

**LAPORAN PROYEK UJIAN AKHIR SEMESTER
PEMROGRAMAN BERORIENTASI OBJEK
PROGRAM PENJUALAN SEMBAKO**



Dosen Pembimbing :

Taufik Ridwan, M.T.

Disusun oleh kelompok 5:

Ajriel Saputra	(2310631250004)
Chika Putri Pratama	(2310631250010)
Saut Parulian Simatupang	(2310631250077)
George David Josef Calvin Sirait	(2310631250091)

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SINGAPERBANGSA KARAWANG
TAHUN 2025**

A. LATAR BELAKANG

Kebutuhan akan bahan pokok seperti beras, gula, minyak goreng, dan kebutuhan rumah tangga lainnya menjadi hal yang sangat vital dalam kehidupan masyarakat sehari-hari. Toko sembako sebagai penyedia utama kebutuhan tersebut memiliki peranan penting dalam menjaga kelangsungan distribusi bahan pokok di lingkungan masyarakat. Namun, dalam prakteknya, banyak toko sembako masih menggunakan metode konvensional dalam pengelolaan transaksi dan pencatatan stok barang. Proses manual ini rentan terhadap kesalahan manusia (*human error*), seperti salah mencatat harga, lupa mengurangi stok, atau kesulitan dalam menelusuri histori transaksi pelanggan.

Seiring dengan perkembangan teknologi informasi, penggunaan sistem komputerisasi dalam sektor ritel telah menjadi kebutuhan yang tidak terelakkan. Sistem informasi penjualan dan pemesanan berbasis aplikasi menjadi solusi yang efisien dan tepat guna untuk menunjang kinerja operasional toko, mempercepat pelayanan, dan meningkatkan kepuasan pelanggan. Sistem ini juga memungkinkan pengelola toko untuk melakukan monitoring stok secara real-time, mencetak struk sebagai bukti transaksi, serta mencatat data pelanggan dan status pembayaran secara otomatis.

Dalam konteks ini, dibutuhkan suatu sistem penjualan sembako yang terintegrasi, berbasis desktop, yang dapat membantu pelanggan dalam memilih barang, melakukan pemesanan, dan menyelesaikan pembayaran dengan metode *cash*. Di sisi lain, sistem ini juga membantu kasir dan admin toko dalam mengelola data transaksi, stok barang, dan pencatatan status pembayaran dengan akurat. Aplikasi ini diharapkan tidak hanya memberikan kemudahan dan efisiensi, tetapi juga meningkatkan profesionalisme toko dalam pelayanan dan pencatatan transaksi.

A. RUMUSAN MASALAH DAN TUJUAN

1. Rumusan Masalah

- 1) Bagaimana merancang dan mengembangkan sistem aplikasi desktop untuk penjualan sembako yang mendukung pemesanan barang, pengelolaan stok, serta proses pembayaran tunai (*cash*) secara efisien?
- 2) Bagaimana konsep *Object-Oriented Programming* (OOP) seperti enkapsulasi, pewarisan, dan polimorfisme dapat diterapkan dalam pengembangan kode program sistem ini?
- 3) Bagaimana menggambarkan struktur dan relasi antar komponen sistem menggunakan pemodelan UML, khususnya use case, sebagai dasar perancangan sistem?
- 4) Bagaimana proses implementasi sistem dengan bahasa pemrograman Java dan pustaka GUI Java Swing dapat dilakukan sesuai desain yang telah dirancang?
- 5) Bagaimana menguji fungsionalitas sistem agar seluruh fitur seperti pemesanan, pengurangan stok, pembayaran, dan pencetakan struk berjalan dengan baik sesuai kebutuhan pengguna?

2. Tujuan

- 1) Mengembangkan aplikasi penjualan sembako berbasis desktop menggunakan Java Swing, yang menyediakan fitur pemesanan barang, pengelolaan stok, dan pembayaran secara *cash*.
- 2) Menerapkan prinsip-prinsip *Object-Oriented Programming* dalam pengembangan sistem untuk menghasilkan kode program yang modular, mudah dikembangkan, dan terstruktur.
- 3) Merancang *use case* menggunakan Unified Modeling Language (UML) untuk mendokumentasikan struktur dan relasi antar kelas dalam sistem sebagai acuan implementasi.
- 4) Mewujudkan implementasi sistem berdasarkan rancangan UML dan menerapkannya dalam kode program Java sesuai dengan alur kerja sistem yang direncanakan.
- 5) Melakukan pengujian terhadap fitur-fitur utama sistem untuk memastikan sistem berjalan sesuai kebutuhan dan dapat digunakan oleh pengguna akhir (admin dan pelanggan) secara efektif.

