapat menggunakan tombol ini. Dengan implementasi ini, aplikasi laundry dapat melakukan proses autentikasi pengguna berdasarkan data yang ada di database, dan memberikan respons yang sesuai kepada pengguna tergantung pada kecocokan data yang ditemukan.

```
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    });
}
```

Pada blok kode di dalam method main, terdapat pengaturan "look and feel" untuk tampilan GUI menggunakan Nimbus. "Look and feel" adalah tampilan visual dan perilaku dari komponen GUI. Dalam blok try, program mencoba mencari dan mengatur "look and feel" Nimbus jika tersedia. Jika tidak tersedia, maka akan menggunakan "look and feel" default dari sistem operasi.

Selanjutnya, dalam blok create and display the form, sebuah objek dari kelas Login dibuat dan ditetapkan sebagai tampilan yang terlihat dengan memanggil method setVisible(true). Hal ini menyebabkan tampilan GUI login ditampilkan kepada pengguna saat program dijalankan.

Dengan implementasi ini, ketika program dijalankan, tampilan GUI login akan ditampilkan kepada pengguna sesuai dengan pengaturan "look and feel" yang telah ditentukan. Pengguna dapat memasukkan informasi login dan melanjutkan ke tampilan selanjutnya jika login berhasil.

A screenshot of a code editor showing Java variable declarations. The code is enclosed in a dark-themed window with three colored window control buttons (red, yellow, green) in the top-left corner. The code itself is as follows:

```
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton blogin;
private javax.swing.JButton blogin1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel2;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JPasswordField pass;
private javax.swing.JTextField user;
// End of variables declaration//GEN-END:variables
}
```

Setiap komponen GUI diberi nama dan tipe data yang sesuai, serta diberi aksesibilitas private agar hanya dapat diakses oleh kelas yang mendeklarasikannya. Variabel-variabel ini digunakan untuk mengakses dan mengatur properti serta perilaku dari komponen-komponen GUI tersebut.

Dengan adanya deklarasi variabel-variabel ini, kita dapat mengakses dan memanipulasi komponen-komponen GUI dalam kode program lainnya. Misalnya, untuk menambahkan event listener pada tombol atau mengambil nilai yang dimasukkan pengguna dari field input. Secara keseluruhan, deklarasi variabel-variabel ini memfasilitasi penggunaan dan interaksi dengan komponen-komponen GUI yang ada dalam tampilan login.

5.3. Listing Pembayaran

```
package laundry;
import java.awt.Desktop;
import java.sql.Connection;
import java.sql.Statement;
import java.awt.Dimension;
import java.awt.HeadlessException;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.io.File;
import java.net.URL;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.HashMap;
import java.util.Locale;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.Timer;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author LENOVO
 */
public class Pembayaran extends javax.swing.JFrame {
    Model model = new Model();
    java.util.Date tglsekarang = new java.util.Date();
    private final SimpleDateFormat smpdfmt = new SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());
    private final String tanggal = smpdfmt.format(tglsekarang);
    private Statement st;
    private Connection con;
    private ResultSet rs;
    public DefaultTableModel tabmodel;
    public DefaultTableModel tabmodel2;
    public DefaultTableModel tbldata;
```

Kelas Pembayaran memiliki beberapa import statement yang mengimpor kelas-kelas yang diperlukan untuk fungsi-fungsi yang akan digunakan dalam kelas ini. Misalnya, import untuk mengatur aksi event, mengelola koneksi ke database, mengelola waktu dan tanggal, serta mengatur tampilan tabel.

Kelas Pembayaran juga memiliki beberapa atribut dan objek yang dideklarasikan di dalamnya, seperti objek Model, objek SimpleDateFormat, objek-objek untuk mengatur koneksi ke database, serta objek DefaultTableModel. Atribut dan objek tersebut digunakan untuk menyimpan data, mengelola koneksi ke database, dan mengatur tampilan tabel pada antarmuka pengguna.

Dalam konteks ini, kelas Pembayaran bertanggung jawab untuk mengatur logika dan tampilan yang berkaitan dengan proses pembayaran dalam sistem laundry. Hal ini mencakup pengelolaan waktu dan tanggal, pengelolaan koneksi ke database, pengambilan data dari database, serta pengaturan tampilan tabel untuk menampilkan data pembayaran.

```

private Object displaytext;
private void koneksi() {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con = (Connection) DriverManager.getConnection("jdbc:mysql://localhost/db_laundry", "root", "");
        st = (Statement) con.createStatement();
        System.out.println("berhasil");
    } catch (ClassNotFoundException | SQLException e) {
        System.err.println("koneksi gagal" + e.getMessage());
    }
}
String hasil;
/**
 * Creates new form pembayaran
 */
public Pembayaran() throws SQLException {
    initComponents();
    koneksi();
    judul();
    tampildata();
    judul1();
    tampildata1();
    tgl.setText(tanggal);
}

/**
 *
 * @throws SQLException
 */
public void tampildata(){
    try{
        st = con.createStatement();
        tabmodel.getDataVector().removeAllElements();
        tabmodel.fireTableDataChanged();
        rs = st.executeQuery("SELECT * FROM tbl_pesan");
        while(rs.next()){
            Object[] data = {
                rs.getString("no_antrian"),
                rs.getString("nama"),
                rs.getString("no_telp"),
                rs.getString("alamat"),
                rs.getString("j_pewangi"),
                rs.getString("jumlah"),
                rs.getString("harga"),
                rs.getString("tanggal")
            };
            tabmodel.addRow(data);
        }
    } catch (Exception e){
        e.printStackTrace();
    }
}

```

Metode koneksi() digunakan untuk melakukan koneksi ke database dengan menggunakan driver MySQL. Jika koneksi berhasil, maka akan ditampilkan pesan "berhasil". Jika terjadi exception pada proses koneksi, maka akan ditampilkan pesan "koneksi gagal" beserta pesan error yang terkait. Konstruktor Pembayaran digunakan untuk inisialisasi objek kelas.

Pada konstruktor ini, dilakukan pemanggilan metode koneksi() untuk melakukan koneksi ke database. Selain itu, juga dilakukan pemanggilan metode judul() dan tampildata() untuk mengatur judul dan menampilkan data dalam tabel pada antarmuka pengguna. Selain itu, tanggal saat ini juga ditampilkan pada komponen tgl.

