**LAPORAN *PROJECT***

**STUDI KASUS : *“QEEMLA SKIN & BODY CARE”***

**PEMROGRAMAN BERORIENTASI OBJEK F-081**



**Dosen Pengampu :**

**Fawwaz Ali Akbar, S.Kom.,M.Kom**

**Oleh :**

|  |  |
| --- | --- |
| **1. Nabila Farah Hanani** | **(23081010003)** |
| **2. Clarisah Razendri Ingnasia** | **(23081010016)** |
| **3. Lovina Putri Pratama** | **(23081010018)** |
| **4. Salman Alfarizi** | **(23081010020)** |
| **5. Achmad Zaki Al Akbar** | **(23081010022)** |

**PROGRAM STUDI INFORMATIKA FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR 2025**

## DAFTAR ISI

[**DAFTAR ISI i**](#_Toc185236037)

[**DAFTAR GAMBAR iii**](#_Toc185236038)

[**KATA PENGANTAR iv**](#_Toc185236039)

[**BAB I PENDAHULUAN 1**](#_Toc185236040)

[**1.1 Latar Belakang 1**](#_Toc185236042)

[**1.2 Rumusan Masalah 2**](#_Toc185236043)

[**1.3 Tujuan Laporan 2**](#_Toc185236044)

[**BAB II FUNGSI DAN FITUR APLIKASI 3**](#_Toc185236045)

[**BAB III FLOW APLIKASI 6**](#_Toc185236045)

[**3.1 Proses Awal 7**](#_Toc185236042)

[**3.2 Proses Login 7**](#_Toc185236043)

[**3.3 Halaman Utama 7**](#_Toc185236044)

[**3.4 Input Produk 7**](#_Toc185236042)

[**3.5 Daftar Produk 7**](#_Toc185236043)

[**3.6 Cek Expired Produk 8**](#_Toc185236044)

[**3.7 Remind Produk 8**](#_Toc185236042)

[**3.8 Log Aktivitas 8**](#_Toc185236042)

[**3.9 Logout 8**](#_Toc185236042)

[**BAB IV TEKNOLOGI YANG DIGUNAKAN 9**](#_Toc185236045)

[**4.1 Bahasa Pemrograman Java 9**](#_Toc185236042)

[**4.2 Jawa Swing 9**](#_Toc185236043)

[**4.3 Java Database Connectivity (JDBC) 9**](#_Toc185236044)

[**4.4 Database MySQL 1**](#_Toc185236042)**0**

[**4.5 Netbeans 10**](#_Toc185236043)

[**4.6 Java Runtime Environment (JRE) 10**](#_Toc185236043)

[**4.7 Apache Tomcat (Opsional) 11**](#_Toc185236043)

[**BAB V HASIL DAN PEMBAHASAN 12**](#_Toc185236045)

[**5.1 Login Qeemla 12**](#_Toc185236042)

[**5.2 Dashboard Qeemla 17**](#_Toc185236043)

[**5.3 Input Produk 25**](#_Toc185236044)

[**5.4 Daftar Produk Qeemla 48**](#_Toc185236042)

[**5.5 Cek Expired Qeemla 62**](#_Toc185236043)

[**5.6 Remind Produk Qeemla 69**](#_Toc185236042)

[**5.7 Log Aktivitas 73**](#_Toc185236043)

[**BAB VI KESIMPULAN DAN SARAN 77**](#_Toc185236045)

[**6.1 Proses Awal 77**](#_Toc185236042)

[**6.2 Proses Login 77**](#_Toc185236043)

[**BAB VII LINK GITHUB 78**](#_Toc185236045)

## DAFTAR GAMBAR

[**Gambar 1. Flow Aplikasi Qeemla Skin & Body Care 6**](#_Toc185235981)

[**Gambar 2. Tampilan Halaman Login 12**](#_Toc185235981)

[**Gambar 3. Tampilan Halaman Login Berhasil 12**](#_Toc185235981)

[**Gambar 4. Tampilan Halaman Login Gagal 13**](#_Toc185235981)

[**Gambar 5. Tampilan Halaman Dashboard Qeemla 17**](#_Toc185235981)

[**Gambar 6. Tampilan Halaman Input Qeemla 25**](#_Toc185235981)

[**Gambar 7. Tampilan Halaman Input Qeemla Bercode Berhasil Dipindahi 26**](#_Toc185235981)

[**Gambar 8. Tampilan Halaman Input Qeemla Berhasil Disimpan Ke Database 26**](#_Toc185235981)

[**Gambar 9. Tampilan Halaman Daftar Produk Qeemla 48**](#_Toc185235981)

[**Gambar 10. Tampilan Halaman Daftar Produk Qeemla Untuk Edit 49**](#_Toc185235981)

[**Gambar 11. Tampilan Halaman Daftar Produk Qeemla Konfirmasi Hapus 50**](#_Toc185235981)

[**Gambar 13. Tampilan Halaman Cek Expired Produk 62**](#_Toc185235981)

[**Gambar 14. Tampilan Halaman Remind Produk 69**](#_Toc185235981)

[**Gambar 15. Tampilan Halaman Riwayat Log Aktivitas 73**](#_Toc185235981)

[**Gambar 16. Tampilan Repository Github 78**](#_Toc185235981)

## KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan karunia-Nya, sehingga kami dapat menyelesaikan laporan proyek dengan judul "Laporan Project Studi Kasus: Sistem Manajemen Aplikasi Qeemla Skin & Body Care" ini. Laporan ini disusun guna memenuhi tugas UAS (Ujian Akhir Semester) dalam mata kuliah Pemrograman Berorientasi Objek F-081.

Dalam laporan ini, kami membahas implementasi sistem manajemen aplikasi untuk Qeemla Skin & Body Care, sebuah produk skincare dan perawatan tubuh lokal dari Gresik, Jawa Timur. Studi kasus ini mencakup pengembangan fitur-fitur utama seperti login pengguna, input dan daftar produk, pengecekan masa kadaluarsa produk (expired check), pengingat otomatis masa kadaluarsa (remind expired), dan logaudit (riwayat aktivitas). Sistem ini juga dilengkapi dengan fitur logout untuk keamanan dan privasi pengguna.

Proyek ini dikembangkan menggunakan bahasa pemrograman Java dengan antarmuka grafis Java Swing, yang memungkinkan pengelolaan data produk dan kontrol sistem secara terstruktur dan efisien. Dengan adanya fitur-fitur ini, aplikasi diharapkan dapat membantu manajemen produk secara lebih baik, khususnya dalam hal pengawasan masa berlaku produk agar kualitas produk tetap terjaga.

Penulisan laporan ini diharapkan dapat memberikan gambaran yang jelas mengenai proses pengembangan dan penerapan sistem manajemen aplikasi Qeemla Skin & Body Care, sekaligus menambah wawasan mengenai pemanfaatan teknologi Java dalam mendukung usaha lokal. Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kami sangat mengharapkan segala bentuk kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan wawasan baru bagi pembaca, khususnya dalam memahami penerapan sistem digital berbasis Java untuk layanan skin & body care lokal seperti Qeemla Skin.

**BAB I**

**PENDAHULUAN**

## 1.1 Latar Belakang

Di era digital saat ini, banyak pelaku usaha lokal mulai memanfaatkan teknologi informasi untuk meningkatkan efisiensi operasional, memperluas jangkauan pasar, serta memberikan pelayanan yang lebih optimal kepada pelanggan. Perubahan perilaku konsumen yang semakin terbiasa dengan layanan digital dan transaksi daring menuntut pelaku usaha untuk terus beradaptasi dengan perkembangan teknologi, tidak terkecuali dalam industri skincare dan body care. Qeemla Skin & Body Care merupakan salah satu brand lokal yang bergerak di bidang perawatan kulit dan tubuh, berasal dari Gresik, Jawa Timur. Sebagai pelaku usaha yang tengah berkembang, Qeemla Skin & Body Care menghadapi tantangan dalam mengelola aktivitas operasional secara efisien, khususnya dalam hal pencatatan dan pemantauan produk, serta penyediaan layanan yang responsif dan terpersonalisasi kepada pelanggan. Selama ini, beberapa aktivitas bisnis seperti pengelolaan stok produk dan pemantauan masa kedaluwarsa masih dilakukan secara manual, sehingga berpotensi menimbulkan ketidakteraturan dan keterlambatan dalam pelayanan.

Di sisi lain, permintaan konsumen terhadap layanan yang cepat, akurat, dan terintegrasi semakin tinggi. Konsumen menginginkan kemudahan dalam mengakses informasi produk, memastikan kualitas dan validitas produk yang dibeli, serta memperoleh pengalaman yang lebih aman dan nyaman. Untuk menjawab kebutuhan tersebut, diperlukan sebuah sistem manajemen digital yang dapat membantu pelaku usaha dalam melakukan pengelolaan produk secara menyeluruh dan terstruktur. Sebagai solusi, dikembangkanlah sebuah aplikasi manajemen produk menggunakan bahasa pemrograman Java dengan antarmuka grafis Java Swing. Sistem ini mencakup fitur-fitur utama seperti login pengguna, input dan daftar produk, cek masa kadaluarsa (expired check), pengingat otomatis masa kadaluarsa (remind expired), serta logout. Aplikasi ini dirancang untuk mempermudah pengelolaan stok dan waktu edar produk, sekaligus membantu pengguna dalam menjaga kualitas produk yang ditawarkan. Dengan adanya sistem ini, proses manajemen operasional Qeemla Skin & Body Care menjadi lebih terstruktur, efisien, dan ramah pengguna (user-friendly). Penggunaan teknologi pemrograman berorientasi objek (OOP) dalam pengembangan aplikasi ini juga menjadi bukti nyata bahwa solusi digital dapat diimplementasikan secara praktis untuk menjawab.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian atau pengembangan sistem ini adalah sebagai berikut:

1. Bagaimana cara mengimplementasikan sistem manajemen aplikasi Qeemla Skin & Body Care menggunakan bahasa pemrograman Java agar dapat mengelola berbagai fitur layanan seperti login pengguna, input data produk, daftar produk, pengecekan tanggal kadaluarsa (expired), pengingat produk mendekati expired (remind expired), serta fitur logout secara efisien dan terintegrasi?
2. Bagaimana pemilihan dan penerapan struktur data yang tepat dalam Java dapat menjamin akurasi, kecepatan, dan keamanan dalam pengelolaan data pengguna serta manajemen data produk, termasuk status kadaluarsa produk secara otomatis dan real-time?
3. Bagaimana konsep pemrograman berorientasi objek (OOP) dapat digunakan untuk mengembangkan sistem yang mendukung proses login, input, edit, hapus data produk, pengecekan expired, pengingat expired, serta logout pengguna secara modular, efisien, dan mudah dikembangkan.

## 1.3 Tujuan Laporan

Tujuan dari pembuatan laporan ini diantaranya:

1. Mengimplementasikan sistem manajemen aplikasi Qeemla Skin & Body Care yang mampu menangani fitur layanan utama seperti login pengguna, input data produk, daftar katalog produk, pengecekan masa kadaluarsa (expired), pengingat produk mendekati expired (remind expired), serta proses logout.
2. Menerapkan struktur data yang sesuai dan efisien untuk mengelola data pengguna dan data produk, termasuk validasi otomatis terhadap status expired produk secara real-time, guna menjamin akurasi dan kecepatan dalam proses pengolahan data.
3. Mengevaluasi efektivitas dan kinerja sistem dalam mengintegrasikan proses operasional Qeemla Skin & Body Care secara digital, meningkatkan efisiensi kerja, serta mendukung pengambilan keputusan berbasis data melalui antarmuka aplikasi yang terstruktur dan responsif.

# BAB II

# FUNGSI DAN FITUR APLIKASI

Aplikasi manajemen *Qeemla Skin & Body Care* dirancang untuk membantu pemilik usaha dalam mengelola kegiatan operasional harian secara lebih teratur, cepat, dan efisien. Dalam proses bisnis yang sebelumnya masih dilakukan secara manual atau terpisah-pisah, aplikasi ini berperan penting sebagai solusi digital yang mengintegrasikan berbagai kebutuhan utama usaha dalam satu sistem. Fungsi utama aplikasi ini adalah sebagai sarana pengelolaan data pengguna dan produk, pelayanan konsultasi, proses transaksi, serta sistem reward dan review pelanggan. Dengan adanya sistem ini, berbagai proses yang sebelumnya memerlukan pencatatan manual kini bisa dilakukan secara otomatis dan tersistem. Hal ini tidak hanya mempercepat pekerjaan, tapi juga mengurangi risiko kesalahan dalam pencatatan dan duplikasi data.

Fungsi lainnya adalah memudahkan pemilik usaha dalam memantau dan memperbarui katalog produk, termasuk informasi seperti harga, deskripsi, atau stok barang. Selain itu, aplikasi juga memfasilitasi komunikasi antara pelanggan dan pihak *Qeemla Skin & Body Care* melalui fitur konsultasi, yang memungkinkan pengguna bertanya langsung seputar masalah kulit dan produk yang cocok, tanpa perlu datang langsung ke toko.

Dalam hal penjualan, aplikasi ini juga berfungsi sebagai penghubung antara *Qeemla Skin & Body Care* dan marketplace seperti Shopee, sehingga pelanggan bisa melakukan transaksi langsung dari aplikasi atau diarahkan ke toko resmi secara online. Fungsi ini memperluas jangkauan pasar dan memberikan lebih banyak opsi pembelian kepada pelanggan.

Tidak hanya itu, aplikasi juga memiliki sistem poin reward yang mendorong pelanggan untuk lebih aktif, seperti dengan cara mengumpulkan poin dari transaksi atau konsultasi, yang nantinya bisa ditukar dengan layanan gratis atau potongan harga. Ditambah lagi, fitur review memungkinkan pelanggan untuk memberikan penilaian dan tanggapan terhadap produk yang telah mereka gunakan, yang sangat membantu dalam peningkatan kualitas layanan *Qeemla Skin & Body Care*  ke depannya.

Secara keseluruhan, fungsi aplikasi ini tidak hanya mendukung aktivitas manajemen internal usaha, tapi juga meningkatkan kenyamanan dan pengalaman pengguna sebagai pelanggan. Dengan memanfaatkan pendekatan pemrograman berbasis objek (OOP) dan antarmuka Java Swing, aplikasi ini menjadi lebih terstruktur dan mudah digunakan oleh semua pihak yang terlibat.

Pada studi kasus Sistem Manajemen Aplikasi *Qeemla Skin & Body Care,* sistem ini mencakup beberapa fitur utama, di antaranya:

1. Login dan Logout

Sistem menyediakan autentikasi pengguna (login) untuk membatasi akses ke dalam sistem hanya bagi pihak yang berwenang. Fitur logout juga disediakan untuk menjaga keamanan data selama penggunaan aplikasi.

1. Input dan Manajemen Data Produk

Aplikasi memungkinkan pengguna untuk melakukan input produk baru, termasuk informasi seperti nama, harga, stok, dan tanggal kedaluwarsa. Proses ini dibuat mudah melalui form antarmuka grafis Java Swing yang intuitif.