B. FITUR-FITUR APLIKASI

1. Login admin dan akses role tertentu
Sistem menerapkan kontrol akses berbasis peran (role-based access) di mana hanya admin yang dapat melakukan login melalui halaman khusus. Setelah berhasil login, admin akan diarahkan ke halaman utama admin yang menyediakan akses ke dua fitur utama: CRUD barang dan halaman kasir. Sementara itu, pelanggan tidak perlu login, melainkan langsung dapat mengakses halaman pemesanan barang.
2. CRUD Data Barang (Admin)
Fitur ini memungkinkan admin untuk mengelola data barang sembako secara langsung melalui antarmuka grafis yang intuitif:
 - 1) Tambah: Admin dapat memasukkan nama barang, harga satuan, dan stok awal.
 - 2) Ubah: Admin dapat memperbarui data barang yang sudah ada, seperti mengganti harga atau menyesuaikan stok.
 - 3) Hapus: Data barang yang sudah tidak relevan dapat dihapus dari sistem.
 - 4) Tabel Barang: Daftar barang ditampilkan dalam JTable yang dinamis, menampilkan nama, harga, dan jumlah stok, serta langsung merefleksikan perubahan secara real-time.
3. Pemesanan Barang oleh Pelanggan
Pelanggan dapat langsung mengakses halaman pemesanan, memilih barang yang tersedia, dan menentukan jumlah pembelian. Sistem akan menghitung otomatis subtotal untuk setiap item berdasarkan jumlah dan harga, dan menampilkannya di tabel keranjang.
4. Keranjang Belanja
Semua item yang ditambahkan akan masuk ke keranjang belanja, ditampilkan dalam sebuah tabel. Pelanggan masih dapat:
 - Mengubah jumlah pembelian sebelum mencetak struk.
 - Menghapus item tertentu dari keranjang.
 - Melihat total harga keseluruhan yang diperbarui secara otomatis.
5. Metode Pembayaran: Cash
Sistem mendukung metode pembayaran:
 - Cash: Transaksi disimpan sebagai "Belum Lunas", dan pelanggan mendapatkan struk sementara. Pembayaran secara tunai dikonfirmasi oleh kasir melalui halaman khusus.
6. Struk Transaksi Otomatis
Setelah proses pembayaran dilakukan, sistem akan secara otomatis menampilkan dan mencetak struk transaksi, yang berisi:
 - Daftar barang yang dibeli
 - Jumlah dan harga per item
 - Total pembayaran

- Metode pembayaran yang dipilih
- Status pembayaran (Lunas / Belum Lunas)

7. Halaman Kasir untuk Konfirmasi Pembayaran Cash

Fitur ini digunakan oleh kasir atau admin untuk mengkonfirmasi transaksi dengan status "Belum Lunas". Halaman ini menampilkan daftar pesanan yang belum dibayar. Kasir dapat memilih pesanan, memasukkan jumlah uang yang dibayarkan oleh pelanggan, lalu sistem secara otomatis:

- Menghitung kembalian
- Mengkonfirmasi pembayaran
- Mengubah status transaksi menjadi "Lunas"

8. Pengurangan Stok Otomatis Setelah Pembayaran

Setelah transaksi dinyatakan selesai dan statusnya "Lunas", sistem akan otomatis mengurangi stok barang yang dibeli dari database. Pengurangan dilakukan secara akurat berdasarkan jumlah barang yang dibeli, sehingga stok tetap terjaga dan data tetap konsisten.

9. Navigasi Antarmuka dan Tombol Reset

Aplikasi menyediakan tombol-tombol navigasi seperti:

- Kembali: Untuk berpindah ke halaman sebelumnya, misalnya dari halaman kasir ke halaman utama admin.
- Reset: Untuk mengosongkan seluruh input atau keranjang, memberikan fleksibilitas kepada pengguna saat ingin memulai ulang proses pemesanan atau pengeditan data.

10. Tampilan User Friendly dan Terstruktur

Antarmuka pengguna (GUI) dibangun menggunakan Java Swing dengan struktur panel dan tata letak yang jelas. Elemen seperti tombol, tabel, dan form input diatur sedemikian rupa agar:

- Mudah dipahami oleh pengguna awam
- Menghindari kekacauan tampilan
- Meningkatkan efisiensi dan kecepatan penggunaan
- Tampilan juga sudah dipisahkan berdasarkan peran pengguna (admin, pelanggan, dan kasir), sesuai prinsip desain antarmuka berbasis peran.

C. PENJELASAN KONSEP OOP PADA KODE PROGRAM

Object-Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasi pada objek. Dalam OOP, program dibangun menggunakan kelas (class) dan objek (object) sebagai komponen utama. Dalam pengembangan aplikasi "Program Penjualan Sembako", seluruh fitur yang dijelaskan sebelumnya dibangun dengan menggunakan prinsip-prinsip OOP secara konsisten.

Berikut penjelasan dan contoh penerapan konsep OOP pada kode program :

1. Enkapsulasi

Enkapsulasi menyembunyikan detail implementasi suatu kelas dan hanya mengekspos bagian yang dibutuhkan. Ini biasanya dilakukan dengan menggunakan private untuk variabel dan menyediakan metode publik untuk akses.