Metode tampildata() digunakan untuk mengambil data dari tabel tbl_pesan dalam database dan menampilkannya dalam tabel pada antarmuka pengguna. Proses pengambilan data dilakukan dengan menjalankan query SQL `SELECT * FROM`

tbl_pesan, kemudian data yang dihasilkan dimasukkan ke dalam objek tabmodel yang bertipe DefaultTableModel menggunakan metode addRow(data).

```
private void judul(){
    Object[] judul = {"No Antrian", "Nama", "No Telepon", "Alamat", "Jenis Pewangi", "Jumlah", "Harga", "Tanggal"};
    tabmodel = new DefaultTableModel(null, judul);
    tabel1.setModel(tabmodel);
}

public void tampildata1(){
    try {
        st = con.createStatement();
        tabmodel2.getDataVector().removeAllElements();
        tabmodel2.fireTableDataChanged();
        rs = st.executeQuery("SELECT * FROM tbl_ambil");
        while (rs.next()) {
            Object[] data = {
                rs.getString("no_antrian"),
                rs.getString("nama"),
                rs.getString("no_telp"),
                rs.getString("alamat"),
                rs.getString("j_pewangi"),
                rs.getString("jumlah"),
                rs.getString("harga"),
                rs.getString("tanggal"),
                rs.getString("total"),
                rs.getString("bayar"),
                rs.getString("kembali");
            tabmodel2.addRow(data);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private void judul1(){
    Object[] judul = {"Nomor Antrian", "Nama", "No Telepon", "Alamat", "Jenis Pewangi", "Jumlah", "Harga", "Tanggal", "Total", "Bayar", "Kembalian"};
    tabmodel2 = new DefaultTableModel(null, judul);
    tabel2.setModel(tabmodel2);
}

private void reset()
{
    no_transaksi.setText("");
    tgl.setText("");
    nama.setText("");
    j_pewangi.setSelectedIndex(-1);
    jumlah.setText("");
    j_pewangi.setSelectedItem("");
    nama.setText("");
    no_telp.setText("");
    alamat.setText("");
    ttanggal.setText("");
    harga.setText("");
    totall.setText("");
    bayar.setText("");
    kembalian.setText("");
    no_transaksi.requestFocus();
}
```

Metode judul() digunakan untuk mengatur judul kolom pada tabel tabel1 yang menampilkan data pesanan. Pada metode ini, didefinisikan array judul yang berisi nama-nama kolom yang akan ditampilkan, yaitu "No Antrian", "Nama", "No Telepon", "Alamat", "Jenis Pewangi", "Jumlah", "Harga", dan "Tanggal". Selanjutnya, objek tabmodel yang bertipe DefaultTableModel dibuat dengan menggunakan array judul sebagai header kolom, dan tabmodel tersebut diatur sebagai model tabel pada tabel1.

Metode tampildata1() digunakan untuk mengambil data dari tabel tbl_ambil dalam database dan menampilkannya dalam tabel pada antarmuka pengguna. Proses pengambilan data dilakukan dengan menjalankan query SQL SELECT * FROM

tbl_ambil, kemudian data yang dihasilkan dimasukkan ke dalam objek tabmodel2 yang bertipe DefaultTableModel menggunakan metode addRow(data).

Metode judul1() digunakan untuk mengatur judul kolom pada tabel tabel2 yang menampilkan data pengambilan pesanan. Pada metode ini, juga didefinisikan array judul yang berisi nama-nama kolom yang akan ditampilkan pada tabel tabel2, yaitu "Nomor Antrian", "Nama", "No Telepon", "Alamat", "Jenis Pewangi", "Jumlah", "Harga", "Tanggal", "Total", "Bayar", dan "Kembalian". Selanjutnya, objek tabmodel2 yang bertipe DefaultTableModel dibuat dengan menggunakan array judul sebagai header kolom, dan tabmodel2 tersebut diatur sebagai model tabel pada tabel2.

Metode reset() digunakan untuk mengatur ulang nilai-nilai pada komponen-komponen antarmuka pengguna menjadi kosong atau nilai default. Pada metode ini, nilai-nilai pada berbagai komponen seperti no_transaksi, tgl, nama, j_pewangi, jumlah, no_telp, alamat, ttanggal, harga, totall, bayar, dan kembalian diatur menjadi kosong atau nilai default. Komponen no_transaksi juga diberikan fokus agar dapat langsung diisi setelah direset.

```

private void ttanggalActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_ttanggalActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_ttanggalActionPerformed

private void tabelMouseClicked(java.awt.event.MouseEvent evt) {GEN-FIRST:event_tabelMouseClicked
    // TODO add your handling code here:
} //GEN-LAST:event_tabelMouseClicked

private void bayarKeyReleased(java.awt.event.KeyEvent evt) {GEN-FIRST:event_bayarKeyReleased
    if (Integer.parseInt(totall.getText()) > Integer.parseInt(bayar.getText())) {
        kembalian.setText("Uang Kurang!");
    } else {
        Integer kembali = Integer.parseInt(bayar.getText()) - Integer.parseInt(totall.getText());
        kembalian.setText(kembali.toString());
    }
} //GEN-LAST:event_bayarKeyReleased

private void selesaiActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_selesaiActionPerformed
    int a, b;
    a = Integer.parseInt(totall.getText());
    b = Integer.parseInt(bayar.getText());
    if (b < a) {
        JOptionPane.showMessageDialog(null, "Uang Anda Kurang");
    } else {
        String sql = "no_antrian =" + no_transaksi.getText()
        + ", nama=" + nama.getText() + ", no_telp=" + no_telp.getText() + ", alamat=" + alamat.getText()
        + ", j_pewangi=" + j_pewangi.getSelectedItem() + ", jumlah=" + jumlah.getText() + ", harga=" + harga.getText() + ", tanggal=" + ttangg
        al.getText() + ", total=" + totall.getText() + ", bayar=" + bayar.getText() + ", kembali=" + kembalian.getText() + """;
        if (model.hapus("tbl_pesanan", "no_antrian", no_transaksi.getText())) {
            if (model.simpan("tbl_ambil", sql)) {
                JOptionPane.showMessageDialog(null, "Berhasil");
                tampildata();
                tampildata1();
                if (Desktop.isDesktopSupported()) {
                    try {
                        Desktop.getDesktop().browse(new URL("http://localhost/Laporanjava/struk.php?id=" +
                        no_transaksi.getText().toURI()));
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
                reset();
            } else {
                JOptionPane.showMessageDialog(null, "gagal simpan");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Gagal Hapus");
        }
    }
} //GEN-LAST:event_selesaiActionPerformed

private void kembaliActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_kembaliActionPerformed
    // TODO add your handling code here:
    new Home().setVisible(true);
    dispose();
} //GEN-LAST:event_kembaliActionPerformed

private void ReportActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_ReportActionPerformed
    // TODO add your handling code here:
    if (Desktop.isDesktopSupported()) {
        try {
            Desktop.getDesktop().browse(new URL("http://localhost/Laporanjava/laporan.php").toURI());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
} //GEN-LAST:event_ReportActionPerformed

private void ReportMouseClicked(java.awt.event.MouseEvent evt) {GEN-FIRST:event_ReportMouseClicked
    // TODO add your handling code here:
} //GEN-LAST:event_ReportMouseClicked

private void tabel2MouseClicked(java.awt.event.MouseEvent evt) {GEN-FIRST:event_tabel2MouseClicked
    // TODO add your handling code here:
} //GEN-LAST:event_tabel2MouseClicked

