

**LAPORAN TUGAS BESAR**  
**BBK1JAB3 - PEMROGRAMAN BERORIENTASI OBJEK**  
**SEMESTER GANJIL 2024/2025**

**Nama Studi Kasus:**  
*Administrator of SkinSense*



**Disusun oleh:**  
**Kelompok 1**  
**Ilham Bagaskara - 102022300087**  
**Vanesa Rizka Alfatihah - 102022300121**  
**Abdullah Muchtar Rasyid - 102022300227**  
**Muhammad Zakiyy Mujahid - 102022330243**

**Program Studi S1 Sistem Informasi**  
**Fakultas Rekayasa Industri**  
**Telkom University**  
**2024**

# DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>PEMBAHASAN.....</b>	<b>1</b>
1. Studi Kasus.....	1
2. Class Diagram.....	1
2.1. Deskripsi Kelas.....	3
2.1.1. Kelas MainAPP.....	3
2.1.2. Kelas ProductTask.....	3
2.1.3. Kelas TaskFileManager.....	3
2.1.4. Kelas loginCon.....	3
2.1.5. Kelas crudCon.....	4
2.1.6. Kelas MissingFieldException.....	4
2.1.7. Kelas TaskNotSelectedException.....	4
2.2. Relasi Antarkelas.....	4
2.2.1. Dependency.....	4
2.2.2. Association.....	4
2.2.3. Composition.....	5
2.2.4. Generalization.....	5
2.2.5. Bidirectional Association.....	5
2.2.6. Uses.....	5
3. Use Case.....	5
3.1. Login ke Aplikasi.....	5
3.2. Melihat Daftar Produk.....	8
3.3. Menambahkan Produk Baru.....	10
3.4. Memperbarui Produk yang Ada.....	12
3.5. Menghapus Produk.....	14

3.6. Logout dari Aplikasi.....	16
3.7. Data Kosong saat Menyimpan Produk.....	18
3.8. Format Angka Tidak Valid.....	19
3.9. Tidak Ada Data yang Dipilih untuk Dihapus atau Diperbarui.....	20
3.10. Password Salah saat Login.....	22