Contoh

```
public class HalamanAdmin extends JFrame {  
    private JTable table;  
    private DefaultTableModel model;  
    private JTextField tfNama, tfHarga, tfStok;  
    private JLabel lblGambarPath;  
    private Connection conn;
```

Pada Variabel tfNama, tfHarga, dan lainnya disembunyikan dari luar kelas dengan akses modifier private, menunjukkan penerapan enkapsulasi untuk menjaga data tetap aman dari manipulasi langsung.

2. Pewarisan (Inheritance)

Inheritance memungkinkan suatu kelas untuk mewarisi atribut dan method dari kelas lain. Kata kunci yang digunakan adalah extends.

Contoh

```
public class Beranda extends JFrame {  
    public Beranda() {  
        setTitle(title: "Beranda");  
        setSize(width: 900, height: 600);  
        setLocationRelativeTo(c: null);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setResizable(resizable: false);
```

Class Beranda mewarisi dari JFrame (Class GUI bawaan Java). Ini berarti beranda dapat menggunakan semua method dari JFrame, seperti setSize(), setTitle() dan method lain yang ada dari class JFrame.

3. Polimorfisme (Polymorphism)

Polimorfisme memungkinkan satu method memiliki banyak bentuk, biasanya dengan override method dari class induk.

Contoh

```
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    if (bgImage != null) {
        g.drawImage(bgImage, x:0, y:0, getWidth(), getHeight(), this);
    }
}
```

Method `paintComponent(Graphics g)` pada gambar di atas adalah method yang di-*override* dari class `JPanel`. Method ini digunakan untuk menggambar background khusus pada panel, yaitu dengan menggambar objek gambar (`bgImage`) jika tidak bernilai null. Ini merupakan contoh polimorfisme dalam OOP, karena method `paintComponent()` yang sama memiliki perilaku berbeda saat di-*override* di subclass `BackgroundPanel`, dibandingkan implementasi default-nya di superclass `JPanel`.

4. Exception Handling

Exception handling memungkinkan program menangani error atau kondisi tak terduga secara elegan tanpa menyebabkan program berhenti secara tiba-tiba. Kata kunci yang digunakan adalah `try`, `catch`, dan terkadang `finally`.

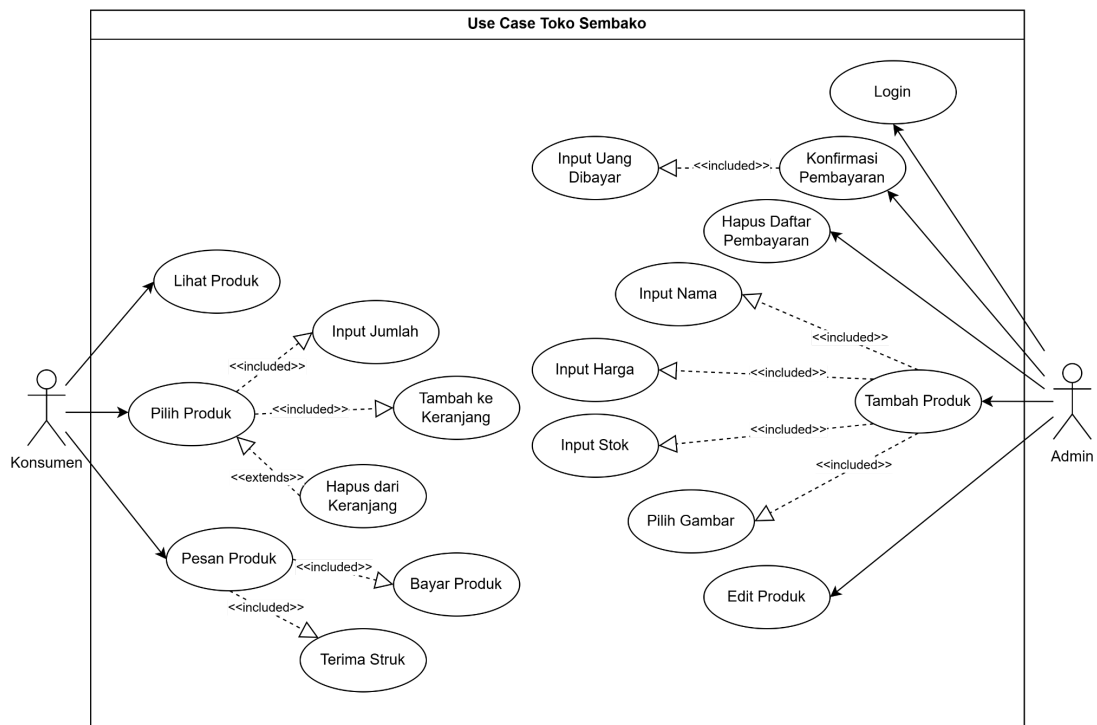
Contoh

```
private void connectDB() {
    try {
        conn = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/toko_sembako", user:"root", password:"");
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Gagal koneksi ke database: " + ex.getMessage());
    }
}
```

Method `connectDB()` menggunakan blok `try-catch` untuk menangani kemungkinan kesalahan saat mencoba koneksi ke database menggunakan `DriverManager.getConnection()`. Jika koneksi gagal, `SQLException` akan ditangkap dan pesan kesalahan ditampilkan menggunakan `JOptionPane.showMessageDialog()`.