```

Metode **ttanggalActionPerformed** digunakan sebagai penanganan aksi saat pengguna melakukan tindakan tertentu pada ttanggal, namun belum diimplementasikan lebih lanjut. Metode **tabelMouseClicked** digunakan untuk mengatur respons saat pengguna melakukan klik pada tabel1, namun juga belum diimplementasikan.

Metode **bayarKeyReleased** akan dijalankan setiap kali pengguna melepaskan tombol pada bayar. Metode ini digunakan untuk menghitung kembalian berdasarkan total

harga dan jumlah uang yang dibayarkan. Jika jumlah uang yang dibayarkan kurang dari total harga, maka kembalian akan ditampilkan sebagai "Uang Kurang!".

Metode **selesaiActionPerformed** akan dijalankan ketika pengguna menekan tombol "Selesai". Metode ini digunakan untuk menyimpan data pesanan ke dalam database dan menampilkan pesan berhasil jika proses berhasil. Jika jumlah uang yang dibayarkan kurang dari total harga, maka akan ditampilkan pesan "Uang Anda Kurang". Setelah proses selesai, data akan ditampilkan ulang, form akan direset, dan laporan struk akan ditampilkan menggunakan browser bawaan sistem.

Metode **kembaliActionPerformed** akan dijalankan ketika pengguna menekan tombol "Kembali". Metode ini digunakan untuk mengarahkan pengguna kembali ke halaman utama dengan memanggil objek Home. Metode **ReportActionPerformed** akan dijalankan ketika pengguna menekan tombol "Report". Metode ini digunakan untuk menampilkan laporan dalam bentuk web menggunakan browser bawaan sistem.

Metode **ReportMouseClicked** digunakan untuk mengatur respons saat pengguna melakukan klik pada Report, namun belum diimplementasikan. Metode **tabel2MouseClicked** digunakan untuk mengatur respons saat pengguna melakukan klik pada tabel2, namun belum diimplementasikan.

```

private void no_transaksiKeyReleased(java.awt.event.KeyEvent evt) {GEN-FIRST:event_no_transaksiKeyReleased
    if (!no_transaksi.getText().equals("")) {
        try {
            st = con.createStatement();
            rs = st.executeQuery("select * from tbl_pesan where no_antrian='"+no_transaksi.getText()+"'");
            if (rs.next()) {
                nama.setText(rs.getString("nama"));
                no_telp.setText(rs.getString("no_telp"));
                alamat.setText(rs.getString("alamat"));
                j_pewangi.setSelectedItem(rs.getString("j_pewangi"));
                jumlah.setText(rs.getString("jumlah"));
                harga.setText(rs.getString("harga"));
                ttanggal.setText(rs.getString("tanggal"));
                totall.setText(rs.getString("harga"));
            }else{
                nama.setText("");
                no_telp.setText("");
                alamat.setText("");
                j_pewangi.setSelectedItem("");
                jumlah.setText("");
                harga.setText("");
                ttanggal.setText("");
                totall.setText("");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}GEN-LAST:event_no_transaksiKeyReleased

private void jumlahActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_jumlahActionPerformed
    // TODO add your handling code here:
}GEN-LAST:event_jumlahActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Pembayaran.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Pembayaran.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Pembayaran.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Pembayaran.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                new Pembayaran().setVisible(true);
            } catch (SQLException ex) {
                Logger.getLogger(Pembayaran.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    });
}

```

Metode **no_transaksiKeyReleased** akan dijalankan setiap kali pengguna melepaskan tombol pada no_transaksi. Metode ini digunakan untuk mengambil data pesanan dari database berdasarkan nomor transaksi yang diinputkan. Jika data ditemukan, maka data akan ditampilkan pada komponen-komponen yang sesuai seperti nama, no_telp, alamat, j_pewangi, jumlah, harga, ttanggal, dan totall. Jika data tidak ditemukan, maka komponen-komponen tersebut akan dikosongkan.

Metode **jumlahActionPerformed** digunakan untuk mengatur respons saat pengguna melakukan aksi tertentu pada jumlah. Namun, belum diimplementasikan lebih lanjut. Di dalam metode main, tampilan dan gaya tampilan aplikasi diatur menggunakan UIManager dengan menggunakan tampilan "Nimbus". Kemudian, objek Pembayaran dibuat dan ditampilkan.

```
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton Report;
private javax.swing.JTextField alamat;
private javax.swing.JTextField bayar;
private javax.swing.JTextField harga;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JComboBox<String> j_pewangi;
private javax.swing.JTextField jumlah;
private javax.swing.JButton kembali;
private javax.swing.JTextField kembalian;
private javax.swing.JTextField nama;
private javax.swing.JTextField no_telp;
private javax.swing.JTextField no_transaksi;
private javax.swing.JButton selesai;
private javax.swing.JTable tabel1;
private javax.swing.JTable tabel2;
private javax.swing.JLabel tgl;
private javax.swing.JTextField totall;
private javax.swing.JTextField ttanggal;
// End of variables declaration//GEN-END:variables
```

Beberapa komponen yang dideklarasikan antara lain adalah tombol Report, serta berbagai jenis JTextField untuk memasukkan data seperti alamat, bayar, harga, jumlah, nama, no_telp, no_transaksi, kembalian, totall, dan ttanggal. Terdapat juga JComboBox dengan nama j_pewangi yang digunakan untuk memilih jenis pewangi.