1. Daftar Produk dan Informasi Detail

Pengguna dapat melihat daftar semua produk yang tersimpan di sistem secara real-time. Informasi produk ditampilkan secara lengkap dan sistematis untuk memudahkan pemantauan dan pencarian.

1. Cek Tanggal Kadaluarsa Produk (Expired Check)

Sistem secara otomatis memeriksa tanggal kadaluarsa setiap produk. Produk yang telah melewati masa berlaku akan ditandai sebagai expired, sehingga pengguna dapat mengambil tindakan lanjutan.

5. Pengingat Untuk Masa Kadaluarsa Produk (Remind Product)

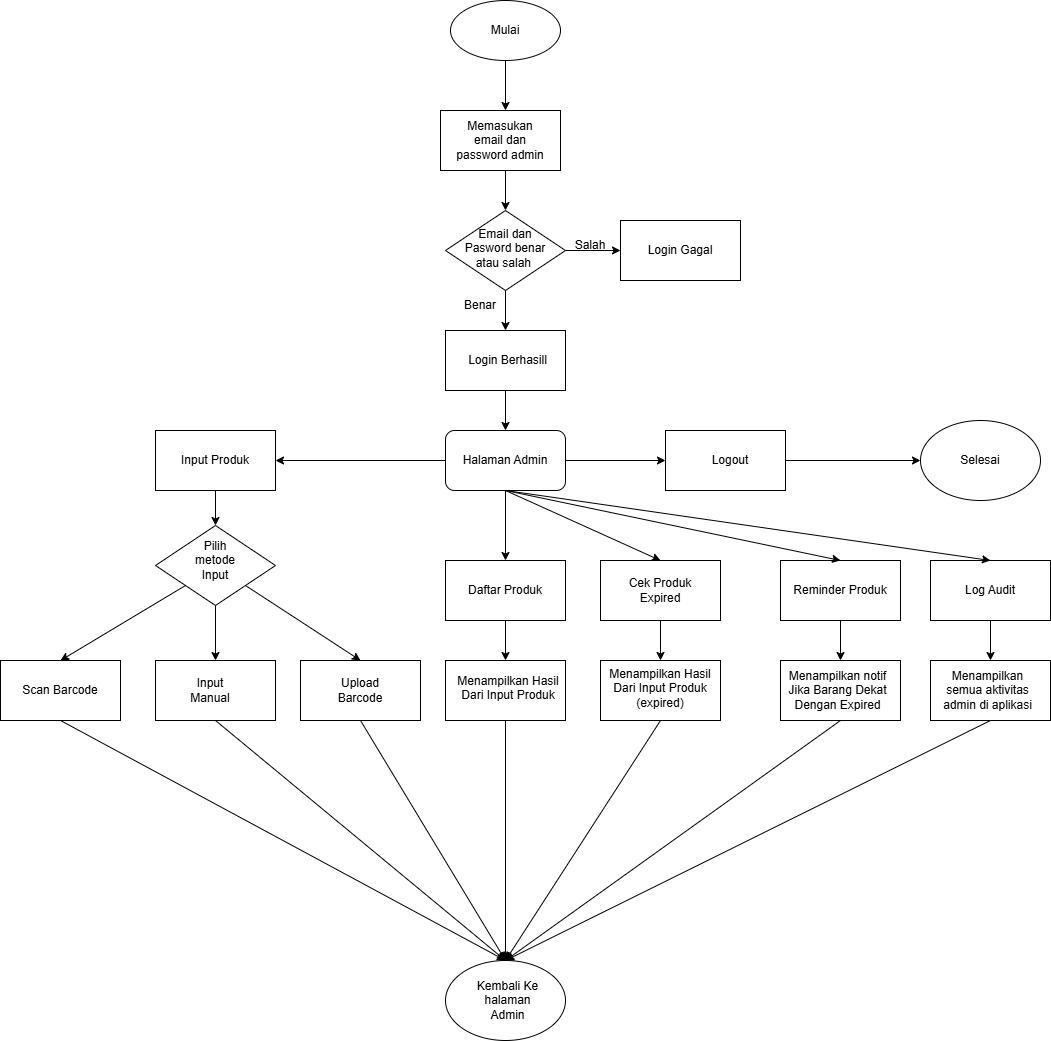
Aplikasi memberikan notifikasi/pengingat untuk produk yang akan kadaluarsa dalam waktu dekat. Fungsi ini sangat penting untuk menjaga kualitas produk yang dijual dan menghindari kerugian akibat produk kadaluarsa.

Untuk mengimplementasikan berbagai fitur dalam sistem manajemen aplikasi Qeemla Skin & Body Care, digunakan pendekatan Pemrograman Berorientasi Objek (OOP) dengan bahasa Java, serta antarmuka grafis berbasis Java Swing. Pendekatan ini memungkinkan pengembangan sistem yang modular, terstruktur, dan mudah dikembangkan sesuai kebutuhan usaha. Fitur utama seperti login, input produk, daftar produk, cek expired, remind expired, dan logout diimplementasikan melalui pemisahan kelas-kelas fungsional. Komponen antarmuka seperti JFrame, JPanel, JButton, JTextField, JLabel, JTable, dan JOptionPane digunakan untuk membangun UI yang interaktif dan ramah pengguna. Sementara itu, logika bisnis seperti validasi input, pengecekan tanggal kadaluarsa, serta penjadwalan pengingat dilakukan melalui event handling dan mekanisme listener yang mengatur respon aplikasi terhadap aksi pengguna.

Fitur cek expired secara otomatis menandai produk yang sudah melewati tanggal kadaluarsa, sedangkan remind expired memberikan notifikasi untuk produk yang mendekati masa kadaluarsa , berdasarkan logika perbandingan tanggal sistem. Sistem logout berfungsi sebagai pemisah sesi untuk menjaga keamanan data pengguna. Dengan pendekatan ini, aplikasi tidak hanya bekerja secara terstruktur dan responsif, tetapi juga mudah dikembangkan tanpa memerlukan manajemen data kompleks. Semua data disimpan dan diproses secara efisien menggunakan struktur data sederhana yang sesuai dengan skala dan kebutuhan sistem.

## BAB III

## FLOW APLIKASI



*Gambar 1. Flow Aplikasi Qeemla Skin & Body Care*

## 3.1 Proses Awal dimulai administrator untuk masuk ke sistem aplikasi Qeemla dan harus login menggunakan email dan kata sandi yang telah dibuat.

## 3.2 Proses login sistem memverifikasi data admin, termasuk email dan kata sandi untuk memastikan bahwa:

## Login Gagal yakni sistem akan memberi tahu admin jika email atau kata sandi salah.

## Login Berhasil akan menampilkan pesan login berhasil akan ditampilkan oleh sistem jika data yang dimasukkan benar, dan admin akan diarahkan ke halaman utama aplikasi.

## 3.3 Halaman utama admin pada langkah selanjutnya, setelah berhasil untuk login, admin akan masuk ke halaman utama, yang berfungsi sebagai pusat kontrol aplikasi. Di halaman ini, admin dapat mengakses beberapa fitur utama, seperti:

## Input Produk

## Daftar Produk

## Pengecekan Produk

## Pengingat untuk tanggal kadaluarsa

## Log Aktivitas

## 3.4 Input Produk pada halaman ini admin memiliki beberapa opsi untuk memasukkan data produk ke dalam sistem pada fitur ini. Opsi-opsi ini termasuk:

## Scan barcode digunakan untuk admin mengscan barcode produk dengan menggunakan webcam yang ada pada aplikasi Qeemla.

## Input manual digunakan untuk mengisi data produk secara manual tanpa barcode.

## Upload barcode memiliki kemampuan untuk mengunggah file gambar barcode produk. Sistem akan menampilkan informasi produk yang baru dimasukkan setelah produk dimasukkan.

## 3.5 Daftar Produk pada fitur ini memungkinkan melihat manajer semua produk yang terdaftar dalam sistem, baik yang telah dimasukkan sebelumnya maupun yang baru saja dimasukkan. Admin dapat memeriksa daftar produk dan memilih untuk kembali ke halaman utama.

## 3.6 Pengecekan Produk Kadaluarsa pada fitur ini memungkinkan sistem menampilkan produk yang telah mencapai atau mendekati tanggal kadaluarsa. Produk yang akhirnya dapat diperiksa oleh admin dan dikembalikan ke halaman utama setelah selesai.

## 3.7 Pengingat Produk pada sistem ini memberikan pemberitahuan kepada admin tentang produk yang mendekati tanggal kadaluarsa. Pengingat ini membantu admin untuk lebih cepat menangani produk yang perlu diperhatikan lebih lanjut. Setelah admin membaca notifikasi, mereka dapat kembali ke halaman utama.

## 3.8 Log Aktivitas pada fitur log aktivitas menampilkan semua kegiatan yang dilakukan oleh admin melalui aplikasi, yang dapat digunakan untuk pelacakan atau audit. Admin dapat kembali ke halaman utama setelah meninjau log aktivitas.

## 3.9 Logout dan Akhiri Proses digunakan setelah semua tugas selesai, admindapat keluar dari aplikasi. Ini mengakhiri sesi kerja admin, dan aplikasi akan menutup otomatis untuk memulai proses akhir.

# BAB IV

# TEKNOLOGI YANG DIGUNAKAN

Teknologi yang digunakan yakni berbasis Java yang digunakan dalam sistem pengembangan , dengan penekanan khusus pada Java Swing sebagai kerangka pengembangan antarmuka pengguna (GUI). Selain itu, akan dibahas teknologi lain yang terkait dengan pengembangan aplikasi.

**4.1 Bahasa pemrograman Java**

Bahasa Java adalah bahasa pemrograman yang berorientasi pada objek yang digunakan untuk mengembangkan aplikasi ini. Keunggulan utama Jawa adalah kemampuan untuk menyediakan platform independen yang disebut Write Once, Run Anywhere (WORA), yang memungkinkan pengguna menjalankan aplikasi di berbagai platform tanpa mengubah kode sumbernya. Jawa sangat disukai karena akustiknya, keamanannya, dan komunitas penggunanya yang besar. Selain itu, ekosistemnya yang matang dan kaya dengan berbagai pustaka dan kerangka membantu pengembangan aplikasi dengan cepat.

**4.2 Java Swing**

Java Swing adalah bagian dari Java Foundation Classes (JFC), yang menyediakan serangkaian komponen untuk membangun antarmuka pengguna grafis (GUI) untuk aplikasi desktop berbasis Java. Kemampuannya untuk membuat antarmuka pengguna yang interaktif dan responsif merupakan alasan utama penggunaan Java Swing dalam pengembangan aplikasi ini. Java Swing memiliki komponen seperti tombol, label, area teks , dan panel yang digunakan untuk membangun layout aplikasi. Selain itu, Java Swing mendukung berbagai fitur, seperti manajemen tata letak, pengendalian peristiwa, dan komponen rendering grafis yang dapat disesuaikan, yang memungkinkan pengembang membuat antarmuka aplikasi yang tampak profesional.

**4.3 Java Database Connectivity (JDBC)**

JDBC adalah API yang memungkinkan aplikasi Java berinteraksi dengan berbagai DBMS, seperti MySQL dan PostgreSQL, dan memungkinkan aplikasi Java untuk menyimpan dan mengelola data dalam aplikasi ini. Aplikasi dapat melakukan tugas seperti menyimpan, mengambil, dan memperbarui data dengan cepat berkat teknologi ini. Untuk memastikan proses pengelolaan data berjalan lancar, proses koneksi antara aplikasi dan database menggunakan JDBC sangat penting. Aplikasi dapat melakukan query SQL untuk mendapatkan data dari database dan menampilkan hasilnya melalui antarmuka pengguna yang dibangun dengan Java Swing.

**4.4 Database MySQL**

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang digunakan untuk menyimpan data aplikasi. Dalam sistem ini, MySQL bertindak sebagai tempat penyimpanan data produk, pengguna, transaksi, dan elemen lainnya. MySQL dipilih karena stabil, mudah digunakan untuk mengelola data, dan kompatibel dengan teknologi Java melalui JDBC. MySQL mendukung fitur seperti query SQL, yang memungkinkan aplikasi mengakses dan memanipulasi data dengan cepat. Selain itu, MySQL sangat cocok untuk menangani data yang besar.

**4.5 NetBeans**

NetBeans adalah integrasi IDE yang digunakan pengembang untuk mengembangkan aplikasi Java ini. IDE ini menawarkan lingkungan pengembangan yang lengkap untuk pengembang untuk menulis, menguji, dan mengdebug kode Java. IDE ini juga memiliki banyak fitur, seperti manajemen proyek, integrasi dengan database, editor kode, dan fitur debugging, yang membantu pengembang mengembangkan aplikasi dengan lebih mudah. Selain itu, NetBeans mendukung Java Swing secara langsung, yang memungkinkan pengembang membuat antarmuka pengguna visual dan menghubungkannya ke kode logika aplikasi. Dengan menggunakan NetBeans, pengembang dapat mempercepat proses pengembangan aplikasi serta mengurangi jumlah kesalahan kode.

**4.6 Java Runtime Environment (JRE)**

JRE adalah perangkat lunak yang diperlukan untuk menjalankan aplikasi Java. JRE menyediakan semua pustaka dan dependensi yang diperlukan agar aplikasi Java dapat berjalan pada mesin pengguna. Selain itu, JRE memastikan bahwa aplikasi yang dibangun menggunakan Java dapat berjalan dengan baik di berbagai sistem operasi tanpa masalah kompatibilitas.JRE mengeksekusi kode bytecode yang dibuat oleh proses kompilasi aplikasi Java. Selain itu, JRE mendukung pengelolaan memori dan thread untuk membuat aplikasi yang cepat dan responsif.