D. PERANCANGAN *UNIFIED MODELING LANGUAGE*

Menggunakan use case :



Use case diagram "Toko Sembako" menggambarkan dua aktor utama, yaitu Pelanggan dan Admin, serta interaksi mereka dengan sistem. Pelanggan berperan dalam proses pembelian barang, mulai dari melihat produk, memilih dan memasukkan jumlah produk, hingga menambahkannya ke keranjang. Pelanggan juga dapat menghapus produk dari keranjang sebagai tindakan opsional. Setelah itu, pelanggan bisa memesan produk, melakukan pembayaran dengan menginput jumlah uang yang dibayar, mengkonfirmasi pembayaran, dan menerima struk sebagai bukti transaksi.

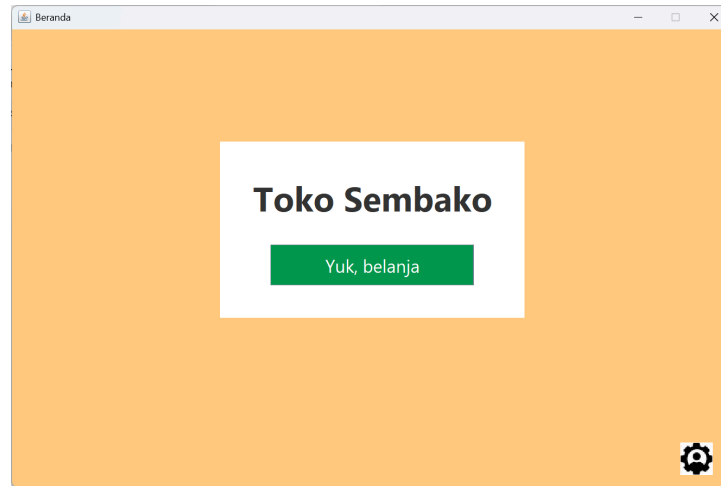
Sementara itu, Admin bertugas mengelola data produk dalam sistem. Admin harus login terlebih dahulu, kemudian dapat menambah produk baru atau mengedit produk yang sudah ada. Proses penambahan produk mencakup input nama, harga, stok, dan gambar produk.

Dalam diagram ini terdapat dua jenis relasi antar use case, yaitu `<<include>>` dan `<<extend>>`. Relasi include berarti use case tersebut selalu disertakan dalam use case utama, contohnya saat pelanggan memilih produk, ia wajib mengisi jumlah. Sedangkan relasi extend menunjukkan bahwa use case hanya terjadi dalam kondisi tertentu, misalnya pelanggan bisa saja menghapus produk dari keranjang jika dibutuhkan.

Secara umum, alur pelanggan mencakup proses melihat produk, memilih dan menambahkan ke keranjang, memesan, membayar, dan menerima struk. Sedangkan alur admin meliputi login, lalu menambah atau mengedit produk. Diagram ini mencerminkan alur sistem penjualan sembako secara lengkap dan terstruktur.