Selain itu, terdapat juga beberapa label (JLabel) yang digunakan untuk menampilkan teks atau informasi seperti jLabel2, jLabel5, jLabel6, jLabel7, jLabel8, jLabel9, jLabel10, jLabel11, jLabel12, jLabel13, jLabel14, dan jLabel15.

Terdapat pula dua tabel (JTable) dengan nama tabel1 dan tabel2 yang digunakan untuk menampilkan data dalam bentuk tabel. Komponen-komponen ini ditempatkan dalam dua panel (jPanel1 dan jPanel2) yang digunakan untuk mengatur tata letak komponen. Variabel-variabel tersebut dideklarasikan sebagai private, sehingga hanya dapat diakses di dalam kelas yang sama.

5.4. Listing Pemesanan

```
package laundry;

import java.sql.Connection;
import java.sql.Statement;
import java.awt.Dimension;
import java.awt.HeadlessException;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Date;
import java.util.Locale;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.Timer;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import javax.swing.text.DateFormatter;

/**
 *
 * @author LENOVO
 */
public class Pemesanan extends javax.swing.JFrame {

    private static final DateTimeFormatter smpdtfmt = DateTimeFormatter.ofPattern("yyyy/MM/dd");
    LocalDateTime tglsekarang = LocalDateTime.now();
    private final String ltanggal = smpdtfmt.format(tglsekarang);
    private Statement st;
    private Connection con;
    private ResultSet rs;
    public DefaultTableModel tabmodel;
    public long total;
    public long pewangi;
    private Object displaytext;

    private void koneksi() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = (Connection) DriverManager.getConnection("jdbc:mysql://localhost/db_laundry", "root", "");
            st = (Statement) con.createStatement();
        } catch (ClassNotFoundException | SQLException e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    int harga_pewangi = 0;

    /**
     * Creates new form pelanggan
     */
    public Pemesanan() {
        initComponents();
        koneksi();
        judul();
        tampildata();
        Date date = new Date();
        get_antrian();

        reset();
        auto();
        // Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = getSize();
        setLocation(
            (screenSize.width - frameSize.width) / 2,
            (screenSize.height - frameSize.height) / 2);
    }
}
```

Kelas Pemesanan merupakan kelas utama yang mewarisi sifat-sifat dari kelas `javax.swing.JFrame` sehingga dapat digunakan untuk membuat tampilan antarmuka pengguna (UI) aplikasi. Di dalam konstruktor kelas Pemesanan, beberapa inisialisasi dilakukan, seperti mengatur koneksi ke database dengan menggunakan driver MySQL, username, dan password yang sesuai. Kemudian, fungsi `koneksi()` dipanggil untuk membuat koneksi dengan database.

Selain itu, terdapat juga beberapa variabel dan objek yang dideklarasikan, seperti `st` untuk objek `Statement`, `con` untuk objek `Connection`, `rs` untuk objek `ResultSet`, `tabmodel` untuk objek `DefaultTableModel`, dan sebagainya. Kelas Pemesanan juga memiliki beberapa fungsi dan metode, seperti `judul()` untuk mengatur judul kolom dalam tabel, `tampildata()` untuk menampilkan data dalam tabel, dan `reset()` untuk mengosongkan atau mereset nilai-nilai pada komponen-komponen antarmuka.

Terakhir, terdapat pemanggilan fungsi-fungsi dan konstruktor lainnya, serta pengaturan tampilan frame aplikasi agar berada di tengah layar. Kode ini merupakan inti dari fitur pemesanan pada aplikasi Laundry dan berperan penting dalam mengatur interaksi dengan pengguna serta pengelolaan data pemesanan.


```

public void get_antrian() {
    try {
        st = con.createStatement();
        rs = st.executeQuery("select max(no_antrian) from tbl_pesan");
        int jumlah = rs.getRow() + 1;
        no_transaksi.setText(""+jumlah);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void tampildata() {
    try {
        st = con.createStatement();
        tabmodel.getDataVector().removeAllElements();
        tabmodel.fireTableDataChanged();
        rs = st.executeQuery("SELECT * FROM tbl_pesan");
        while (rs.next()) {
            Object[] data = {
                rs.getString("no_antrian"),
                rs.getString("nama"),
                rs.getString("no_telp"),
                rs.getString("alamat"),
                rs.getString("j_pewangi"),
                rs.getString("jumlah"),
                rs.getString("harga"),
                rs.getString("tanggal")
            };
            tabmodel.addRow(data);
        }
        tbl_beli.setModel(tabmodel);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

private void judul() {
    Object[] judul = {"Nomor Antrian", "Nama", "Nomor Telepon", "Alamat", "Jenis_pewangi", "Jumlah", "Harga",
        "Tanggal"};
    tabmodel = new DefaultTableModel(null, judul);
    tbl_beli.setModel(tabmodel);
}

private void reset() {
    no_transaksi.setText("");
    nama.setText("");
    jumlah.setText("");
    j_pewangi.setSelectedIndex(0);
    nama.setText("");
    no_telp.setText("");
    alamat.setText("");
    tottal.setText("");
    no_transaksi.requestFocus();
    simpan.setEnabled(true);
    hapus.setEnabled(false);
}

public void FilterAngka(KeyEvent b) {
    if (Character.isAlphabetic(b.getKeyChar())) {
        b.consume();
        JOptionPane.showMessageDialog(null, "Masukkan angka saja!", "Peringatan", JOptionPane.WARNING_MESSAGE);
    }
}

public void FilterHuruf(KeyEvent a) {
    if (Character.isDigit(a.getKeyChar())) {
        a.consume();
        JOptionPane.showMessageDialog(null, "Masukkan huruf saja!", "Peringatan", JOptionPane.WARNING_MESSAGE);
    }
}
}

```

Fungsi **get_antrian()** digunakan untuk mendapatkan nomor antrian terakhir dari tabel **tbl_pesan** dalam database. Fungsi ini mengambil nilai maksimum dari kolom **no_antrian** dan menambahkannya dengan 1. Hasilnya kemudian ditampilkan pada komponen **no_transaksi**.