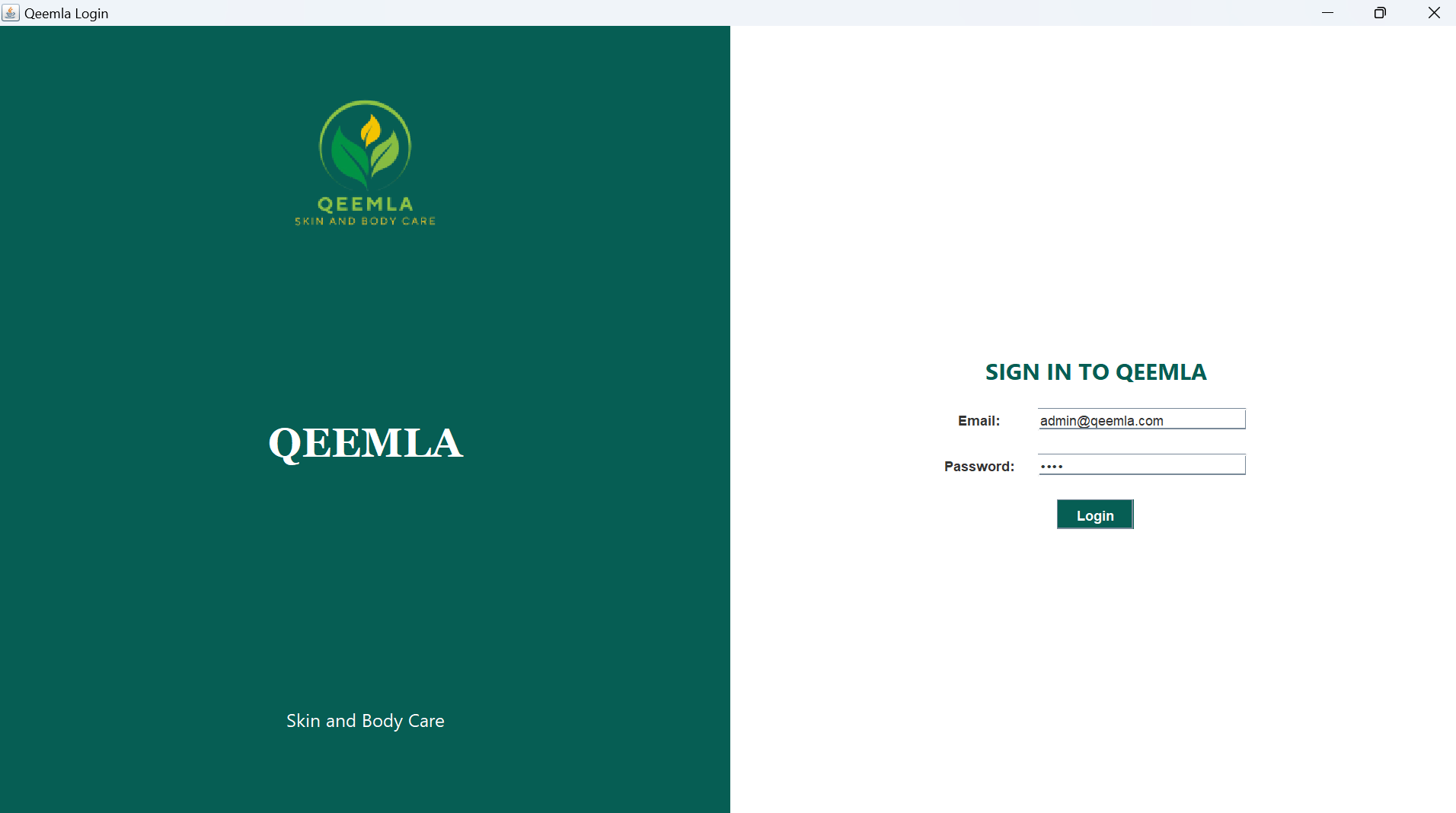
**4.7 Apache Tomcat (Opsional)**

Apache Tomcat adalah server web open-source yang digunakan untuk menjalankan aplikasi web berbasis Java. Jika aplikasi ini memerlukan fitur berbasis web atau aplikasi client-server, Apache Tomcat dapat digunakan sebagai server aplikasi untuk men-deploy aplikasi berbasis Java Servlet dan JSP (JavaServer Pages). Meskipun tidak digunakan di semua aplikasi Java desktop, jika aplikasi ini dikembangkan menjadi aplikasi berbasis web, Tomcat akan menjadi pilihan yang tepat untuk menghosting aplikasi tersebut.

# BAB V

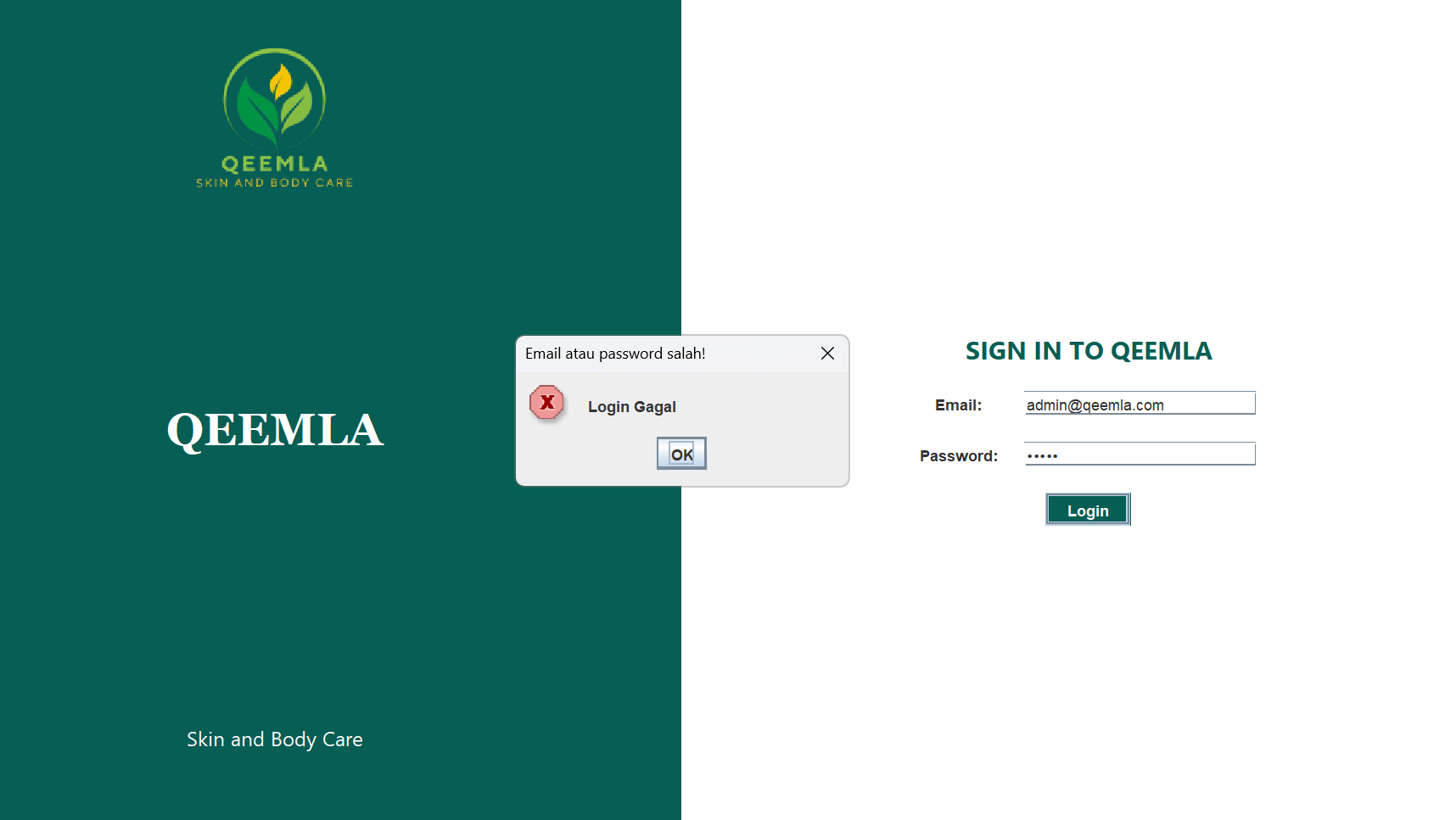
# HASIL DAN PEMBAHASAN

## 5.1 Login Qeemla



*Gambar 2. Tampilan Halaman Login*

*Gambar 3. Tampilan Halaman Login Berhasil*



*Gambar 4. Tampilan Halaman Login Gagal*

import javax.swing.\*;

import java.awt.\*;

public class QeemlaLoginForm extends JFrame {

private JTextField emailField;

private JPasswordField passwordField;

public QeemlaLoginForm() {

setTitle("Qeemla Login");

setSize(700, 400);

setLocationRelativeTo(null);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLayout(new GridLayout(1, 2));

// Panel kiri: branding Qeemla

JPanel leftPanel = new JPanel();

leftPanel.setBackground(new Color(6, 94, 84));

leftPanel.setLayout(new BorderLayout());

ImageIcon logoIcon = new ImageIcon("assets/logo\_qeemla.png");

Image logo = logoIcon.getImage().getScaledInstance(200, 200, Image.SCALE\_SMOOTH);

JLabel logoLabel = new JLabel(new ImageIcon(logo));

logoLabel.setHorizontalAlignment(JLabel.CENTER);

JLabel brandLabel = new JLabel("QEEMLA", JLabel.CENTER);

brandLabel.setFont(new Font("Georgia", Font.BOLD, 36));

brandLabel.setForeground(Color.WHITE);

JLabel tagline = new JLabel("Skin and Body Care", JLabel.CENTER);

tagline.setFont(new Font("Segoe UI", Font.PLAIN, 16));

tagline.setForeground(Color.WHITE);

JPanel brandingPanel = new JPanel(new GridLayout(3, 1));

brandingPanel.setOpaque(false);

brandingPanel.add(logoLabel);

brandingPanel.add(brandLabel);

brandingPanel.add(tagline);

leftPanel.add(brandingPanel, BorderLayout.CENTER);

// Panel kanan: form login

JPanel rightPanel = new JPanel();

rightPanel.setBackground(Color.WHITE);

rightPanel.setLayout(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

JLabel titleLabel = new JLabel("SIGN IN TO QEEMLA");

titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 20));

titleLabel.setForeground(new Color(6, 94, 84));

JLabel emailLabel = new JLabel("Email:");

emailField = new JTextField(20);

JLabel passwordLabel = new JLabel("Password:");

passwordField = new JPasswordField(20);

JButton loginButton = new JButton("Login");

loginButton.setBackground(new Color(6, 94, 84));

loginButton.setForeground(Color.WHITE);

loginButton.setFocusPainted(false);

// Aksi tombol login

loginButton.addActionListener(e -> {

String email = emailField.getText();

String password = new String(passwordField.getPassword());

if (email.equalsIgnoreCase("admin@qeemla.com") && password.equals("1234")) {

JOptionPane.showMessageDialog(this, "Login Berhasil", "Welcome", JOptionPane.INFORMATION\_MESSAGE);

// Tutup form login

dispose();

// Buka dashboard

String username = email.split("@")[0];

DashboardMenu dashboard = new DashboardMenu(username);

dashboard.setVisible(true);

} else {

JOptionPane.showMessageDialog(this, "Login Gagal", "Email atau password salah!", JOptionPane.ERROR\_MESSAGE);

}

});

// Tambah komponen ke form login

gbc.insets = new Insets(10, 10, 10, 10);

gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 2;

rightPanel.add(titleLabel, gbc);

gbc.gridwidth = 1;

gbc.gridy++;

rightPanel.add(emailLabel, gbc);

gbc.gridx = 1;

rightPanel.add(emailField, gbc);

gbc.gridx = 0; gbc.gridy++;

rightPanel.add(passwordLabel, gbc);

gbc.gridx = 1;

rightPanel.add(passwordField, gbc);

gbc.gridx = 0; gbc.gridy++; gbc.gridwidth = 2;

rightPanel.add(loginButton, gbc);

// Gabungkan panel kiri dan kanan

add(leftPanel);

add(rightPanel);

setVisible(true);

}

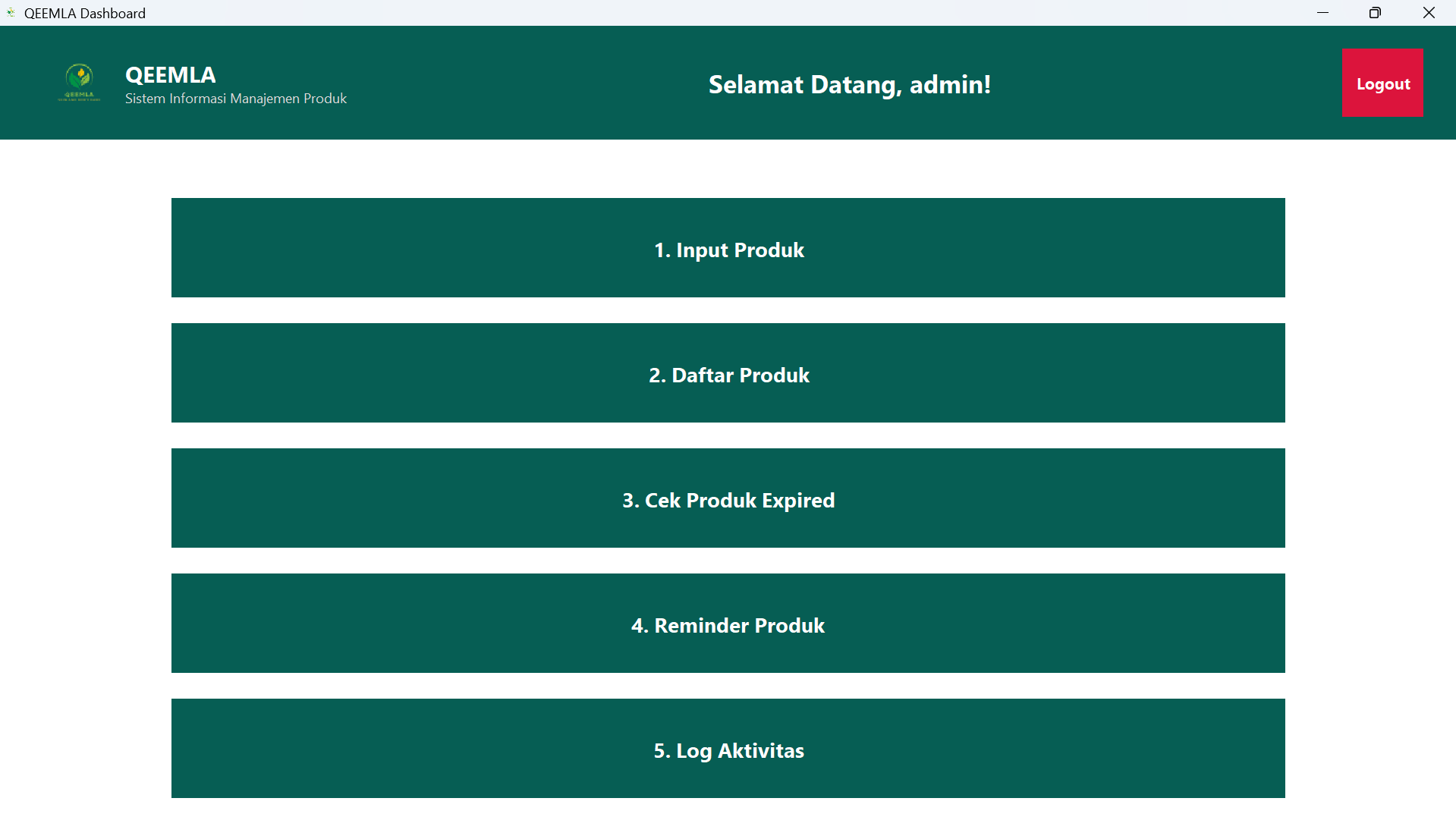
public static void main(String[] args) {

SwingUtilities.invokeLater(QeemlaLoginForm::new);