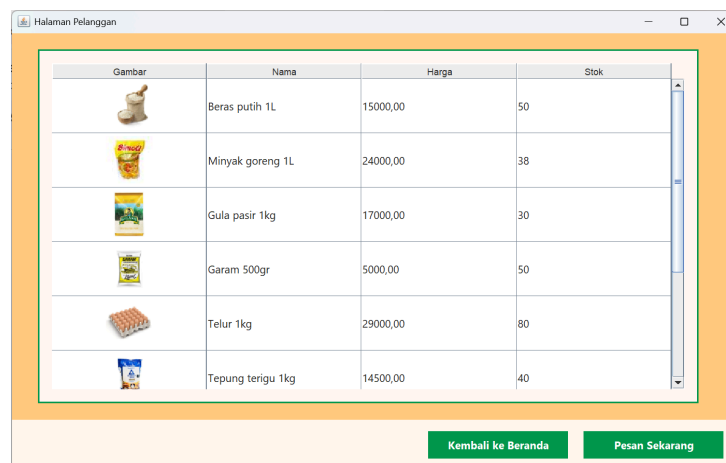
E. IMPLEMENTASI KODE PROGRAM DAN PENGUJIAN

1. Beranda



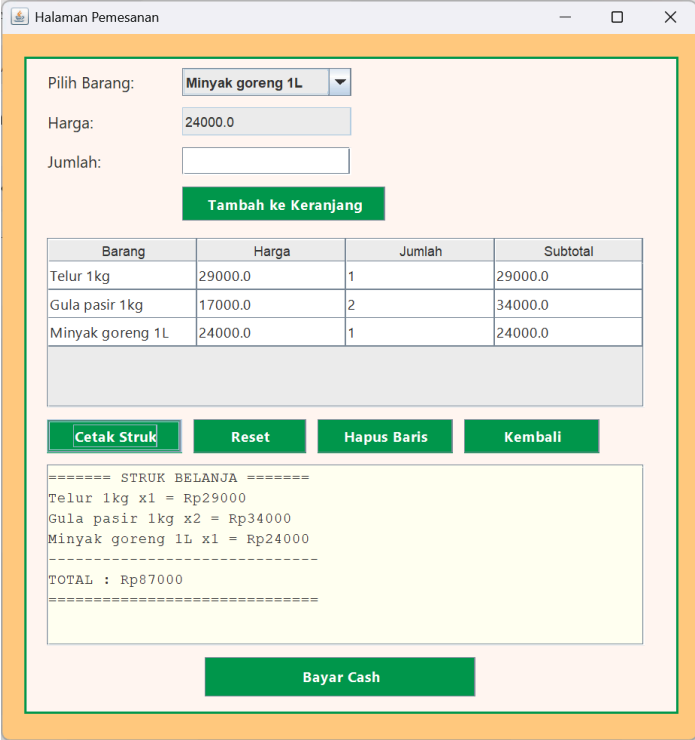
Halaman beranda aplikasi Toko Sembako berfungsi sebagai titik awal interaksi pelanggan, menampilkan nama toko dan tombol utama "Yuk, belanja" yang mengarahkan pengguna ke halaman utama pelanggan. Tampilan ini dirancang sederhana untuk memudahkan pelanggan memulai proses belanja. Selain itu, tersedia ikon pengaturan di pojok kanan bawah yang memungkinkan admin atau kasir mengakses menu pengelolaan sistem secara terpisah dari pelanggan.

2. Halaman utama pelanggan



Halaman ini berfungsi untuk menampilkan produk apa saja yang tersedia di Toko sembako, sehingga pelanggan dapat melihat produk sebelum menentukan akan bertransaksi atau tidak. Terdapat tombol "pesan sekarang" untuk lanjut ke tahap halaman pemesanan.

3. Halaman pemesanan



Pilih Barang: Minyak goreng 1L

Harga: 24000.0

Jumlah:

Tambah ke Keranjang

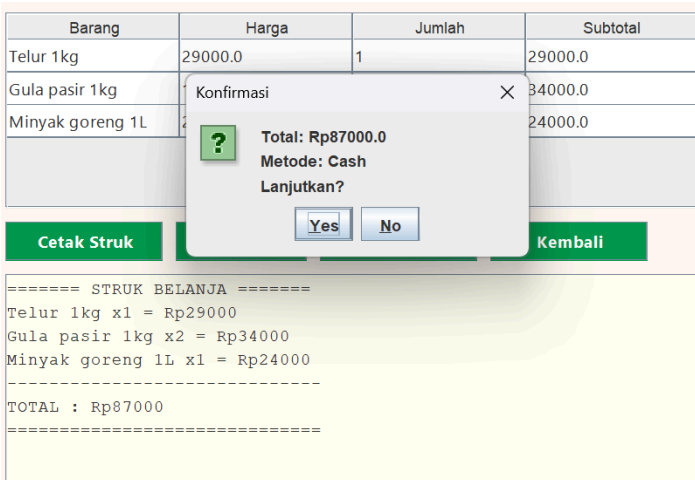
Barang	Harga	Jumlah	Subtotal
Telur 1kg	29000.0	1	29000.0
Gula pasir 1kg	17000.0	2	34000.0
Minyak goreng 1L	24000.0	1	24000.0

Cetak Struk **Reset** **Hapus Baris** **Kembali**

```
===== STRUK BELANJA =====  
Telur 1kg x1 = Rp29000  
Gula pasir 1kg x2 = Rp34000  
Minyak goreng 1L x1 = Rp24000  
-----  
TOTAL : Rp87000  
=====
```

Bayar Cash

Tampilan utama halaman pemesanan, dimana pelanggan dapat memilih barang dari dropdown, melihat harga otomatis, mengisi jumlah, dan menambahkan barang ke keranjang melalui tombol "Tambah ke Keranjang". Di bawahnya terdapat tabel keranjang yang menampilkan daftar belanja (barang, harga, jumlah, subtotal) serta area struk dan tombol aksi seperti Cetak Struk, Reset, Hapus Baris, Kembali, dan Bayar Cash. Tampilan ini memudahkan pelanggan mengelola daftar belanja sebelum melanjutkan pembayaran.