Fungsi **tampildata()** digunakan untuk menampilkan data pemesanan dalam tabel. Fungsi ini mengambil data dari tabel **tbl_pesanan** dalam database dan menampilkannya dalam tabel **tbl_beli**. Seluruh data dalam tabel **tbl_beli** akan dihapus terlebih dahulu menggunakan **tabmodel.getDataVector().removeAllElements()** untuk menghindari duplikasi data, kemudian data baru ditambahkan ke dalam tabel.

Fungsi **judul()** digunakan untuk mengatur judul kolom dalam tabel **tbl_beli**. Judul kolom diinisialisasi dalam array **judul** dan diset pada objek **tabmodel**. Fungsi **reset()** digunakan untuk mengosongkan atau mereset nilai-nilai pada komponen-komponen antarmuka. Fungsi ini akan mengatur nilai-nilai komponen keadaan awal, seperti mengosongkan teks pada **no_transaksi**, **nama**, **jumlah**, dan lain-lain. Beberapa komponen juga diatur keadaan enabled atau disabled, seperti **simpan** yang diaktifkan dan **hapus** yang dinonaktifkan.

Terakhir, terdapat dua fungsi **FilterAngka()** dan **FilterHuruf()** yang digunakan untuk melakukan filter inputan pada event **KeyEvent**. Fungsi **FilterAngka()** digunakan untuk membatasi penggunaan karakter non-angka pada input, sementara **FilterHuruf()** digunakan untuk membatasi penggunaan karakter non-huruf pada input. Jika karakter yang tidak sesuai ditemukan, maka akan ditampilkan pesan peringatan.

```
private void simpanActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_simpanActionPerformed
    if(j_pewangi.equals("") || jumlah.getText().equals("") || nama.equals("") || no_telp.equals("") ||
    alamat.equals("") || total.equals("")){
        JOptionPane.showMessageDialog(null, "Data perlu diisi!");
    }else{
        String sql = "no_antrian=NULL,nama='"+nama.getText()+"',no_telp='"+no_telp.getText()+"',alamat='"+
        alamat.getText()+"',j_pewangi='"+j_pewangi.getSelectedItem()+"',jumlah='"+jumlah.getText()+"',harga='"+total
        .getText()+"',tanggal='"+smpdfmt.format(tglsekarang).toString()+"'";
        Model model = new Model();
        if (model.simpan("tbl_pesanan", sql)) {
            reset();
            tampildata();
            JOptionPane.showMessageDialog(null, "Berhasil Menyimpan Data");
        }
    }
}

}GEN-LAST:event_simpanActionPerformed

private void hapusActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_hapusActionPerformed
    // TODO add your handling code here:
    try {
        st.executeUpdate("delete from tbl_pesanan where " + "no_antrian=" + no_transaksi.getText() + "");
        reset();
        JOptionPane.showMessageDialog(null, "Berhasil");
    } catch (HeadlessException e) {
        JOptionPane.showMessageDialog(null, "pesan salah : " + e);
    } catch (SQLException ex) {
        Logger.getLogger(Pemesanan.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        tampildata();
    }
}

}GEN-LAST:event_hapusActionPerformed
```

Pada event handler **simpanActionPerformed**, terlebih dahulu dilakukan validasi apakah semua data telah diisi. Jika ada data yang kosong, maka akan ditampilkan pesan peringatan bahwa data perlu diisi. Jika semua data telah diisi, maka dilakukan penyimpanan data ke dalam database dengan menggunakan objek **Model** dan method **simpan()**. Data yang akan disimpan diambil dari komponen-komponen antarmuka seperti **nama**, **no_telp**, **alamat**, **j_pewangi**, **jumlah**, **tottal**, dan **tglsekarang**. Setelah proses penyimpanan berhasil, komponen-komponen antarmuka dikosongkan menggunakan **reset()** dan data dalam tabel diperbarui dengan memanggil **tampildata()**. Terakhir, ditampilkan pesan sukses bahwa data berhasil disimpan.

Pada event handler **hapusActionPerformed**, terlebih dahulu dilakukan penghapusan data dari tabel **tbl_pesan** berdasarkan nomor antrian yang terdapat pada komponen **no_transaksi**. Jika proses penghapusan berhasil, maka komponen-komponen antarmuka dikosongkan menggunakan **reset()** dan ditampilkan pesan sukses. Jika terjadi kesalahan dalam proses penghapusan, ditampilkan pesan error. Setelah itu, data dalam tabel diperbarui dengan memanggil **tampildata()**.

```
private void selesaiActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_selesaiActionPerformed
    // TODO add your handling code here:
    new Home().setVisible(true);
    dispose();
} //GEN-LAST:event_selesaiActionPerformed

private void resetActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_resetActionPerformed
    // TODO add your handling code here:
    reset();
} //GEN-LAST:event_resetActionPerformed

private void formWindowOpened(java.awt.event.WindowEvent evt) {GEN-FIRST:event_formWindowOpened
    tampildata();
    no_transaksi.disable();
} //GEN-LAST:event_formWindowOpened

private void formWindowActivated(java.awt.event.WindowEvent evt) {GEN-FIRST:event_formWindowActivated
    // TODO add your handling code here:
} //GEN-LAST:event_formWindowActivated

private void no_transaksiActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_no_transaksiActionPerformed
    get_antrian();
} //GEN-LAST:event_no_transaksiActionPerformed

private void tottalActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_tottalActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_tottalActionPerformed

private void jumlahActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_jumlahActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_jumlahActionPerformed

private void hitungActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_hitungActionPerformed
    if (j_pewangi.getSelectedIndex() == 0) {
        JOptionPane.showMessageDialog(null, "Isi data!");
    } else if (j_pewangi.getSelectedIndex() == 1) {
        harga_pewangi = 5000;
    } else if (j_pewangi.getSelectedIndex() == 2) {
        harga_pewangi = 3000;
    }
    int totall = 0;
    totall = Integer.parseInt(String.valueOf(jumlah.getText()));
    int total = (int) ((totall * 10000) + harga_pewangi);
    tottal.setText(String.valueOf(total));
} //GEN-LAST:event_hitungActionPerformed
```

Pada event handler **selesaiActionPerformed**, ketika tombol "Selesai" di klik, antarmuka akan menutup jendela saat ini dan membuka jendela **Home** yang merupakan halaman utama aplikasi Laundry. Pada event handler **resetActionPerformed**, ketika tombol "Reset" di klik, fungsi **reset()** akan dipanggil untuk mengosongkan nilai dari beberapa komponen antarmuka.