## PEMBAHASAN

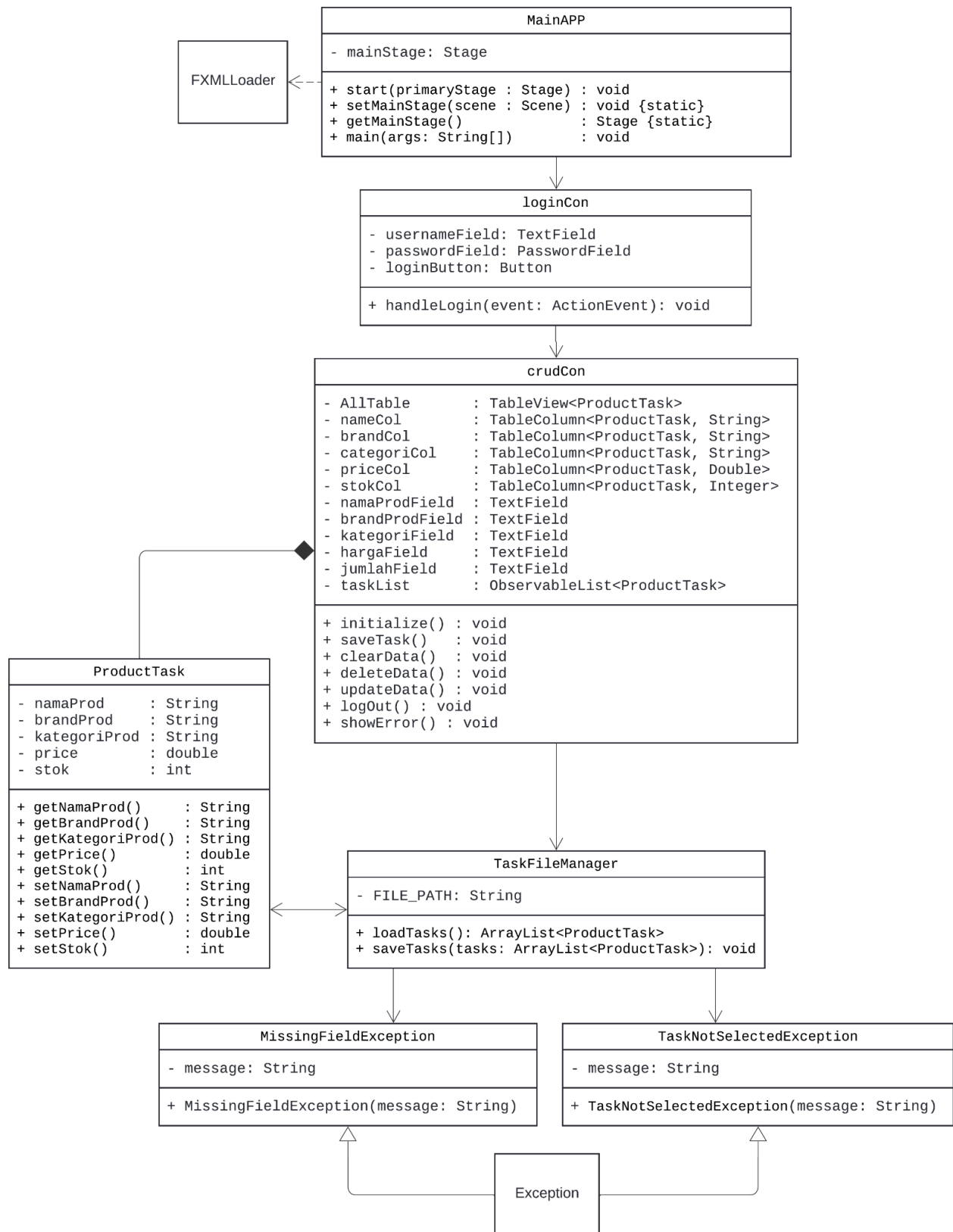
### 1. Studi Kasus

Galuh adalah seorang admin dari aplikasi SkinSense. saat galuh ingin membuka aplikasi SkinSense, Galuh harus memasukan username dan passwordnya untuk menjaga fitur fitur internal dari aplikasi SkinSense. Jika Galuh salah memasukan *password* atau *username*, maka akan keluar notifikasi “*Password* anda salah, silahkan coba lagi”. Jika *username* dan *password*-nya sudah benar, maka aplikasi akan memperlihatkan fitur-fitur admin dari aplikasi SkinSense. Terdapat kolom Nama Produk, *Brand* Produk, Kategori, Harga, dan Jumlah Produk. Kolom tersebut itu bisa Galuh pakai untuk menambahkan, memperbarui, ataupun menghapus produknya. Selain itu terdapat fitur log out jikalau Galuh ingin keluar dari aplikasi SkinSense.

### 2. Class Diagram

Subbab ini akan membahas *Class Diagram* yang akan digunakan untuk menggambarkan struktur kelas dalam sistem berbasis objek, menggunakan standar *Unified Modeling Language* (UML). *Class Diagram* ini akan dikembangkan berdasarkan studi kasus yang sudah dijabarkan sebelumnya, yaitu aplikasi SkinSense. Aplikasi ini mencakup berbagai fitur, seperti pengelolaan data produk, *login* pengguna, serta penyimpanan dan pemuatan data menggunakan *file*. Melalui Class Diagram ini, kita dapat memvisualisasikan hubungan antar kelas, seperti asosiasi, komposisi, serta dependensi antar komponen-komponen yang ada. Dengan menggunakan UML, *Class Diagram* ini diharapkan dapat memberikan gambaran yang jelas tentang desain sistem dan bagaimana objek-objek dalam sistem berinteraksi satu sama lain.