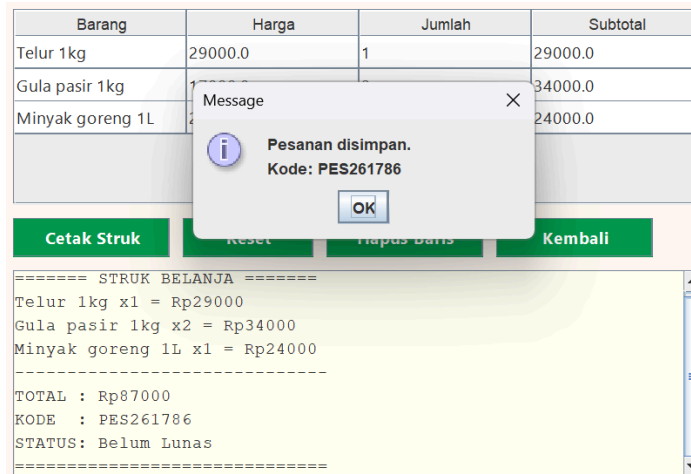
Barang	Harga	Jumlah	Subtotal
Telur 1kg	29000.0	1	29000.0
Gula pasir 1kg	Konfirmasi	2	34000.0
Minyak goreng 1L	24000.0	1	24000.0

Cetak Struk **Kembali**

```
===== STRUK BELANJA =====  
Telur 1kg x1 = Rp29000  
Gula pasir 1kg x2 = Rp34000  
Minyak goreng 1L x1 = Rp24000  
-----  
TOTAL : Rp87000  
=====
```

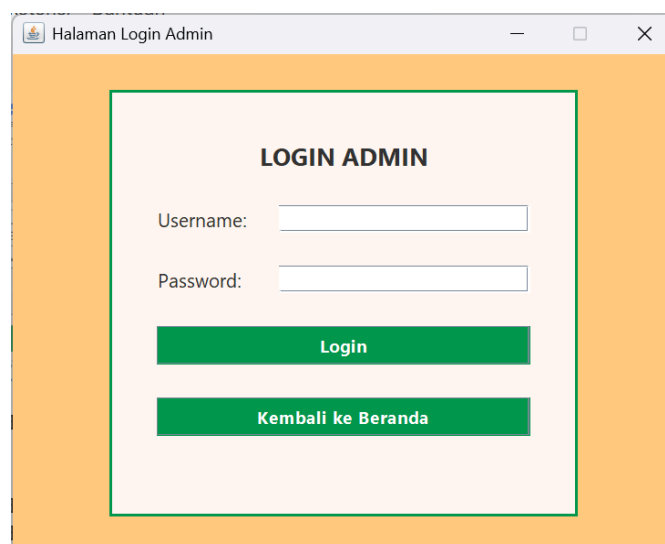
Setelah pelanggan menekan tombol "Bayar Cash", muncul jendela konfirmasi yang menampilkan total belanja, metode pembayaran (Cash), dan pilihan

untuk melanjutkan transaksi atau membatalkannya. Dialog ini berfungsi untuk memastikan pelanggan yakin dengan transaksi yang akan dilakukan sebelum data disimpan ke sistem.



Jika pelanggan memilih "Yes" pada dialog sebelumnya, sistem akan menampilkan pesan bahwa pesanan berhasil disimpan, lengkap dengan **kode pesanan unik** sebagai bukti transaksi. Notifikasi ini menjadi penanda bahwa seluruh proses telah selesai dan data tersimpan di sistem dengan aman. Selanjutnya pelanggan akan ke kasir untuk konfirmasi pembayaran dan menerima produk belanja.

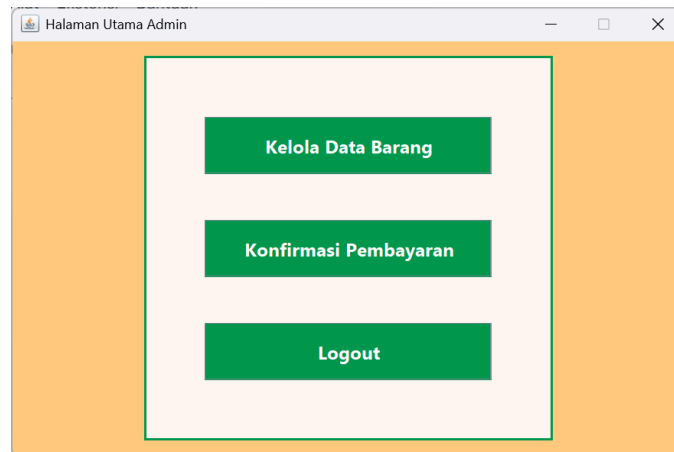
4. Halaman login admin



Halaman login admin berfungsi mengidentifikasi pengguna khusus seperti admin atau kasir. Pengguna harus mengisi kolom username dan password dengan benar untuk dapat mengakses menu administrasi. Tombol "Login" akan memverifikasi identitas pengguna, sementara tombol "Kembali ke