}

}

## 5.2 Dashboard Qeemla



*Gambar 5. Tampilan Halaman Dashboard Qeemla*

// File: DashboardMenu.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.net.URL;

public class DashboardMenu extends JFrame {

private final String loggedInUsername;

private final ProductDAO productDAO;

private final ExpiryChecker expiryChecker;

private DaftarProductForm activeDaftarProductForm = null;

private CekExpiredForm activeCekExpiredForm = null;

public DashboardMenu(String username) {

this.loggedInUsername = username;

this.productDAO = new ProductDAO();

this.expiryChecker = new ExpiryChecker(productDAO);

setTitle("QEEMLA Dashboard");

setSize(800, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

try {

URL iconUrl = getClass().getClassLoader().getResource("assets/logo\_qeemla.png");

if (iconUrl != null) {

ImageIcon frameIcon = new ImageIcon(iconUrl);

setIconImage(frameIcon.getImage());

}

} catch (Exception e) {

e.printStackTrace();

}

JPanel mainPanel = new JPanel(new BorderLayout());

mainPanel.setBackground(Color.WHITE);

// ========== HEADER ==========

JPanel topBarPanel = new JPanel(new BorderLayout());

topBarPanel.setBackground(new Color(6, 94, 84));

topBarPanel.setBorder(BorderFactory.createEmptyBorder(20, 30, 20, 30));

JPanel leftHeaderPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 0));

leftHeaderPanel.setOpaque(false);

JLabel logoImageLabel = new JLabel();

try {

URL logoUrl = getClass().getClassLoader().getResource("assets/logo\_qeemla.png");

if (logoUrl != null) {

Image originalImage = new ImageIcon(logoUrl).getImage();

Image scaledImage = originalImage.getScaledInstance(60, 60, Image.SCALE\_SMOOTH);

logoImageLabel.setIcon(new ImageIcon(scaledImage));

}

} catch (Exception e) {

e.printStackTrace();

}

JPanel textPanel = new JPanel();

textPanel.setOpaque(false);

textPanel.setLayout(new BoxLayout(textPanel, BoxLayout.Y\_AXIS));

JLabel qeemlaTextLabel = new JLabel("QEEMLA");

qeemlaTextLabel.setFont(new Font("Segoe UI", Font.BOLD, 20));

qeemlaTextLabel.setForeground(Color.WHITE);

JLabel simproTextLabel = new JLabel("Sistem Informasi Manajemen Produk");

simproTextLabel.setFont(new Font("Segoe UI", Font.PLAIN, 12));

simproTextLabel.setForeground(new Color(220, 220, 220));

textPanel.add(qeemlaTextLabel);

textPanel.add(simproTextLabel);

leftHeaderPanel.add(logoImageLabel);

leftHeaderPanel.add(textPanel);

JLabel welcomeLabel = new JLabel("Selamat Datang, " + loggedInUsername + "!");

welcomeLabel.setFont(new Font("Segoe UI", Font.BOLD, 22));

welcomeLabel.setHorizontalAlignment(SwingConstants.CENTER);

welcomeLabel.setForeground(Color.WHITE);

JButton logoutButton = new JButton("Logout");

styleLogoutButton(logoutButton);

topBarPanel.add(leftHeaderPanel, BorderLayout.WEST);

topBarPanel.add(welcomeLabel, BorderLayout.CENTER);

topBarPanel.add(logoutButton, BorderLayout.EAST);

mainPanel.add(topBarPanel, BorderLayout.NORTH);

// ========== TOMBOL ==========

JPanel centerButtonsPanel = new JPanel(new GridLayout(5, 1, 0, 20));

centerButtonsPanel.setBorder(BorderFactory.createEmptyBorder(50, 150, 50, 150));

centerButtonsPanel.setOpaque(false);

JButton inputProductBtn = createDashboardButton("1. Input Produk");

JButton listProductBtn = createDashboardButton("2. Daftar Produk");

JButton checkExpiredBtn = createDashboardButton("3. Cek Produk Expired");

JButton reminderBtn = createDashboardButton("4. Reminder Produk");

JButton logAuditBtn = createDashboardButton("5. Log Aktivitas");

centerButtonsPanel.add(inputProductBtn);

centerButtonsPanel.add(listProductBtn);

centerButtonsPanel.add(checkExpiredBtn);

centerButtonsPanel.add(reminderBtn);

centerButtonsPanel.add(logAuditBtn);

mainPanel.add(centerButtonsPanel, BorderLayout.CENTER);

setContentPane(mainPanel);

// ========== EVENT HANDLER ==========

inputProductBtn.addActionListener(e -> {

InputProductForm inputForm = new InputProductForm(productDAO, loggedInUsername);

inputForm.addWindowListener(new WindowAdapter() {

public void windowClosed(WindowEvent e) {

if (activeDaftarProductForm != null && activeDaftarProductForm.isVisible())

activeDaftarProductForm.refreshTable();

if (activeCekExpiredForm != null && activeCekExpiredForm.isVisible())

activeCekExpiredForm.populateExpiredProducts();

}

});

inputForm.setVisible(true);

});

listProductBtn.addActionListener(e -> {

if (activeDaftarProductForm == null || !activeDaftarProductForm.isVisible()) {

activeDaftarProductForm = new DaftarProductForm(productDAO);

activeDaftarProductForm.addWindowListener(new WindowAdapter() {

public void windowClosed(WindowEvent e) {

activeDaftarProductForm = null;

}

});

activeDaftarProductForm.setVisible(true);

} else {

activeDaftarProductForm.toFront();

activeDaftarProductForm.refreshTable();

}

});

checkExpiredBtn.addActionListener(e -> {

if (activeCekExpiredForm == null || !activeCekExpiredForm.isVisible()) {

activeCekExpiredForm = new CekExpiredForm(productDAO);

activeCekExpiredForm.addWindowListener(new WindowAdapter() {

public void windowClosed(WindowEvent e) {

activeCekExpiredForm = null;

}

});

activeCekExpiredForm.setVisible(true);

} else {

activeCekExpiredForm.toFront();

activeCekExpiredForm.populateExpiredProducts();

}

});

reminderBtn.addActionListener(e -> {

expiryChecker.showReminder(); // ✅ Pakai method showReminder() yang langsung tampilkan JOptionPane

});

logAuditBtn.addActionListener(e -> {

LogAuditForm logAuditForm = new LogAuditForm();

logAuditForm.setVisible(true);

});

logoutButton.addActionListener(e -> {

int confirm = JOptionPane.showConfirmDialog(DashboardMenu.this, "Anda yakin ingin logout?", "Konfirmasi Logout", JOptionPane.YES\_NO\_OPTION);

if (confirm == JOptionPane.YES\_OPTION) {

dispose();

new QeemlaLoginForm().setVisible(true);

}

});

}

private JButton createDashboardButton(String text) {

JButton button = new JButton(text);

button.setFont(new Font("Segoe UI", Font.BOLD, 18));

button.setBackground(new Color(6, 94, 84));

button.setForeground(Color.WHITE);

button.setFocusPainted(false);

button.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));

return button;

}

private void styleLogoutButton(JButton button) {

button.setFont(new Font("Segoe UI", Font.BOLD, 14));

button.setBackground(new Color(220, 20, 60));

button.setForeground(Color.WHITE);

button.setFocusPainted(false);

button.setBorder(BorderFactory.createEmptyBorder(6, 12, 6, 12));

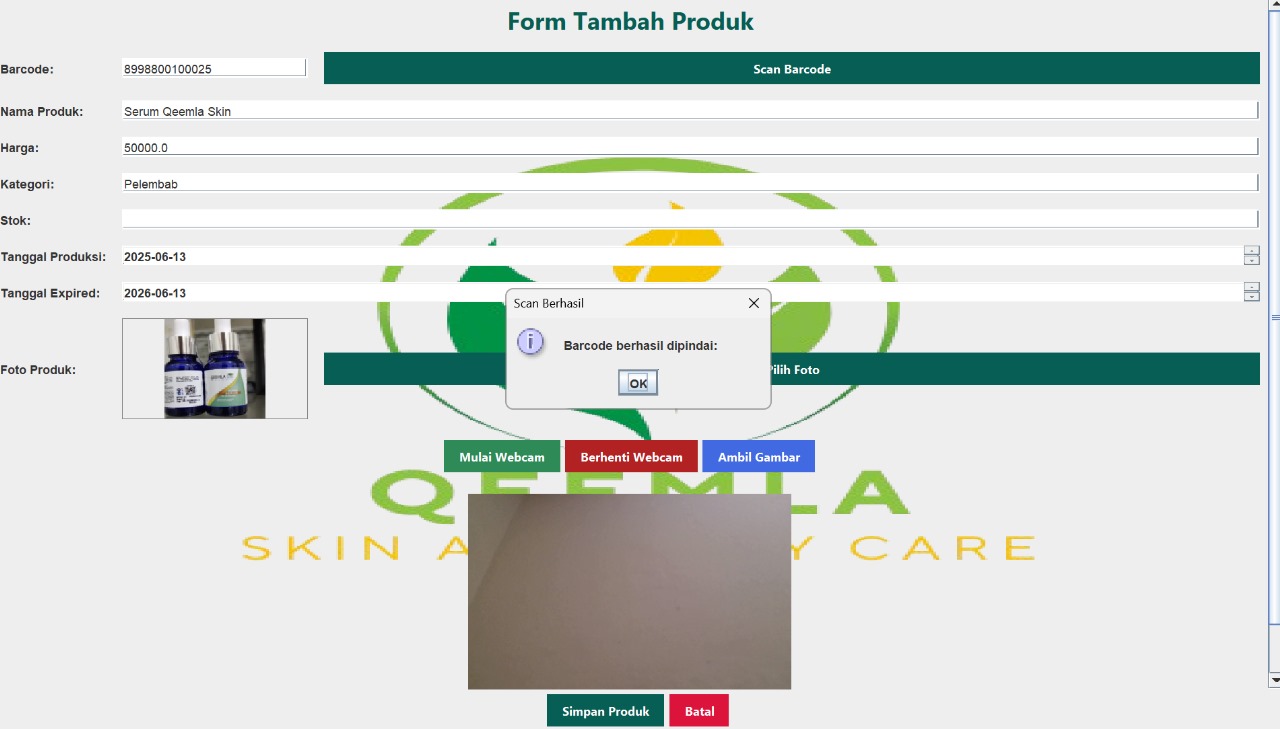
}}

## 

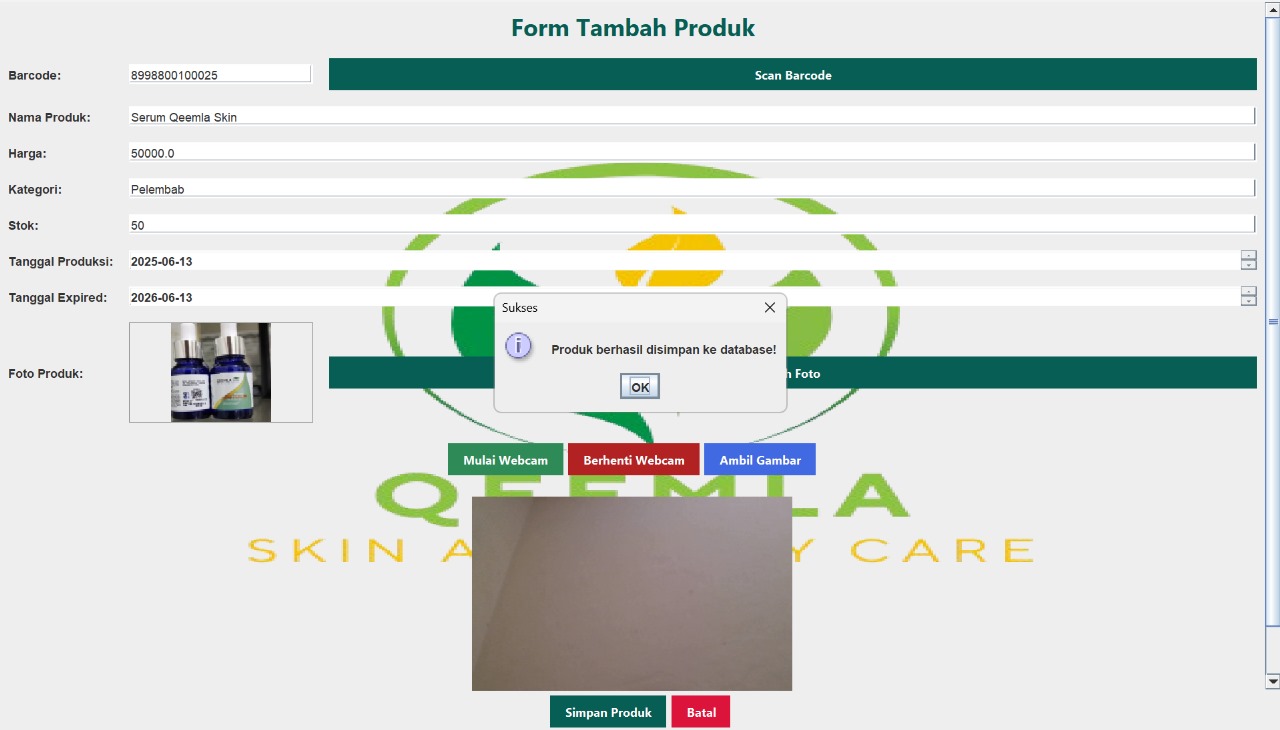
## 5.3 Input Qeemla



*Gambar 6. Tampilan Halaman Input Qeemla*



*Gambar 7. Tampilan Halaman Input Qeemla Barcode Berhasil Dipindahi*



*Gambar 8. Tampilan Halaman Input Qeemla Produk Berhasil Disimpan Ke Database*

**// File: InputProductForm.java**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.io.File;**

**import java.util.Date;**

**import java.util.List;**

**import java.awt.image.BufferedImage;**

**import java.net.URL;**

**import java.time.LocalDate;**

**import java.time.ZoneId;**

**import java.time.format.DateTimeFormatter;**

**import java.time.temporal.ChronoUnit;**

**import java.util.HashMap;**

**import java.util.Map;**

**import java.sql.SQLException;**

**import javax.imageio.ImageIO; // import ini untuk ImageIO**

**import java.text.SimpleDateFormat;**

**import java.util.Calendar;**

**// import github**

**import com.github.sarxos.webcam.Webcam;**

**import com.github.sarxos.webcam.WebcamPanel;**

**import com.github.sarxos.webcam.WebcamResolution;**

**import com.google.zxing.BinaryBitmap;**

**import com.google.zxing.LuminanceSource;**

**import com.google.zxing.MultiFormatReader;**

**import com.google.zxing.NotFoundException;**

**import com.google.zxing.Result;**

**import com.google.zxing.client.j2se.BufferedImageLuminanceSource;**

**import com.google.zxing.common.HybridBinarizer;**

**public class InputProductForm extends JFrame {**

**// --- PENAMBAHAN DEKLARASI tfStok ---**

**private JTextField tfBarcode, tfNama, tfHarga, tfKategori, tfStok;**

**private JSpinner spinnerProduksi, spinnerExpired;**

**private ProductDAO productDAO;**

**private JLabel photoLabel;**

**private File selectedPhoto;**

**private Webcam webcam = null;**

**private WebcamPanel webcamPanel = null;**

**private Thread barcodeScannerThread = null;**

**private JButton startWebcamBtn, stopWebcamBtn, captureBtn;**

**private JPanel webcamControlPanel;**

**private JPanel contentPanel;**

**private boolean isEditMode = false;**

**private String originalCode = null;**

**private String emailLogin;**

**private static class PredefinedProductInfo {**

**String name;**

**double price;**

**String category;**

**String imagePath;**

**public PredefinedProductInfo(String name, double price, String category, String imagePath) {**

**this.name = name;**

**this.price = price;**

**this.category = category;**

**this.imagePath = imagePath;**

**}**

**}**

**private static final Map<String, PredefinedProductInfo> predefinedProducts = new HashMap<>();**

**static {**

**predefinedProducts.put("8998800100018", new PredefinedProductInfo("Facial Wash Brightening", 35000.0, "Pembersih Wajah", "assets/facialwash\_qeemla\_skin.jpg"));**