Pada event handler **formWindowOpened**, ketika jendela aplikasi dibuka, fungsi **tampildata()** akan dipanggil untuk menampilkan data pada tabel **tbl_pesan**. Selain itu, komponen **no_transaksi** akan dinonaktifkan agar tidak dapat diubah oleh pengguna. Pada event handler **formWindowActivated**, ketika jendela aplikasi diaktifkan, tidak ada kode yang dijalankan.

Pada event handler **no_transaksiActionPerformed**, ketika nilai pada komponen **no_transaksi** diubah atau diketik, fungsi **get_antrian()** akan dipanggil untuk mendapatkan nomor antrian terbaru. Pada event handler **tottalActionPerformed**, ketika nilai pada komponen **tottal** diubah atau diketik, tidak ada kode yang dijalankan. Pada event handler **jumlahActionPerformed**, ketika nilai pada komponen **jumlah** diubah atau diketik, tidak ada kode yang dijalankan.

Pada event handler **hitungActionPerformed**, ketika tombol "Hitung" di klik, dilakukan pengecekan pada nilai yang dipilih pada komponen **j_pewangi**. Jika nilainya adalah 0, maka akan ditampilkan pesan bahwa data perlu diisi. Jika nilainya adalah 1, maka variabel **harga_pewangi** akan diatur menjadi 5000. Jika nilainya adalah 2, maka **harga_pewangi** akan diatur menjadi 3000. Kemudian, dilakukan perhitungan total dengan mengambil nilai dari komponen **jumlah**, mengalikannya dengan 10000, dan menambahkannya dengan **harga_pewangi**. Hasil perhitungan tersebut ditampilkan pada komponen **tottal**.

```

private void j_pewangiActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_j_pewangiActionPerformed
    // TODO add your handling code here:
}GEN-LAST:event_j_pewangiActionPerformed

private void j_pewangiItemStateChanged(java.awt.event.ItemEvent evt) {GEN-FIRST:event_j_pewangiItemStateChanged
    // TODO add your handling code here:
}GEN-LAST:event_j_pewangiItemStateChanged

private void no_telpActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_no_telpActionPerformed
    // TODO add your handling code here:
}GEN-LAST:event_no_telpActionPerformed

private void namaKeyTyped(java.awt.event.KeyEvent evt) {GEN-FIRST:event_namaKeyTyped
    // TODO add your handling code here:
    FilterHuruf(evt);
}GEN-LAST:event_namaKeyTyped

private void namaActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_namaActionPerformed
    // TODO add your handling code here:
}GEN-LAST:event_namaActionPerformed

private void tbl_beliMouseClicked(java.awt.event.MouseEvent evt) {GEN-FIRST:event_tbl_beliMouseClicked
    // TODO add your handling code here:

    no_transaksi.setText(tabmodel.getValueAt(tbl_beli.getSelectedRow(), 0) + "");
    j_pewangi.setSelectedItem(tabmodel.getValueAt(tbl_beli.getSelectedRow(), 4) + "");
    jumlah.setText(tabmodel.getValueAt(tbl_beli.getSelectedRow(), 5) + "");
    nama.setText(tabmodel.getValueAt(tbl_beli.getSelectedRow(), 1) + "");
    no_telp.setText(tabmodel.getValueAt(tbl_beli.getSelectedRow(), 2) + "");
    alamat.setText(tabmodel.getValueAt(tbl_beli.getSelectedRow(), 3) + "");
    tottal.setText(tabmodel.getValueAt(tbl_beli.getSelectedRow(), 6) + "");

    simpan.setEnabled(false);
    hapus.setEnabled(true);
}GEN-LAST:event_tbl_beliMouseClicked

private void alamatActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_alamatActionPerformed
    // TODO add your handling code here:
}GEN-LAST:event_alamatActionPerformed

```

Pada event handler **j_pewangiActionPerformed**, ketika nilai pada komponen **j_pewangi** diubah, tidak ada kode yang dijalankan. Pada event handler **j_pewangiItemStateChanged**, ketika nilai pada komponen **j_pewangi** berubah, tidak ada kode yang dijalankan. Pada event handler **no_telpActionPerformed**, ketika nilai pada komponen **no_telp** diubah atau diketik, tidak ada kode yang dijalankan.

Pada event handler **namaKeyTyped**, ketika nilai pada komponen **nama** diubah atau diketik, fungsi **FilterHuruf()** akan dipanggil untuk memfilter masukan pengguna sehingga hanya huruf yang diperbolehkan. Pada event handler **namaActionPerformed**, ketika nilai pada komponen **nama** diubah atau diketik, tidak ada kode yang dijalankan.

Pada event handler **tbl_beliMouseClicked**, ketika pengguna mengklik salah satu baris pada tabel **tbl_beli**, nilai-nilai pada baris yang dipilih akan diambil dan ditampilkan pada komponen-komponen terkait seperti **no_transaksi**, **j_pewangi**, **jumlah**, **nama**, **no_telp**, **alamat**, dan **tottal**. Selain itu, tombol **simpan** akan dinonaktifkan dan tombol **hapus** akan diaktifkan.