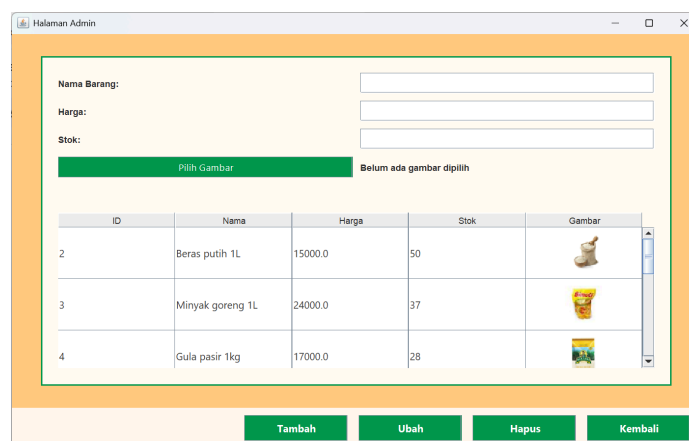
Beranda” memungkinkan pengguna kembali ke halaman utama pelanggan. Halaman ini berfungsi untuk membatasi akses agar fitur manajemen hanya digunakan oleh pihak yang berwenang.

5. Halaman utama admin



Setelah berhasil login, pengguna diarahkan ke halaman utama admin yang berfungsi sebagai pusat navigasi fitur manajemen. Terdapat dua tombol utama, yaitu “Kelola Data Barang” untuk mengatur daftar produk sembako dan “Konfirmasi Pembayaran” untuk memverifikasi transaksi yang belum selesai. Tombol “Logout” digunakan untuk keluar dari mode admin dan kembali ke halaman login. Tampilan ini disusun sederhana agar memudahkan admin mengakses fitur sesuai tugasnya.

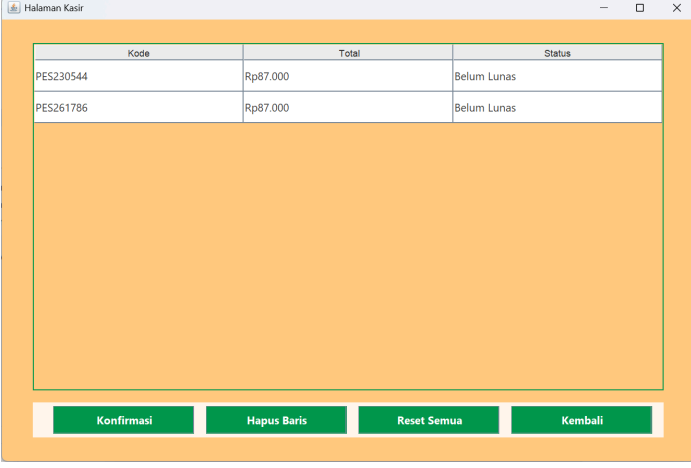
6. Halaman admin (kelola data barang)



Halaman ini memungkinkan admin mengelola seluruh data barang yang tersedia di toko. Admin dapat menambahkan barang baru dengan mengisi nama, harga, dan memilih gambar produk, serta melihat daftar barang dalam bentuk tabel. Tersedia tombol aksi seperti “Tambah”, “Ubah”, “Hapus”, dan “Reset” untuk mengatur data sesuai kebutuhan. Fitur ini dirancang untuk

memastikan informasi produk selalu akurat dan up to date, demi kelancaran proses transaksi di sisi pelanggan.

7. Halaman kasir (konfirmasi pembayaran)



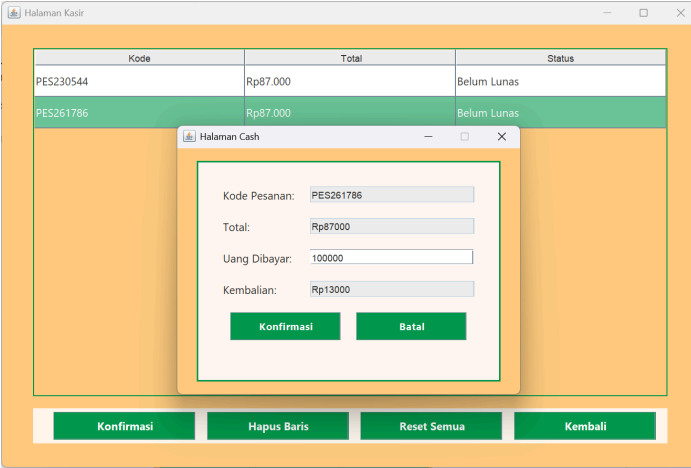
The screenshot shows a web application window titled "Halaman Kasir". It contains a table with three columns: "Kode", "Total", and "Status". There are two rows of data. Below the table, there are four green buttons: "Konfirmasi", "Hapus Baris", "Reset Semua", and "Kembali".