**predefinedProducts.put("8998800100025", new PredefinedProductInfo("Serum Qeemla Skin", 50000.0, "Pelembab", "assets/booster serum qeemla skin.png"));**

**predefinedProducts.put("8998800100032", new PredefinedProductInfo("Sunscreen SPF 50 PA+++", 45000.0, "Perlindungan Matahari", "assets/day cream qeemla skin.png"));**

**predefinedProducts.put("8998800100049", new PredefinedProductInfo("Toner", 40000.0, "Perawatan Jerawat", "assets/toner qeemla skin.png"));**

**}**

**public InputProductForm(ProductDAO productDAO, String emailLogin) {**

**super(); // Memanggil constructor JFrame**

**this.productDAO = productDAO;**

**this.emailLogin = emailLogin;**

**setTitle("➕ Input Produk Baru");**

**setSize(800, 750);**

**setMinimumSize(new Dimension(800, 750));**

**setLocationRelativeTo(null);**

**setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);**

**initComponents(); // panggil UI builder**

**}**

**// Constructor untuk mode edit produk**

**public InputProductForm(ProductDAO productDAO, Product productToEdit, String emailLogin) {**

**super(); // Memanggil constructor JFrame**

**this.productDAO = productDAO;**

**this.emailLogin = emailLogin;**

**this.isEditMode = true;**

**this.originalCode = productToEdit.getCode();**

**setTitle("✏️ Edit Produk");**

**setSize(800, 750);**

**setMinimumSize(new Dimension(800, 750));**

**setLocationRelativeTo(null);**

**setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);**

**initComponents(); // panggil UI builder**

**tfBarcode.setText(productToEdit.getCode());**

**tfBarcode.setEditable(false);**

**loadProductData(productToEdit);**

**}**

**private void initComponents() {**

**CustomPanel background = new CustomPanel("assets/logo\_qeemla.png");**

**setContentPane(background);**

**background.setLayout(new BorderLayout());**

**JPanel contentPanel = new JPanel(new GridBagLayout());**

**contentPanel.setOpaque(false);**

**GridBagConstraints gbc = new GridBagConstraints();**

**gbc.insets = new Insets(8, 8, 8, 8);**

**gbc.fill = GridBagConstraints.HORIZONTAL;**

**JLabel titleLabel = new JLabel("Form Tambah Produk");**

**titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 24));**

**titleLabel.setForeground(new Color(6, 94, 84));**

**titleLabel.setHorizontalAlignment(SwingConstants.CENTER);**

**gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 3;**

**contentPanel.add(titleLabel, gbc);**

**gbc.gridwidth = 1;**

**gbc.gridx = 0; gbc.gridy = 1; contentPanel.add(new JLabel("Barcode:"), gbc);**

**tfBarcode = new JTextField(20);**

**gbc.gridx = 1; contentPanel.add(tfBarcode, gbc);**

**JButton scanBarcodeBtn = new JButton("Scan Barcode");**

**styleButton(scanBarcodeBtn, new Color(6, 94, 84), Color.WHITE);**

**gbc.gridx = 2; contentPanel.add(scanBarcodeBtn, gbc);**

**gbc.gridx = 0; gbc.gridy = 2; contentPanel.add(new JLabel("Nama Produk:"), gbc);**

**tfNama = new JTextField(20);**

**gbc.gridx = 1; gbc.gridwidth = 2; contentPanel.add(tfNama, gbc); gbc.gridwidth = 1;**

**gbc.gridx = 0; gbc.gridy = 3; contentPanel.add(new JLabel("Harga:"), gbc);**

**tfHarga = new JTextField(20);**

**gbc.gridx = 1; gbc.gridwidth = 2; contentPanel.add(tfHarga, gbc); gbc.gridwidth = 1;**

**gbc.gridx = 0; gbc.gridy = 4; contentPanel.add(new JLabel("Kategori:"), gbc);**

**tfKategori = new JTextField(20);**

**gbc.gridx = 1; gbc.gridwidth = 2; contentPanel.add(tfKategori, gbc); gbc.gridwidth = 1;**

**gbc.gridx = 0; gbc.gridy = 5; contentPanel.add(new JLabel("Stok:"), gbc);**

**tfStok = new JTextField(20);**

**gbc.gridx = 1; gbc.gridwidth = 2; contentPanel.add(tfStok, gbc); gbc.gridwidth = 1;**

**gbc.gridx = 0; gbc.gridy = 6; contentPanel.add(new JLabel("Tanggal Produksi:"), gbc);**

**spinnerProduksi = new JSpinner(new SpinnerDateModel(new Date(), null, null, Calendar.DAY\_OF\_MONTH));**

**spinnerProduksi.setEditor(new JSpinner.DateEditor(spinnerProduksi, "yyyy-MM-dd"));**

**gbc.gridx = 1; gbc.gridwidth = 2; contentPanel.add(spinnerProduksi, gbc); gbc.gridwidth = 1;**

**gbc.gridx = 0; gbc.gridy = 7; contentPanel.add(new JLabel("Tanggal Expired:"), gbc);**

**spinnerExpired = new JSpinner(new SpinnerDateModel(new Date(), null, null, Calendar.DAY\_OF\_MONTH));**

**spinnerExpired.setEditor(new JSpinner.DateEditor(spinnerExpired, "yyyy-MM-dd"));**

**gbc.gridx = 1; gbc.gridwidth = 2; contentPanel.add(spinnerExpired, gbc); gbc.gridwidth = 1;**

**gbc.gridx = 0; gbc.gridy = 8; contentPanel.add(new JLabel("Foto Produk:"), gbc);**

**photoLabel = new JLabel("Tidak ada foto", SwingConstants.CENTER);**

**photoLabel.setPreferredSize(new Dimension(100, 100));**

**photoLabel.setBorder(BorderFactory.createLineBorder(Color.GRAY));**

**gbc.gridx = 1; contentPanel.add(photoLabel, gbc);**

**JButton choosePhotoBtn = new JButton("Pilih Foto");**

**styleButton(choosePhotoBtn, new Color(6, 94, 84), Color.WHITE);**

**gbc.gridx = 2; contentPanel.add(choosePhotoBtn, gbc);**

**webcamControlPanel = new JPanel(new FlowLayout());**

**webcamControlPanel.setOpaque(false);**

**startWebcamBtn = new JButton("Mulai Webcam");**

**stopWebcamBtn = new JButton("Berhenti Webcam");**

**captureBtn = new JButton("Ambil Gambar");**

**styleButton(startWebcamBtn, new Color(46, 139, 87), Color.WHITE);**

**styleButton(stopWebcamBtn, new Color(178, 34, 34), Color.WHITE);**

**styleButton(captureBtn, new Color(65, 105, 225), Color.WHITE);**

**webcamControlPanel.add(startWebcamBtn);**

**webcamControlPanel.add(stopWebcamBtn);**

**webcamControlPanel.add(captureBtn);**

**gbc.gridx = 0; gbc.gridy = 9; gbc.gridwidth = 3;**

**contentPanel.add(webcamControlPanel, gbc);**

**webcam = Webcam.getDefault();**

**if (webcam != null) {**

**webcam.setViewSize(WebcamResolution.QVGA.getSize());**

**webcamPanel = new WebcamPanel(webcam);**

**gbc.gridx = 0; gbc.gridy = 10;**

**gbc.gridwidth = 3;**

**gbc.fill = GridBagConstraints.CENTER;**

**gbc.weightx = 1.0;**

**gbc.weighty = 1.0;**

**contentPanel.add(webcamPanel, gbc);**

**}**

**JPanel buttonPanel = new JPanel(new FlowLayout());**

**buttonPanel.setOpaque(false);**

**JButton simpanBtn = new JButton("Simpan Produk");**

**JButton batalBtn = new JButton("Batal");**

**styleButton(simpanBtn, new Color(6, 94, 84), Color.WHITE);**

**styleButton(batalBtn, new Color(220, 20, 60), Color.WHITE);**

**buttonPanel.add(simpanBtn);**

**buttonPanel.add(batalBtn);**

**JScrollPane scrollPane = new JScrollPane(contentPanel);**

**scrollPane.setOpaque(false);**

**scrollPane.getViewport().setOpaque(false);**

**scrollPane.setBorder(BorderFactory.createEmptyBorder());**

**background.add(scrollPane, BorderLayout.CENTER);**

**background.add(buttonPanel, BorderLayout.SOUTH);**

**scanBarcodeBtn.addActionListener(e -> startBarcodeScanner());**

**choosePhotoBtn.addActionListener(e -> choosePhoto());**

**simpanBtn.addActionListener(e -> simpanProduk());**

**batalBtn.addActionListener(e -> dispose());**

**startWebcamBtn.addActionListener(e -> startWebcam());**

**stopWebcamBtn.addActionListener(e -> stopWebcam());**

**captureBtn.addActionListener(e -> capturePhotoFromWebcam());**

**addWindowListener(new WindowAdapter() {**

**public void windowClosing(WindowEvent e) {**

**stopWebcam();**

**}**

**});**

**tfBarcode.addKeyListener(new KeyAdapter() {**

**public void keyReleased(KeyEvent e) {**

**if (e.getKeyCode() == KeyEvent.VK\_ENTER) {**

**fillProductInfo(tfBarcode.getText());**

**}**

**}**

**});**

**}**

**private void styleButton(JButton button, Color bgColor, Color fgColor) {**

**button.setBackground(bgColor);**

**button.setForeground(fgColor);**

**button.setFocusPainted(false);**

**button.setBorder(BorderFactory.createEmptyBorder(8, 15, 8, 15));**

**button.setFont(new Font("Segoe UI", Font.BOLD, 12));**

**}**

**private void startBarcodeScanner() {**

**if (webcam == null || !webcam.isOpen()) {**

**JOptionPane.showMessageDialog(this, "Webcam belum dimulai. Harap mulai webcam terlebih dahulu.", "Error", JOptionPane.ERROR\_MESSAGE);**

**return;**

**}**

**if (barcodeScannerThread != null && barcodeScannerThread.isAlive()) {**

**JOptionPane.showMessageDialog(this, "Scanner barcode sudah berjalan.", "Info", JOptionPane.INFORMATION\_MESSAGE);**

**return;**

**}**

**barcodeScannerThread = new Thread(() -> {**

**boolean barcodeFound = false;**

**while (!barcodeFound && !Thread.currentThread().isInterrupted()) {**

**try {**

**Thread.sleep(100);**

**} catch (InterruptedException ex) {**

**Thread.currentThread().interrupt();**

**break;**

**}**

**Result result = null;**

**BufferedImage image = null;**

**if (webcam.isOpen()) {**

**image = webcam.getImage();**

**}**

**if (image != null) {**

**LuminanceSource source = new BufferedImageLuminanceSource(image);**

**BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));**

**try {**

**result = new MultiFormatReader().decode(bitmap);**

**} catch (NotFoundException e) {**

**// No barcode found**

**}**

**}**

**if (result != null) {**

**String barcode = result.getText();**

**SwingUtilities.invokeLater(() -> {**

**tfBarcode.setText(barcode);**

**fillProductInfo(barcode);**

**});**

**barcodeFound = true;**

**}**

**}**

**if (barcodeFound) {**

**JOptionPane.showMessageDialog(InputProductForm.this, "Barcode berhasil dipindai: " + tfBarcode.getText(), "Scan Berhasil", JOptionPane.INFORMATION\_MESSAGE);**

**} else if (!Thread.currentThread().isInterrupted()) {**

**JOptionPane.showMessageDialog(InputProductForm.this, "Scanning barcode dihentikan.", "Info", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**});**

**barcodeScannerThread.start();**

**JOptionPane.showMessageDialog(this, "Mulai scanning barcode...", "Info", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**// Ganti metode fillProductInfo Anda dengan yang ini**

**private void fillProductInfo(String barcode) {**

**PredefinedProductInfo info = predefinedProducts.get(barcode);**

**if (info != null) {**

**// Mengisi field yang sudah ada**

**tfNama.setText(info.name);**

**tfHarga.setText(String.valueOf(info.price));**

**tfKategori.setText(info.category);**

**// --- LOGIKA PENGATURAN TANGGAL OTOMATIS ---**

**// Tanggal hari ini**

**Date today = new Date();**

**// Atur tanggal produksi ke hari ini**

**spinnerProduksi.setValue(today);**

**// Hitung tanggal satu tahun dari sekarang untuk tanggal expired**

**Calendar cal = Calendar.getInstance();**

**cal.setTime(today); // Mulai dari hari ini**

**cal.add(Calendar.YEAR, 1); // Tambahkan 1 tahun**

**Date expiryDate = cal.getTime(); // Dapatkan hasilnya sebagai objek Date**

**// Atur tanggal expired**

**spinnerExpired.setValue(expiryDate);**

**// Logika untuk menampilkan foto (tidak berubah)**

**if (info.imagePath != null && !info.imagePath.isEmpty()) {**

**File photoFile = new File(info.imagePath);**

**if (photoFile.exists()) {**

**selectedPhoto = photoFile;**

**displayPhoto(selectedPhoto);**

**} else {**

**System.err.println("Predefined photo not found: " + info.imagePath);**

**photoLabel.setIcon(null);**

**photoLabel.setText("Foto tidak ditemukan");**

**}**

**} else {**

**photoLabel.setIcon(null);**

**photoLabel.setText("Tidak ada foto");**

**}**

**} else {**

**// Jika barcode tidak ditemukan, kosongkan semua field**

**tfNama.setText("");**

**tfHarga.setText("");**

**tfKategori.setText("");**

**photoLabel.setIcon(null);**

**photoLabel.setText("Tidak ada foto");**

**// Kosongkan juga tanggal atau set ke default**

**spinnerProduksi.setValue(new Date());**

**spinnerExpired.setValue(new Date());**

**JOptionPane.showMessageDialog(this, "Produk dengan barcode " + barcode + " tidak ditemukan di data predefinisi.", "Info", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**}**