Pada event handler **alamatActionPerformed**, ketika nilai pada komponen **alamat** diubah atau diketik, tidak ada kode yang dijalankan. Event-handler tersebut berfungsi

untuk mengatur respons dan aksi yang dilakukan saat terjadi perubahan atau interaksi dengan komponen-komponen tersebut dalam antarmuka aplikasi.

```
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Pemesanan.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Pemesanan.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Pemesanan.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Pemesanan.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Pemesanan().setVisible(true);
        }
    });
}
```

Pertama, kode mencoba untuk mengatur tampilan antarmuka grafis dengan menggunakan look and feel "Nimbus" jika tersedia. Jika berhasil, tampilan antarmuka grafis akan disesuaikan dengan tampilan Nimbus. Jika tidak tersedia, maka tampilan antarmuka akan menggunakan tampilan default. Jika terdapat kesalahan dalam mengatur look and feel, informasi tentang kesalahan tersebut akan dicatat dalam log.

Selanjutnya, dalam metode **main**, dibuat sebuah objek dari kelas **Pemesanan** dan objek tersebut ditampilkan ke layar dengan memanggil metode **setVisible(true)**. Dengan demikian, ketika program dijalankan, antarmuka aplikasi **Pemesanan** akan muncul dan siap digunakan oleh pengguna.

```

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JTextField alamat;
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JButton hapus;
private javax.swing.JButton hitung;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JComboBox<String> j_pewangi;
private javax.swing.JTextField jumlah;
private javax.swing.JTextField nama;
private javax.swing.JTextField no_telp;
private javax.swing.JTextField no_transaksi;
private javax.swing.JButton reset;
private javax.swing.JButton selesai;
private javax.swing.JButton simpan;
private javax.swing.JTable tbl_beli;
private javax.swing.JTextField tottal;
// End of variables declaration//GEN-END:variables

private void jenis_pakaian() {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools
    Templates.
}

private void autokd_transaksi() {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools
    Templates.
}

private void autokd_pelanggan() {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools
    Templates.
}

private static class tottal {

    private static void setText(String string) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose
        Tools | Templates.
    }

    public tottal() {
    }

}
}

```

Setiap variabel diberi tipe data dan nama yang sesuai dengan komponen yang mereka wakili dalam antarmuka. Variabel tersebut digunakan untuk mengakses dan memanipulasi komponen-komponen tersebut melalui kode program.

Misalnya, **alamat**, **nama**, **no_telp** adalah variabel bertipe **JTextField** yang mewakili teks field untuk memasukkan alamat, nama, dan nomor telepon dalam antarmuka. Selain variabel-variabel komponen, terdapat juga variabel **buttonGroup1** yang digunakan untuk mengelompokkan tombol-tombol secara logis. Seluruh variabel tersebut dideklarasikan dan diinisialisasi dalam blok "Variables declaration - do not modify" dan diakhiri dengan komentar "// End of variables declaration".

Selain itu, terdapat juga metode-metode yang belum diimplementasikan seperti **jenis_pakaian()**, **autokd_transaksi()**, dan **autokd_pelanggan()**. Implementasi metode-metode ini masih perlu ditambahkan untuk mengatur fungsi-fungsi yang sesuai dalam

program. Dengan menggunakan variabel-variabel ini, komponen-komponen antarmuka dapat diakses dan diperbarui sesuai dengan kebutuhan program.

5.5. Output Halaman Login

5.6. Output Halaman Home



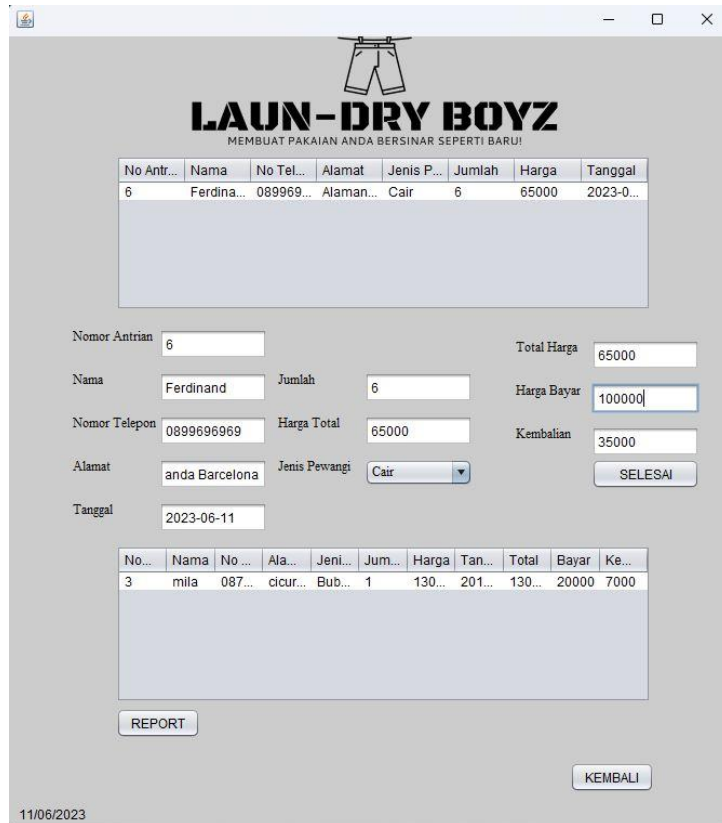
Pada halaman home terdapat tombol pesan dan pengambilan. Pada pemesanan yaitu untuk menggunakan layanan laundry dan pada Pengambilan yaitu untuk melakukan pengambilan barang pelanggan sekaligus untuk mengetahui total membayar layanan laundry.

5.7. Output Halaman Pemesanan

Nomor A...	Nama	Nomor T...	Alamat	Jenis_p...	Jumlah	Harga	Tanggal
------------	------	------------	--------	------------	--------	-------	---------

Pada bagian pemesanan terdapat nama pelanggan, nomor telepon, alamat, Jenis Pewangi, jumlah pakaian dan Harganya. Pada jenis pewangi harganya berbeda, lalu pada jenis pewangi bentuk cairan dikenakan harga 3.000/kg dan harga jenis pewangi bentuk bubuk dikenakan harga 5.000/kg. Setelah itu tombol Hitung berfungsi untuk perhitungan otomatis. Setelah itu jika semua sudah sesuai dengan keinginan pelanggan, lalu klik simpan dan data akan tertampil di tabel.