Kode	Total	Status
PES230544	Rp87.000	Belum Lunas
PES261786	Rp87.000	Belum Lunas

Konfirmasi Hapus Baris Reset Semua Kembali

Halaman ini digunakan oleh kasir untuk memantau dan mengelola daftar pesanan yang belum dibayar. Setiap entri menampilkan kode pesanan, total belanja, metode pembayaran, dan status pembayaran. Kasir dapat memilih salah satu transaksi dan menekan tombol "Konfirmasi" untuk memulai proses pembayaran. Tersedia juga tombol "Hapus Baris" untuk menghapus transaksi tertentu, "Reset Semua" untuk mengosongkan daftar, serta "Kembali" untuk kembali ke menu admin. Halaman ini memastikan transaksi yang tertunda dapat diselesaikan secara tertib dan akurat.

8. Halaman cash (struk pembayaran)



The screenshot shows the "Halaman Kasir" window with the "Halaman Cash" modal open. The modal contains a form with the following fields: "Kode Pesanan" (PES261786), "Total" (Rp87000), "Uang Dibayar" (100000), and "Kembalian" (Rp13000). There are two buttons: "Konfirmasi" and "Batal".

Kode Pesanan: PES261786
Total: Rp87000
Uang Dibayar: 100000
Kembalian: Rp13000

Konfirmasi Batal

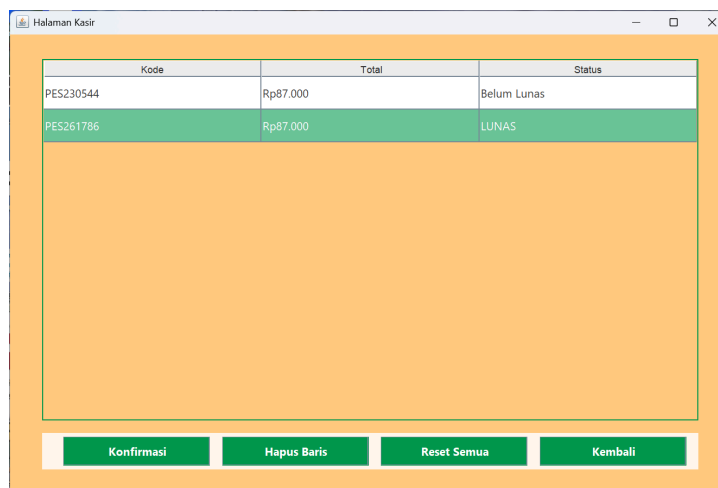
Konfirmasi Hapus Baris Reset Semua Kembali

Halaman ini menunjukkan form pembayaran cash yang muncul setelah kasir memilih transaksi yang belum lunas. Kasir diminta untuk memasukkan jumlah uang yang dibayarkan oleh pelanggan. Sistem akan menghitung kembalian secara otomatis berdasarkan total belanja. Tombol "Konfirmasi" digunakan

untuk menyimpan data pembayaran, sementara tombol “Batal” memungkinkan kasir membatalkan proses jika ada kesalahan input. Tampilan ini memastikan proses pembayaran dilakukan dengan teliti.



Setelah pembayaran dikonfirmasi, sistem akan menampilkan jendela struk pembayaran dalam bentuk digital. Struk ini mencakup informasi lengkap: kode pesanan, total pembayaran, uang yang dibayarkan, jumlah kembalian, dan status pembayaran (LUNAS). Tampilan ini berfungsi sebagai bukti transaksi sah dan memudahkan kasir memberikan informasi yang jelas kepada pelanggan.



Halaman ini menunjukkan daftar transaksi setelah pembayaran dilakukan. Status pada kolom transaksi berubah menjadi “LUNAS”, menandakan bahwa proses pembayaran telah berhasil dan tidak memerlukan tindakan lebih lanjut. Hal ini membantu kasir membedakan transaksi yang sudah selesai dari yang masih tertunda, serta menjaga agar proses pencatatan tetap rapi dan akurat.

F. KESIMPULAN

Proyek Program Penjualan Sembako yang bertujuan untuk menggantikan sistem manual toko sembako dengan aplikasi desktop berbasis Java Swing yang mengadopsi prinsip Object-Oriented Programming. Aplikasi ini mendukung proses pemesanan barang, pengelolaan stok, serta pembayaran dengan metode cash, sehingga meningkatkan efisiensi dan akurasi operasional toko.

Melalui pendekatan OOP seperti enkapsulasi, pewarisan, dan polimorfisme, sistem berhasil dirancang secara modular dan terstruktur. Fitur-fitur utama seperti login admin, CRUD barang, keranjang belanja pelanggan, konfirmasi pembayaran oleh kasir, hingga pencetakan struk, telah diimplementasikan dengan baik. Pengurangan stok otomatis dan antarmuka yang ramah pengguna semakin memperkuat keandalan sistem ini.

Secara keseluruhan, aplikasi ini mampu memenuhi kebutuhan pengguna (admin dan pelanggan) dalam pengelolaan transaksi sembako secara praktis dan efisien, serta menjadi bukti nyata penerapan konsep pemrograman berorientasi objek dalam proyek nyata.