**private void startWebcam() {**

**if (webcam != null && !webcam.isOpen()) {**

**new Thread(() -> {**

**webcam.open();**

**SwingUtilities.invokeLater(() -> {**

**if (webcamPanel != null) webcamPanel.start();**

**JOptionPane.showMessageDialog(this, "Webcam berhasil dimulai.", "Info", JOptionPane.INFORMATION\_MESSAGE);**

**});**

**}).start();**

**} else if (webcam == null) {**

**JOptionPane.showMessageDialog(this, "Tidak ada webcam ditemukan.", "Error", JOptionPane.ERROR\_MESSAGE);**

**} else {**

**JOptionPane.showMessageDialog(this, "Webcam sudah berjalan.", "Info", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**}**

**private void stopWebcam() {**

**if (barcodeScannerThread != null && barcodeScannerThread.isAlive()) {**

**barcodeScannerThread.interrupt();**

**barcodeScannerThread = null;**

**}**

**if (webcam != null && webcam.isOpen()) {**

**new Thread(() -> {**

**if (webcamPanel != null) webcamPanel.stop();**

**webcam.close();**

**SwingUtilities.invokeLater(() -> {**

**JOptionPane.showMessageDialog(this, "Webcam berhasil dihentikan.", "Info", JOptionPane.INFORMATION\_MESSAGE);**

**});**

**}).start();**

**}**

**}**

**private void capturePhotoFromWebcam() {**

**if (webcam != null && webcam.isOpen()) {**

**try {**

**BufferedImage image = webcam.getImage();**

**if (image != null) {**

**File uploadsDir = new File("uploads");**

**if (!uploadsDir.exists()) {**

**uploadsDir.mkdirs();**

**}**

**String fileName = "product\_" + System.currentTimeMillis() + ".png";**

**selectedPhoto = new File(uploadsDir, fileName);**

**ImageIO.write(image, "PNG", selectedPhoto);**

**displayPhoto(selectedPhoto);**

**JOptionPane.showMessageDialog(this, "Foto berhasil diambil!", "Sukses", JOptionPane.INFORMATION\_MESSAGE);**

**} else {**

**JOptionPane.showMessageDialog(this, "Gagal mengambil gambar dari webcam.", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**} catch (Exception e) {**

**JOptionPane.showMessageDialog(this, "Error saat mengambil foto: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);**

**e.printStackTrace();**

**}**

**} else {**

**JOptionPane.showMessageDialog(this, "Webcam tidak aktif.", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**private void choosePhoto() {**

**JFileChooser fileChooser = new JFileChooser();**

**fileChooser.setDialogTitle("Pilih Foto Produk");**

**int result = fileChooser.showOpenDialog(this);**

**if (result == JFileChooser.APPROVE\_OPTION) {**

**selectedPhoto = fileChooser.getSelectedFile();**

**displayPhoto(selectedPhoto);**

**}**

**}**

**private void displayPhoto(File photoFile) {**

**if (photoFile != null && photoFile.exists()) {**

**try {**

**ImageIcon originalIcon = new ImageIcon(photoFile.getAbsolutePath());**

**Image img = originalIcon.getImage().getScaledInstance(100, 100, Image.SCALE\_SMOOTH);**

**photoLabel.setIcon(new ImageIcon(img));**

**photoLabel.setText("");**

**} catch (Exception e) {**

**photoLabel.setIcon(null);**

**photoLabel.setText("Gagal memuat foto");**

**System.err.println("Error displaying photo: " + e.getMessage());**

**}**

**} else {**

**photoLabel.setIcon(null);**

**photoLabel.setText("Tidak ada foto");**

**}**

**}**

**private void simpanProduk() {**

**String code = tfBarcode.getText();**

**String nama = tfNama.getText();**

**String hargaStr = tfHarga.getText();**

**String kategori = tfKategori.getText();**

**String stokStr = tfStok.getText();**

**Date productionDate = (Date) spinnerProduksi.getValue();**

**Date expiryDate = (Date) spinnerExpired.getValue();**

**String photoPath = (selectedPhoto != null) ? selectedPhoto.getAbsolutePath() : null;**

**// Validasi input**

**if (code.isEmpty() || nama.isEmpty() || hargaStr.isEmpty() || kategori.isEmpty() || stokStr.isEmpty()) {**

**JOptionPane.showMessageDialog(this, "Semua field (termasuk Stok) harus diisi!", "Error", JOptionPane.ERROR\_MESSAGE);**

**return;**

**}**

**double harga;**

**int stok;**

**try {**

**harga = Double.parseDouble(hargaStr);**

**stok = Integer.parseInt(stokStr);**

**} catch (NumberFormatException e) {**

**JOptionPane.showMessageDialog(this, "Harga dan Stok harus berupa angka!", "Error", JOptionPane.ERROR\_MESSAGE);**

**return;**

**}**

**try {**

**Product newProduct = new Product(code, nama, kategori, harga, stok, productionDate, expiryDate, photoPath);**

**LogDAO logDAO = new LogDAO(); // Inisialisasi LogDAO**

**if (isEditMode) {**

**productDAO.updateProduct(newProduct);**

**JOptionPane.showMessageDialog(this, "Produk berhasil diperbarui!", "Sukses", JOptionPane.INFORMATION\_MESSAGE);**

**// Simpan log edit**

**logDAO.simpanLog("Edit Produk", "Produk " + nama + " berhasil diperbarui", emailLogin);**

**} else {**

**productDAO.addProduct(newProduct);**

**JOptionPane.showMessageDialog(this, "Produk berhasil disimpan ke database!", "Sukses", JOptionPane.INFORMATION\_MESSAGE);**

**// Simpan log tambah**

**logDAO.simpanLog("Tambah Produk", "Produk " + nama + " berhasil ditambahkan", emailLogin);**

**}**

**clearForm();**

**// Optional: dispose();**

**} catch (SQLException ex) {**

**JOptionPane.showMessageDialog(this, "Error saat menyimpan produk: " + ex.getMessage(), "Database Error", JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