Adapun *Class Diagram* yang digambarkan menggunakan standar UML, untuk menyelesaikan studi kasus di atas, pada Gambar 2.1 berikut:



**Gambar 2.1. UML Class Diagram**

Gambar di atas mendeskripsikan tujuh kelas pada aplikasi ini, yang mana setiap kelas memiliki atribut dan metode tersendiri, serta beragam jenis hubungan antarkelas. Berikut ini penjelasan lebih lanjut mengenai diagram tersebut.

## 2.1. Deskripsi Kelas

### 2.1.1. Kelas MainAPP

- a. Titik masuk aplikasi.
- b. Mengelola *stage* utama dan transisi *scene*.
- c. Menggunakan metode `start()` untuk memuat *form login* (`loginForm.fxml`).

### 2.1.2. Kelas ProductTask

- a. Mewakili produk dengan atribut:
  - 1) `namaProd`: Nama produk.
  - 2) `brandProd`: Nama merek produk.
  - 3) `kategoriProd`: Kategori produk.
  - 4) `price`: Harga produk.
  - 5) `stok`: Jumlah stok produk.
- b. Menyediakan metode *getter* untuk masing-masing *field*.

### 2.1.3. Kelas TaskFileManager

- a. Menangani operasi *file* untuk menyimpan dan memuat tugas produk.
- b. `loadTasks()`: Membaca tugas dari *file* (`tasks.txt`) dan mengonversinya menjadi objek `ProductTask`.
- c. `saveTasks()`: Menulis daftar tugas ke *file*.

### 2.1.4. Kelas loginCon

- a. Kontroler untuk layar *login*.
- b. Memverifikasi input pengguna dengan membandingkan kredensial yang telah ditentukan (admin dan 1234).
- c. Setelah login berhasil, bertransisi ke layar CRUD (`CRUD.fxml`).

### 2.1.5. Kelas crudCon

- a. Kontroler untuk fungsionalitas CRUD.
- b. Mengelola:
  - 1) TableView untuk menampilkan produk.
  - 2) TextFields untuk *input* pengguna.
- c. Metode utama:
  - 1) initialize(): Mengisi TableView dengan data dari TaskFileManager.
  - 2) saveTask(): Menambahkan produk baru ke dalam daftar dan memperbarui *file*.
  - 3) clearData(): Mengosongkan semua *field input*.
  - 4) deleteData(): Menghapus produk yang dipilih dari TableView dan *file*.
  - 5) updateData(): Memperbarui produk yang dipilih dari TableView dan *file*.
  - 6) logOut(): Keluar dari aplikasi dan kembali ke layar *login*.
  - 7) showError(): Menampilkan pesan *error*.

### 2.1.6. Kelas MissingFieldException

- a. *Exception* kustom yang dilemparkan ketika input pengguna tidak lengkap.
- b. Menampilkan pesan kesalahan yang menunjukkan *field* yang hilang.

### 2.1.7. Kelas TaskNotSelectedException

- a. *Exception* kustom yang dilemparkan ketika tidak ada tugas yang dipilih untuk dihapus.
- b. Menampilkan pesan kesalahan yang menunjukkan bahwa pemilihan diperlukan.

## 2.2. Relasi Antarkelas

### 2.2.1. Dependency

- a. MainAPP bergantung pada FXMLLoader untuk memuat *file* FXML.

### 2.2.2. Association

- a. loginCon bertransisi ke crudCon setelah login berhasil.

- b. crudCon berinteraksi dengan ProductTask dan TaskFileManager.

### 2.2.3. Composition

- a. crudCon memiliki hubungan kepemilikan yang kuat dengan ProductTask. Jika crudCon dihapus, objek ProductTask juga akan hilang.

### 2.2.4. Generalization

- a. Digunakan untuk *custom exceptions* (MissingFieldException dan TaskNotSelectedException) yang mewarisi dari kelas *exception* dasar (yang tidak ditunjukkan secara eksplisit).

### 2.2.5. Bidirectional Association

- a. Antara TaskFileManager dan ProductTask, karena TaskFileManager membaca dan menyimpan data ProductTask.