5.8. Output Halaman Pengembalian



LAUN-DRY BOYZ
MEMBUAT PAKAIAN ANDA BERSINAR SEPerti BARU!

No Antr...	Nama	No Tel...	Alamat	Jenis P...	Jumlah	Harga	Tanggal
6	Ferdina...	089969...	Alaman...	Cair	6	65000	2023-0...

Nomor Antrian: 6
 Nama: Ferdinand
 Nomor Telepon: 0899696969
 Alamat: anda Barcelona
 Tanggal: 2023-06-11
 Jumlah: 6
 Harga Total: 65000
 Jenis Pewangi: Cair
 Total Harga: 65000
 Harga Bayar: 100000
 Kembalian: 35000
 SELESAI

No...	Nama	No ...	Ala...	Jeni...	Jum...	Harga	Tan...	Total	Bayar	Ke...
3	mila	087...	cicur...	Bub...	1	130...	201...	130...	20000	7000

REPORT

KEMBALI

11/06/2023

Pada pengembalian hanya cukup memasukkan Nomor Antrian pelanggan maka akan otomatis mengisi data pelanggan yang sebelumnya sudah di masukkan datanya di halaman pemesanan. Jadi admin Laundry Boyz hanya memasukkan jumlah nominal harga bayar yang di berikan oleh pelanggan dan akan tertera otomatis kembaliannya, apabila nominal uang yang di kasih pelanggan lebih dari total harga. Terakhir jika sudah benar semua maka klik tombol SELESAI dan akan tampil seperti gambar dibawah.




LAUN-DRY BOYZ
Jl. Kartawijaya, Kota Bekasi

Nama : Ferdinand
Tanggal : Rp 2023-06-11
Total : Rp 65000
Bayar : Rp 100000
Kembalian : Rp 35000

Selamat Datang Kembali Boyz

© 2023 LAUN-DRY BOYZ

5.9. Output Halaman Laporan (*Report*)



Laporan Laun-Dry Boyz
Jl. Kartawijaya, Kota Bekasi

No Antrian	Nama	No Telepon	Alamat	Jenis Pewangi	Jumlah	Harga	Tanggal	Total	Bayar	Kembalian
3	mila	087770432050	cicurug	Bubuk	1	13000	2018-05-26	13000	20000	7000
6	Ferdinand	0899696969	Alamanda Barcelona	Cair	6	65000	2023-06-11	65000	100000	35000

© 2023 LAUN-DRY BOYZ

Pada halaman ini berfungsi sebagai seluruh laporan pelanggan yang menggunakan layanan Laundry Boys.

BAB IV

PENUTUP

1. KESIMPULAN

Pemrograman berbasis objek adalah pendekatan yang kuat untuk membangun aplikasi yang kompleks dan mudah dikelola. Dalam konteks pembuatan aplikasi Laundry Boyz dengan menggunakan aplikasi NetBeans, penggunaan pemrograman berbasis objek memberikan banyak manfaat. Dalam pengembangan aplikasi ini, penggunaan NetBeans sebagai lingkungan pengembangan terintegrasi (IDE) juga memberikan keuntungan yang signifikan.

Salah satu keuntungan utama dari pendekatan pemrograman berbasis objek adalah kemampuannya untuk mengorganisir dan mengelompokkan kode menjadi objek yang memiliki atribut dan perilaku terkait. Dalam konteks Laundry Boyz, ini memungkinkan pengembang untuk membuat kelas yang merepresentasikan entitas seperti pelanggan, pesanan, dan item laundry. Objek-objek ini dapat saling berinteraksi melalui metode yang didefinisikan di dalam kelas-kelas tersebut, sehingga memudahkan pengelolaan dan pemeliharaan aplikasi.

Selain itu, penggunaan aplikasi NetBeans sebagai IDE juga memberikan manfaat yang signifikan. NetBeans menyediakan alat-alat pengembangan yang kuat, seperti editor kode yang canggih, penyusun tata letak antarmuka pengguna (GUI), dan debugger yang membantu dalam mengembangkan dan menguji aplikasi dengan lebih efisien. NetBeans juga mendukung berbagai bahasa pemrograman, termasuk Java yang umum digunakan dalam pemrograman berbasis objek.

2. SARAN

Berikut adalah beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut aplikasi Laundry Boyz dengan menggunakan pemrograman berbasis objek dengan aplikasi NetBeans

1. Perluasan Fitur yaitu melakukan analisis kebutuhan pengguna dan menambahkan fitur - fitur tambahan yang dapat meningkatkan fungsionalitas aplikasi, seperti integrasi dengan pembayaran online, sistem pengiriman, atau notifikasi pelanggan.
2. Pengoptimalan Kinerja yaitu melakukan peninjauan kembali kode aplikasi untuk memastikan bahwa tidak ada kerugian kinerja atau bottleneck yang dapat mempengaruhi

kecepatan atau responsivitas aplikasi. Mengoptimalkan dan memperbaiki bagian-bagian yang membutuhkan perbaikan untuk meningkatkan kinerja keseluruhan aplikasi.

3. Peningkatan Keamanan yaitu melakukan audit keamanan terhadap aplikasi untuk mengidentifikasi dan mengatasi kerentanan potensial. Mengimplementasikan mekanisme keamanan tambahan, seperti enkripsi data pengguna, autentikasi ganda, atau kontrol akses yang lebih baik untuk melindungi informasi sensitif dan mencegah akses yang tidak sah.
4. Pengujian dan Pemeliharaan yaitu melakukan pengujian secara menyeluruh untuk memastikan bahwa aplikasi berfungsi dengan baik dalam berbagai skenario penggunaan. Selain itu, menjadwalkan pemeliharaan rutin dan pembaruan perangkat lunak untuk memperbaiki bug, mengatasi masalah keamanan, dan menjaga kualitas aplikasi seiring waktu.
5. Pengembangan Lintas Platform yaitu mempertimbangkan untuk mengembangkan versi aplikasi Laundry Boyz yang dapat berjalan pada platform lain, seperti aplikasi seluler (Android atau iOS) atau aplikasi web. Hal ini akan memperluas jangkauan pengguna dan meningkatkan aksesibilitas aplikasi.

DAFTAR PUSTAKA

<https://www.dicoding.com/blog/apa-itu-uml/>