**private void clearForm() {**

**tfBarcode.setText("");**

**tfNama.setText("");**

**tfHarga.setText("");**

**tfKategori.setText("");**

**// --- MEMBERSIHKAN FIELD STOK ---**

**tfStok.setText("");**

**spinnerProduksi.setValue(new Date());**

**spinnerExpired.setValue(new Date());**

**photoLabel.setIcon(null);**

**photoLabel.setText("Tidak ada foto");**

**selectedPhoto = null;**

**tfBarcode.setEditable(true);**

**isEditMode = false;**

**originalCode = null;**

**}**

**private void loadProductData(Product p) {**

**tfBarcode.setText(p.getCode());**

**tfNama.setText(p.getName());**

**tfKategori.setText(p.getCategory());**

**tfHarga.setText(String.valueOf(p.getPrice()));**

**// --- MEMUAT DATA STOK KE FORM ---**

**tfStok.setText(String.valueOf(p.getStock()));**

**spinnerProduksi.setValue(p.getProductionDate());**

**spinnerExpired.setValue(p.getExpiryDate());**

**originalCode = p.getCode();**

**if (p.getPhotoPath() != null) {**

**selectedPhoto = new File(p.getPhotoPath());**

**displayPhoto(selectedPhoto);**

**}**

**}**

**public class CustomPanel extends JPanel {**

**private Image backgroundImage;**

**public CustomPanel(String imagePath) {**

**try {**

**URL imageUrl = getClass().getClassLoader().getResource(imagePath);**

**if (imageUrl != null) {**

**backgroundImage = new ImageIcon(imageUrl).getImage();**

**} else {**

**System.err.println("Background image not found (classpath): " + imagePath);**

**File directFile = new File(imagePath);**

**if (directFile.exists()) {**

**backgroundImage = new ImageIcon(directFile.getAbsolutePath()).getImage();**

**} else {**

**System.err.println("Background image not found (direct path): " + imagePath);**

**backgroundImage = null;**

**}**

**}**

**} catch (Exception e) {**

**System.err.println("Error loading background image: " + imagePath);**

**e.printStackTrace();**

**backgroundImage = null;**

**}**

**}**

**@Override**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**if (backgroundImage != null) {**

**g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);**

**} else {**

**g.setColor(Color.LIGHT\_GRAY);**

**g.fillRect(0, 0, getWidth(), getHeight());**

**}**

**}**

**}**

**}**

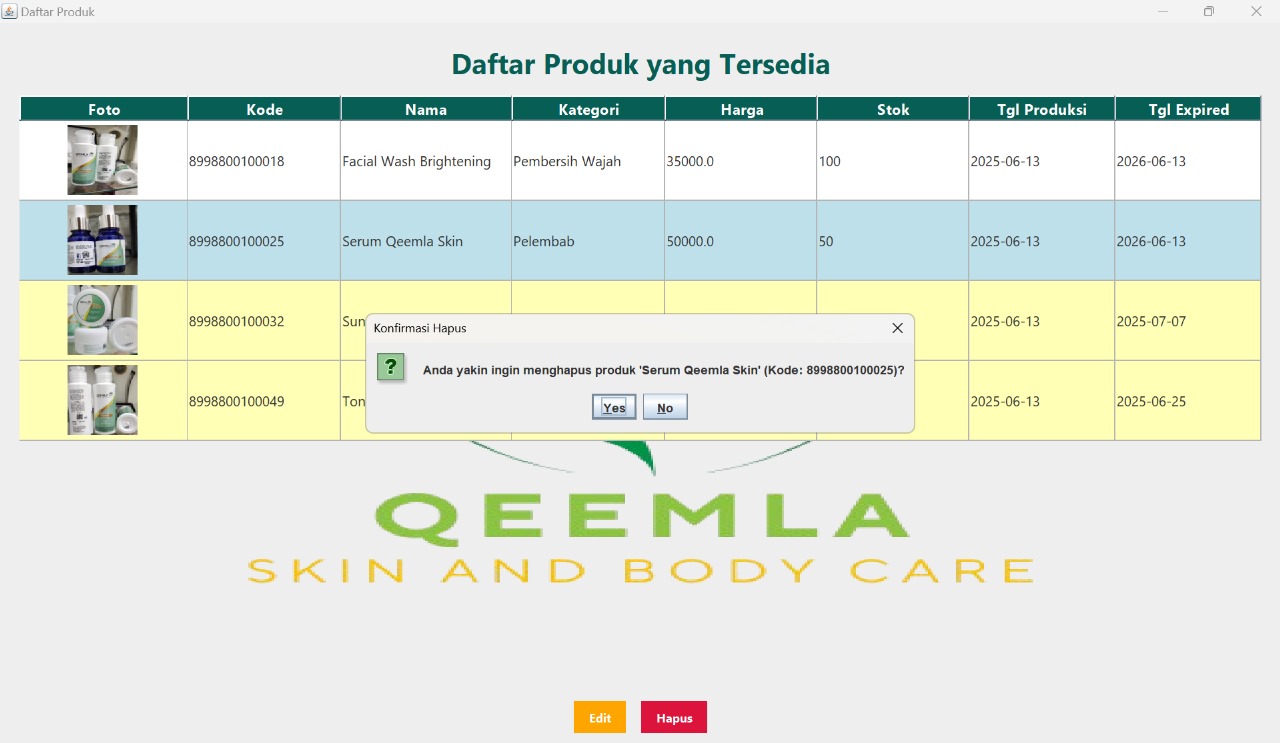
## 5.4 Daftar Produk Qeemla



*Gambar 9. Tampilan Halaman Daftar Produk Qeemla*



*Gambar 10. Tampilan Halaman Daftar Produk Qeemla Untuk Edit Produk*



*Gambar 11. Tampilan Halaman Daftar Produk Qeemla Untuk Konfirmasi Hapus*



*Gambar 12. Tampilan Halaman Daftar Produk Qeemla Produk Berhasil Dihapus*

**import javax.swing.\*;**

**import javax.swing.table.\*;**

**import java.awt.\*;**

**import java.awt.event.WindowAdapter;**

**import java.awt.event.WindowEvent;**

**import java.awt.image.BufferedImage;**

**import java.net.URL;**

**import java.sql.SQLException;**

**import java.text.SimpleDateFormat;**

**import java.util.List;**

**import java.util.Date;**

**import java.io.File;**

**import java.util.concurrent.TimeUnit;**

**public class DaftarProductForm extends JFrame {**

**private JTable table;**

**private DefaultTableModel model;**

**private ProductDAO productDAO;**

**private String emailLogin;**

**public DaftarProductForm(ProductDAO productDAO) {**

**this.productDAO = productDAO;**

**setTitle("Daftar Produk");**

**setSize(1000, 600);**

**setLocationRelativeTo(null);**

**setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);**

**CustomPanel backgroundPanel = new CustomPanel("assets/logo\_qeemla.png");**

**backgroundPanel.setLayout(new BorderLayout());**

**setContentPane(backgroundPanel);**

**JPanel panel = new JPanel(new BorderLayout());**

**panel.setOpaque(false);**

**panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));**

**JLabel titleLabel = new JLabel("Daftar Produk yang Tersedia");**

**titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 28));**

**titleLabel.setForeground(new Color(6, 94, 84));**

**titleLabel.setHorizontalAlignment(SwingConstants.CENTER);**

**titleLabel.setBorder(BorderFactory.createEmptyBorder(0, 0, 15, 0));**

**panel.add(titleLabel, BorderLayout.NORTH);**

**String[] kolom = {"Foto", "Kode", "Nama", "Kategori", "Harga", "Stok", "Tgl Produksi", "Tgl Expired"};**

**model = new DefaultTableModel(kolom, 0) {**

**@Override**

**public boolean isCellEditable(int row, int column) {**

**return false;**

**}**

**@Override**

**public Class<?> getColumnClass(int column) {**

**if (column == 0) return ImageIcon.class;**

**return Object.class;**

**}**

**};**

**table = new JTable(model) {**

**@Override**

**public Component prepareRenderer(TableCellRenderer renderer, int row, int column) {**

**Component c = super.prepareRenderer(renderer, row, column);**

**if (!isRowSelected(row)) {**

**String expiryDateStr = (String) getValueAt(row, 7);**

**if (expiryDateStr != null && !expiryDateStr.isEmpty() && !expiryDateStr.equals("N/A")) {**

**try {**

**SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");**

**Date expiryDate = sdf.parse(expiryDateStr);**

**Date today = new Date();**

**long diffInMillies = expiryDate.getTime() - today.getTime();**

**long diffInDays = TimeUnit.DAYS.convert(diffInMillies, TimeUnit.MILLISECONDS);**

**if (diffInDays < 0) {**

**c.setBackground(new Color(255, 150, 150, 180));**

**} else if (diffInDays <= 30) {**

**c.setBackground(new Color(255, 255, 150, 180));**

**} else {**

**c.setBackground(row % 2 == 0 ? new Color(255, 255, 255, 180) : new Color(240, 248, 255, 180));**

**}**

**} catch (Exception e) {**

**c.setBackground(row % 2 == 0 ? new Color(255, 255, 255, 180) : new Color(240, 248, 255, 180));**

**}**

**} else {**

**c.setBackground(row % 2 == 0 ? new Color(255, 255, 255, 180) : new Color(240, 248, 255, 180));**

**}**

**} else {**

**c.setBackground(new Color(173, 216, 230, 200));**

**}**

**return c;**

**}**

**};**

**table.getColumnModel().getColumn(0).setCellRenderer(new DefaultTableCellRenderer() {**

**@Override**

**public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column) {**

**JLabel label = new JLabel();**

**if (value instanceof ImageIcon) {**

**label.setIcon((ImageIcon) value);**

**}**

**label.setHorizontalAlignment(SwingConstants.CENTER);**

**label.setOpaque(true);**

**label.setBackground(isSelected ? table.getSelectionBackground() : table.getBackground());**

**return label;**

**}**

**});**

**table.getColumnModel().getColumn(0).setPreferredWidth(90);**

**JTableHeader header = table.getTableHeader();**

**header.setFont(new Font("Segoe UI", Font.BOLD, 15));**

**header.setBackground(new Color(6, 94, 84));**

**header.setForeground(Color.WHITE);**

**table.setFont(new Font("Segoe UI", Font.PLAIN, 14));**

**table.setRowHeight(80);**

**table.setGridColor(new Color(180, 180, 180));**

**// ✅ Perubahan responsivitas:**

**table.setAutoResizeMode(JTable.AUTO\_RESIZE\_ALL\_COLUMNS);**

**table.setPreferredScrollableViewportSize(null);**

**JScrollPane scrollPane = new JScrollPane(table);**

**scrollPane.setOpaque(false);**

**scrollPane.getViewport().setOpaque(false);**

**scrollPane.setBorder(BorderFactory.createEmptyBorder());**

**panel.add(scrollPane, BorderLayout.CENTER);**

**JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 15, 0));**

**buttonPanel.setOpaque(false);**

**JButton editButton = new JButton("Edit");**

**styleButton(editButton, new Color(255, 165, 0), Color.WHITE);**

**editButton.addActionListener(e -> editSelectedProduct());**

**JButton deleteButton = new JButton("Hapus");**

**styleButton(deleteButton, new Color(220, 20, 60), Color.WHITE);**

**deleteButton.addActionListener(e -> deleteSelectedProduct());**

**buttonPanel.add(editButton);**

**buttonPanel.add(deleteButton);**

**panel.add(buttonPanel, BorderLayout.SOUTH);**

**backgroundPanel.add(panel, BorderLayout.CENTER);**

**refreshTable();**

**}**

**private void styleButton(JButton button, Color bgColor, Color fgColor) {**

**button.setBackground(bgColor);**

**button.setForeground(fgColor);**

**button.setFocusPainted(false);**

**button.setBorder(BorderFactory.createEmptyBorder(8, 15, 8, 15));**

**button.setFont(new Font("Segoe UI", Font.BOLD, 12));**

**}**

**public void refreshTable() {**

**model.setRowCount(0);**

**SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");**

**try {**

**List<Product> products = productDAO.getAllProducts();**

**for (Product product : products) {**

**String productionDateStr = (product.getProductionDate() != null) ? sdf.format(product.getProductionDate()) : "N/A";**

**String expiryDateStr = (product.getExpiryDate() != null) ? sdf.format(product.getExpiryDate()) : "N/A";**

**ImageIcon imageIcon = null;**

**if (product.getPhotoPath() != null && !product.getPhotoPath().isEmpty()) {**

**File photoFile = new File(product.getPhotoPath());**

**if (photoFile.exists()) {**

**Image img = new ImageIcon(photoFile.getAbsolutePath()).getImage().getScaledInstance(70, 70, Image.SCALE\_SMOOTH);**

**imageIcon = new ImageIcon(img);**

**} else {**

**imageIcon = createPlaceholderImage(70, 70, "No Photo");**

**}**

**} else {**

**imageIcon = createPlaceholderImage(70, 70, "No Photo");**

**}**

**model.addRow(new Object[]{**

**imageIcon,**

**product.getCode(),**

**product.getName(),**

**product.getCategory(),**

**product.getPrice(),**

**product.getStock(),**

**productionDateStr,**

**expiryDateStr**

**});**

**}**

**} catch (SQLException ex) {**

**JOptionPane.showMessageDialog(this, "Error saat memuat daftar produk: " + ex.getMessage(), "Database Error", JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

**private ImageIcon createPlaceholderImage(int width, int height, String text) {**

**BufferedImage placeholder = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_ARGB);**

**Graphics2D g = placeholder.createGraphics();**

**g.setColor(new Color(200, 200, 200, 100));**

**g.fillRect(0, 0, width, height);**

**g.setColor(Color.BLACK);**

**g.setFont(new Font("Segoe UI", Font.PLAIN, 10));**

**FontMetrics fm = g.getFontMetrics();**

**int x = (width - fm.stringWidth(text)) / 2;**

**int y = ((height - fm.getHeight()) / 2) + fm.getAscent();**

**g.drawString(text, x, y);**

**g.dispose();**

**return new ImageIcon(placeholder);**

**}**

**private void editSelectedProduct() {**

**int selectedRow = table.getSelectedRow();**

**if (selectedRow == -1) {**

**JOptionPane.showMessageDialog(this, "Pilih baris yang ingin diedit.", "Peringatan", JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

**String productCode = (String) model.getValueAt(selectedRow, 1);**

**try {**

**Product productToEdit = productDAO.getProductByCode(productCode);**

**if (productToEdit != null) {**

**InputProductForm inputForm = new InputProductForm(productDAO, productToEdit, emailLogin);**

**inputForm.addWindowListener(new WindowAdapter() {**

**@Override**

**public void windowClosed(WindowEvent e) {**

**refreshTable();**

**}**

**});**

**inputForm.setVisible(true);**

**} else {**

**JOptionPane.showMessageDialog(this, "Produk tidak ditemukan.", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**} catch (SQLException ex) {**

**JOptionPane.showMessageDialog(this, "Error saat mengambil data produk: " + ex.getMessage(), "Database Error", JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

**private void deleteSelectedProduct() {**

**int selectedRow = table.getSelectedRow();**

**if (selectedRow == -1) {**

**JOptionPane.showMessageDialog(this, "Pilih baris yang ingin dihapus.", "Peringatan", JOptionPane.WARNING\_MESSAGE);**

**return;**

**}**

**String productCode = (String) model.getValueAt(selectedRow, 1);**

**String productName = (String) model.getValueAt(selectedRow, 2);**

**int confirm = JOptionPane.showConfirmDialog(this,**

**"Anda yakin ingin menghapus produk '" + productName + "' (Kode: " + productCode + ")?",**

**"Konfirmasi Hapus", JOptionPane.YES\_NO\_OPTION);**

**if (confirm == JOptionPane.YES\_OPTION) {**

**try {**

**productDAO.deleteProduct(productCode);**

**refreshTable();**

**JOptionPane.showMessageDialog(this, "Produk berhasil dihapus.", "Sukses", JOptionPane.INFORMATION\_MESSAGE);**

**} catch (SQLException ex) {**

**JOptionPane.showMessageDialog(this, "Error saat menghapus produk: " + ex.getMessage(), "Database Error", JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

**}**

**class CustomPanel extends JPanel {**

**private Image backgroundImage;**

**public CustomPanel(String imagePath) {**

**try {**

**URL imageUrl = getClass().getClassLoader().getResource(imagePath);**

**if (imageUrl != null) {**

**backgroundImage = new ImageIcon(imageUrl).getImage();**

**} else {**

**File directFile = new File(imagePath);**

**if (directFile.exists()) {**

**backgroundImage = new ImageIcon(directFile.getAbsolutePath()).getImage();**

**} else {**

**backgroundImage = null;**

**}**

**}**

**} catch (Exception e) {**

**e.printStackTrace();**

**backgroundImage = null;**

**}**

**}**

**@Override**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**if (backgroundImage != null) {**

**g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);**

**} else {**

**g.setColor(Color.LIGHT\_GRAY);**

**g.fillRect(0, 0, getWidth(), getHeight());**

**}**

**}**

**}**

**}**

## 

## 

## 5.5 Cek Expired Qeemla



*Gambar 13. Tampilan Halaman Cek Expired Produk*

**// File: CekExpiredForm.java**

**import javax.swing.\*;**

**import javax.swing.table.\*;**

**import java.awt.\*;**

**import java.text.SimpleDateFormat;**

**import java.util.List;**

**import java.io.File;**

**import java.sql.SQLException;**

**import java.net.URL;**

**import java.util.Date;**

**import java.util.concurrent.TimeUnit;**

**public class CekExpiredForm extends JFrame {**

**private JTable table;**

**private DefaultTableModel model;**

**private ProductDAO productDAO;**

**private ExpiryChecker expiryChecker;**

**public CekExpiredForm(ProductDAO productDAO) {**

**this.productDAO = productDAO;**

**this.expiryChecker = new ExpiryChecker(productDAO);**

**setTitle("⏰ Produk Kedaluwarsa");**

**setSize(800, 500);**

**setLocationRelativeTo(null);**

**setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);**

**// Gambar background yang kamu upload**

**CustomPanel backgroundPanel = new CustomPanel("assets/logo\_qeemla.png");**

**backgroundPanel.setLayout(new BorderLayout());**

**setContentPane(backgroundPanel);**

**JPanel panel = new JPanel(new BorderLayout());**

**panel.setOpaque(false);**

**panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));**

**JLabel titleLabel = new JLabel("Cek Expired Produk");**

**titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 20));**

**titleLabel.setForeground(new Color(6, 94, 84));**

**titleLabel.setHorizontalAlignment(SwingConstants.CENTER);**

**titleLabel.setBorder(BorderFactory.createEmptyBorder(0, 0, 15, 0));**

**panel.add(titleLabel, BorderLayout.NORTH);**

**String[] kolom = {"Kode", "Nama", "Harga", "Stok", "Tanggal Expired", "Status"};**

**model = new DefaultTableModel(kolom, 0) {**

**@Override**

**public boolean isCellEditable(int row, int column) {**

**return false;**

**}**

**};**

**table = new JTable(model) {**

**@Override**

**public Component prepareRenderer(TableCellRenderer renderer, int row, int column) {**

**Component c = super.prepareRenderer(renderer, row, column);**

**if (!isRowSelected(row)) {**

**String status = (String) getValueAt(row, 5);**

**if (status != null) {**

**if (status.contains("Sudah Kedaluwarsa")) {**

**c.setBackground(new Color(255, 100, 100, 180));**

**} else if (status.contains("BUANG") || status.contains("PROMO")) {**

**c.setBackground(new Color(255, 255, 100, 180));**

**} else {**

**c.setBackground(row % 2 == 0 ? new Color(255, 255, 255, 180) : new Color(240, 248, 255, 180));**

**}**

**}**

**} else {**

**c.setBackground(new Color(173, 216, 230, 200));**

**}**

**return c;**

**}**

**};**

**JTableHeader header = table.getTableHeader();**

**header.setFont(new Font("Segoe UI", Font.BOLD, 15));**

**header.setBackground(new Color(6, 94, 84));**

**header.setForeground(Color.WHITE);**

**table.setFont(new Font("Segoe UI", Font.PLAIN, 14));**

**table.setRowHeight(28);**

**table.setGridColor(new Color(180, 180, 180));**

**JScrollPane scrollPane = new JScrollPane(table);**

**scrollPane.setOpaque(false);**

**scrollPane.getViewport().setOpaque(false);**

**scrollPane.setBorder(BorderFactory.createEmptyBorder());**

**panel.add(scrollPane, BorderLayout.CENTER);**

**backgroundPanel.add(panel, BorderLayout.CENTER);**

**populateExpiredProducts();**

**}**

**public void populateExpiredProducts() {**

**model.setRowCount(0);**

**SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");**

**try {**

**List<Product> products = productDAO.getAllProducts();**

**for (Product product : products) {**

**Date expiryDate = product.getExpiryDate();**

**if (expiryDate != null) {**

**String status = getStatusKadaluarsa(expiryDate);**

**if (status.contains("Kedaluwarsa") || status.contains("BUANG") || status.contains("PROMO")) {**

**model.addRow(new Object[]{**

**product.getCode(),**

**product.getName(),**

**product.getPrice(),**

**product.getStock(),**

**sdf.format(expiryDate),**

**status**

**});**

**}**

**}**

**}**

**} catch (SQLException ex) {**

**JOptionPane.showMessageDialog(this, "Error saat memuat produk dari database: " + ex.getMessage(), "Database Error", JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

**private String getStatusKadaluarsa(Date expiryDate) {**

**try {**

**Date today = new Date();**

**long diffInMillies = expiryDate.getTime() - today.getTime();**

**long days = TimeUnit.DAYS.convert(diffInMillies, TimeUnit.MILLISECONDS);**

**if (days < 0) return "❌ Sudah Kedaluwarsa";**

**else if (days <= 7) return "⚠ BUANG - Exp dalam " + days + " hari";**

**else if (days <= 30) return "🔔 PROMO - Exp dalam " + days + " hari";**

**else return "✅ Aman - Exp dalam " + days + " hari";**

**} catch (Exception e) {**

**e.printStackTrace();**

**return "⛔ Error Cek Tanggal";**

**}**

**}**

**class CustomPanel extends JPanel {**

**private Image backgroundImage;**

**public CustomPanel(String imagePath) {**

**try {**

**File file = new File(imagePath);**

**if (file.exists()) {**

**backgroundImage = new ImageIcon(file.getAbsolutePath()).getImage();**

**} else {**

**URL resource = getClass().getClassLoader().getResource(imagePath);**

**if (resource != null) {**

**backgroundImage = new ImageIcon(resource).getImage();**

**} else {**

**System.err.println("Background image not found: " + imagePath);**

**}**

**}**

**} catch (Exception e) {**

**System.err.println("Error loading background image: " + e.getMessage());**

**}**

**}**

**@Override**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**if (backgroundImage != null) {**

**g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);**

**} else {**

**g.setColor(Color.LIGHT\_GRAY);**

**g.fillRect(0, 0, getWidth(), getHeight());**

**}**

**}**

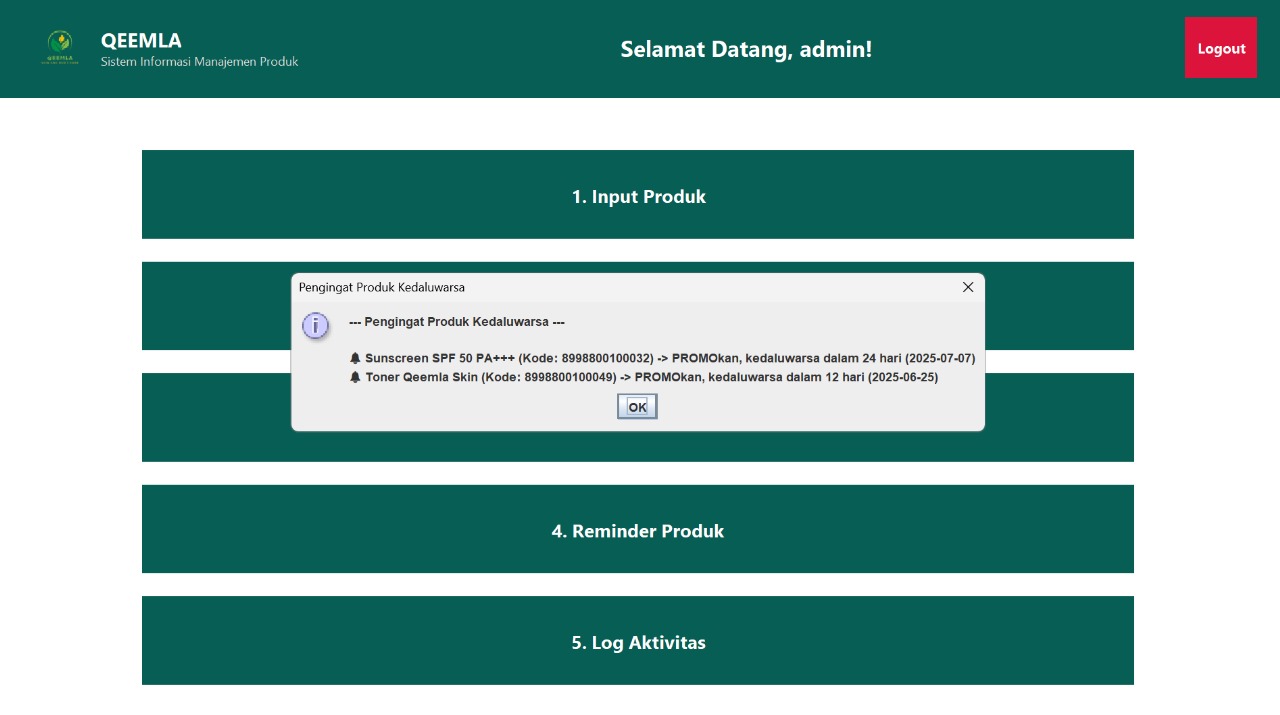
**}**

**}**

## 

## 

## 5.6 Remind Produk Qeemla



*Gambar 14. Tampilan Halaman Remind Produk*

**// File: ExpiryChecker.java**

**import javax.swing.JOptionPane;**

**import java.sql.SQLException;**

**import java.time.LocalDate;**

**import java.time.ZoneId;**

**import java.time.format.DateTimeFormatter;**

**import java.time.temporal.ChronoUnit;**

**import java.util.List;**

**import java.util.Date;**

**public class ExpiryChecker {**

**private ProductDAO productDAO;**

**public ExpiryChecker(ProductDAO productDAO) {**

**this.productDAO = productDAO;**

**}**

**public void showReminder() {**

**StringBuilder output = new StringBuilder();**

**DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");**

**LocalDate today = LocalDate.now();**

**boolean adaReminder = false;**

**List<Product> productList;**

**try {**

**productList = productDAO.getAllProducts();**

**} catch (SQLException e) {**

**e.printStackTrace();**

**JOptionPane.showMessageDialog(null, "Gagal memuat data produk: " + e.getMessage(),**

**"Error", JOptionPane.ERROR\_MESSAGE);**

**return;**

**}**

**if (productList.isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Tidak ada produk dalam daftar.",**

**"Info", JOptionPane.INFORMATION\_MESSAGE);**

**return;**

**}**

**output.append("--- Pengingat Produk Kedaluwarsa ---\n\n");**

**for (Product p : productList) {**

**Date expiryDate = p.getExpiryDate();**

**if (expiryDate == null) {**

**continue;**

**}**

**LocalDate expDate;**

**if (expiryDate instanceof java.sql.Date) {**

**expDate = ((java.sql.Date) expiryDate).toLocalDate();**

**} else {**

**expDate = expiryDate.toInstant().atZone(ZoneId.systemDefault()).toLocalDate();**

**}**

**long daysLeft = ChronoUnit.DAYS.between(today, expDate);**

**if (daysLeft < 0) {**

**output.append("❌ ").append(p.getName())**

**.append(" (Kode: ").append(p.getCode())**

**.append(") -> Sudah KEDALUWARSA pada ")**

**.append(expDate.format(formatter))**

**.append(" (").append(Math.abs(daysLeft)).append(" hari lalu)\n");**

**adaReminder = true;**

**} else if (daysLeft <= 7) {**

**output.append("⚠ ").append(p.getName())**

**.append(" (Kode: ").append(p.getCode())**

**.append(") -> SEGERA BUANG, kedaluwarsa dalam ")**

**.append(daysLeft).append(" hari (").append(expDate.format(formatter)).append(")\n");**

**adaReminder = true;**

**} else if (daysLeft <= 30) {**

**output.append("🔔 ").append(p.getName())**

**.append(" (Kode: ").append(p.getCode())**

**.append(") -> PROMOkan, kedaluwarsa dalam ")**

**.append(daysLeft).append(" hari (").append(expDate.format(formatter)).append(")\n");**

**adaReminder = true;**

**}**

**}**

**if (!adaReminder) {**

**JOptionPane.showMessageDialog(null, "Tidak ada produk yang memerlukan perhatian khusus saat ini.",**

**"Pengingat Produk Kedaluwarsa", JOptionPane.INFORMATION\_MESSAGE);**

**} else {**

**JOptionPane.showMessageDialog(null, output.toString(),**

**"Pengingat Produk Kedaluwarsa", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**}**

**}**

## 5.7 Log Aktivitas

*Gambar 15. Tampilan Halaman Riwayat Log Aktivitas*

**import javax.swing.\*;**

**import javax.swing.table.\*;**

**import java.awt.Color;**

**import java.awt.Font;**

**import java.awt.Component;**

**import java.awt.BorderLayout;**

**import java.sql.Connection;**

**import java.sql.ResultSet;**

**import java.sql.Statement;**

**import java.sql.SQLException;**

**public class LogAuditForm extends JFrame {**

**private JTable table;**

**private DefaultTableModel model;**

**public LogAuditForm() {**

**setTitle("Log Aktivitas");**

**setSize(900, 550);**

**setLocationRelativeTo(null);**

**setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);**

**// Gambar background**

**CustomPanel backgroundPanel = new CustomPanel("assets/logo\_qeemla.png");**

**backgroundPanel.setLayout(new BorderLayout());**

**setContentPane(backgroundPanel);**

**JPanel panel = new JPanel(new BorderLayout());**

**panel.setOpaque(false);**

**panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));**

**JLabel titleLabel = new JLabel("Riwayat Log Aktivitas Pengguna");**

**titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 20));**

**titleLabel.setForeground(new Color(6, 94, 84));**

**titleLabel.setHorizontalAlignment(SwingConstants.CENTER);**

**titleLabel.setBorder(BorderFactory.createEmptyBorder(0, 0, 15, 0));**

**panel.add(titleLabel, BorderLayout.NORTH);**

**String[] columns = {"Waktu", "Aksi", "Detail", "Email Pengguna"};**

**model = new DefaultTableModel(columns, 0) {**

**public boolean isCellEditable(int row, int column) {**

**return false;**

**}**

**};**

**table = new JTable(model) {**

**public Component prepareRenderer(TableCellRenderer renderer, int row, int column) {**

**Component c = super.prepareRenderer(renderer, row, column);**

**if (!isRowSelected(row)) {**

**c.setBackground(row % 2 == 0 ? new Color(255, 255, 255, 180)**

**: new Color(240, 248, 255, 180));**

**} else {**

**c.setBackground(new Color(173, 216, 230, 200));**

**}**

**return c;**

**}**

**};**

**JTableHeader header = table.getTableHeader();**

**header.setFont(new Font("Segoe UI", Font.BOLD, 15));**

**header.setBackground(new Color(6, 94, 84));**

**header.setForeground(Color.WHITE);**

**table.setFont(new Font("Segoe UI", Font.PLAIN, 14));**

**table.setRowHeight(28);**

**table.setGridColor(new Color(180, 180, 180));**

**JScrollPane scrollPane = new JScrollPane(table);**

**scrollPane.setOpaque(false);**

**scrollPane.getViewport().setOpaque(false);**

**scrollPane.setBorder(BorderFactory.createEmptyBorder());**

**panel.add(scrollPane, BorderLayout.CENTER);**

**backgroundPanel.add(panel, BorderLayout.CENTER);**

**loadAuditLog();**

**}**

**private void loadAuditLog() {**

**try (Connection conn = DatabaseManager.getConnection();**

**Statement stmt = conn.createStatement();**

**ResultSet rs = stmt.executeQuery("SELECT waktu\_aksi, aksi, detail, email\_pengguna FROM log\_audit ORDER BY waktu\_aksi DESC")) {**

**while (rs.next()) {**

**model.addRow(new Object[]{**

**rs.getTimestamp("waktu\_aksi"),**

**rs.getString("aksi"),**

**rs.getString("detail"),**

**rs.getString("email\_pengguna")**

**});**

**}**

**} catch (SQLException e) {**

**e.printStackTrace();**

**JOptionPane.showMessageDialog(this, "Gagal memuat data log audit.");**

**}**

**}**

**}**

# BAB VI

# KESIMPULAN DAN SARAN

## 6.1 Kesimpulan

## 

Studi Kasus *"Sistem Manajemen Aplikasi Qeemla Skin & Body Care"* berhasil menunjukkan bagaimana prinsip-prinsip Pemrograman Berorientasi Objek (OOP) diterapkan dalam pengembangan sebuah sistem informasi manajemen produk. Sistem ini dirancang untuk membantu mengelola produk skincare dan body care secara lebih efisien dan terorganisir. Fitur-fitur utama yang dikembangkan, seperti login pengguna, input dan daftar produk, pengecekan masa kadaluarsa (expired check), pengingat otomatis produk kedaluwarsa (remind expired), serta log aktivitas (logaudit) telah berhasil diimplementasikan dengan baik. Penambahan fitur logout juga memberikan nilai tambah dalam hal keamanan dan privasi data pengguna. Dengan adanya sistem ini, manajemen produk menjadi lebih terkontrol, terutama dalam hal pemantauan masa berlaku produk yang krusial dalam industri kosmetik dan perawatan tubuh.

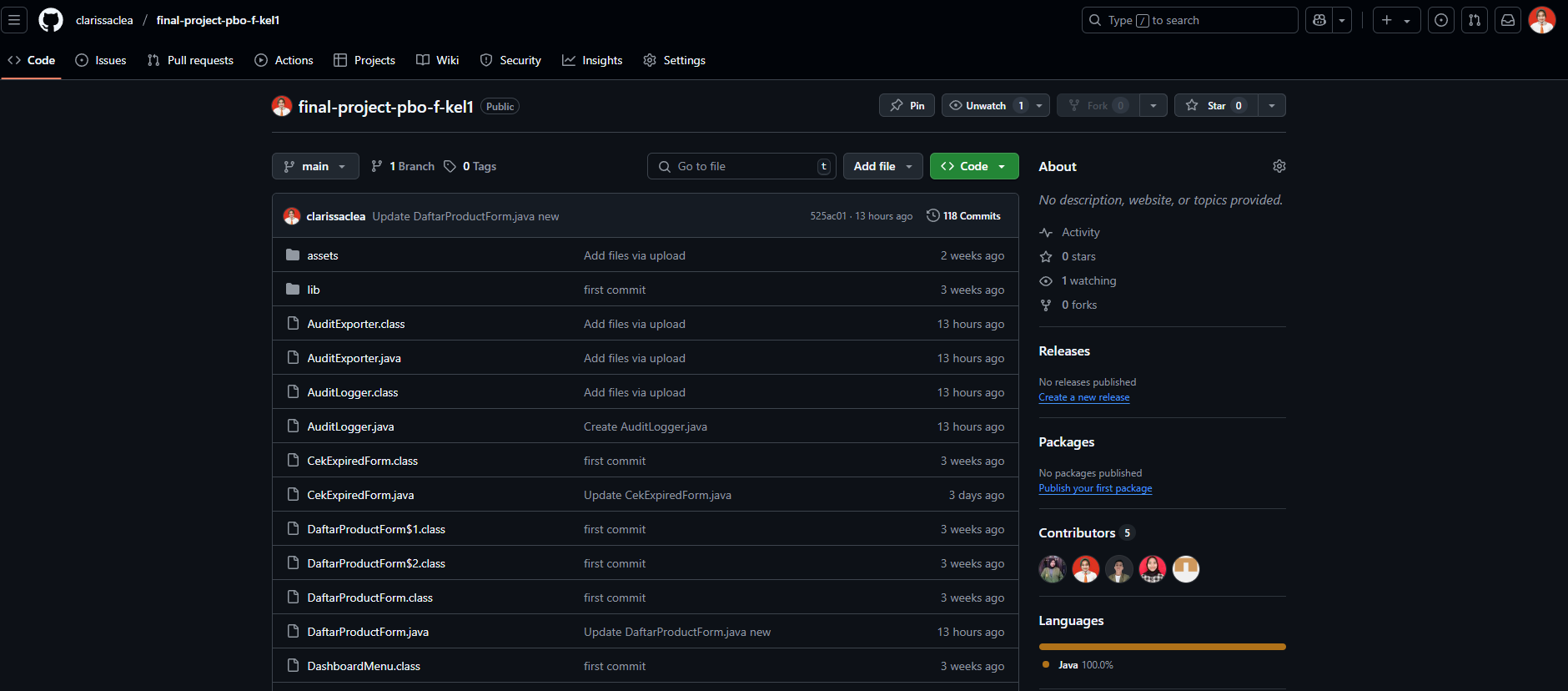
## 6.2 Saran

Untuk pengembangan lebih lanjut, sistem manajemen aplikasi *Qeemla Skin & Body Care* dapat ditingkatkan dari segi fungsionalitas dan pengalaman pengguna. Salah satu saran utama adalah perbaikan antarmuka pengguna (UI/UX) agar tampil lebih menarik dan selaras dengan citra merek skincare modern, sehingga lebih nyaman digunakan oleh staf atau admin. Selain itu, penambahan fitur pencarian dan filter produk akan sangat membantu dalam mempermudah proses pencatatan dan pengelolaan data produk, terutama saat jumlah produk semakin banyak. Disarankan juga untuk mengintegrasikan fitur ekspor data produk dan log aktivitas ke dalam format Excel atau PDF agar memudahkan dalam pembuatan laporan dan dokumentasi. Pengingat masa kedaluwarsa sebaiknya diperluas dengan sistem notifikasi langsung, seperti pop-up atau bahkan melalui email, agar pengguna tidak melewatkan informasi penting. Terakhir, pengembangan sistem multi-level user (dengan role seperti admin, staf, atau manajer) serta pembatasan hak akses juga akan sangat bermanfaat dalam meningkatkan keamanan dan profesionalitas penggunaan sistem ke depannya.

# BAB VII

# LINK GITHUB

[**https://github.com/clarissaclea/final-project-pbo-f-kel1**](https://github.com/clarissaclea/final-project-pbo-f-kel1)

****

*Gambar 16. Tampilan Repository Github*