### 2.2.6. Uses

- a. TaskFileManager menggunakan *custom exceptions* seperti MissingFieldException dan TaskNotSelectedException untuk menangani kesalahan.

## 3. Use Case

Berikut ini adalah 11 *use-case* yang dapat terjadi saat menjalankan aplikasi SkinSense ini, lengkap dengan *code*, layar yang ditampilkan, dan keterangan lainnya:

### 3.1. Login ke Aplikasi

Login ke Aplikasi (1/10)	
Code	<pre>import javafx.event.ActionEvent; import javafx.fxml.FXML; import javafx.fxml.FXMLLoader; import javafx.scene.Scene; import javafx.scene.control.Button; import javafx.scene.control.PasswordField;</pre>



## Login ke Aplikasi (1/10)

```
import javafx.scene.control.TextField;
import javafx.stage.Stage;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;

public class loginCon {

    @FXML
    private TextField usernameField;

    @FXML
    private PasswordField passwordField;

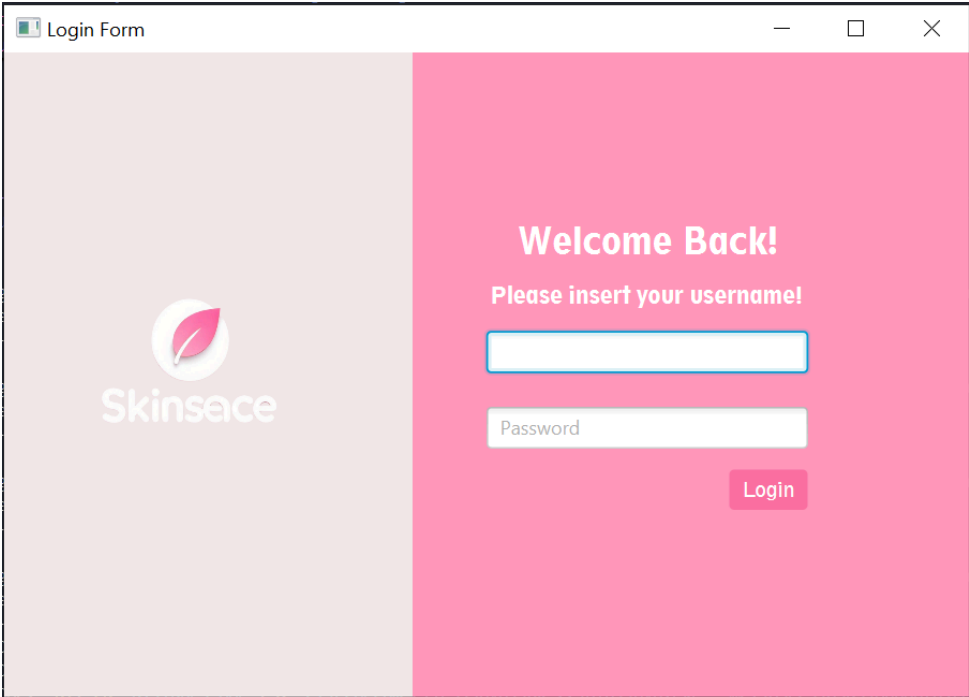
    @FXML
    private Button loginButton;

    @FXML
    private void handleLogin(ActionEvent event) {
        String username = usernameField.getText();
        String password = passwordField.getText();

        if (username.equals("admin") && password.equals("1111")) {
            try {

                FXMLLoader loader = new
FXMLLoader(getClass().getResource("CRUD.fxml"));
                Stage stage =
(Stage)usernameField.getScene().getWindow();
                Scene scene = new Scene(loader.load());

                stage.setScene(scene);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            Alert alert = new Alert(AlertType.ERROR);
            alert.setTitle("ERROR");
            alert.setHeaderText("Password anda salah !");
        }
    }
}
```

Login ke Aplikasi (1/10)	
	<pre>         alert.setContentText("Silahkan coba lagi");         alert.showAndWait();      }  } </pre>
Screenshot	
Notes	<p><b>Prasyarat:</b></p> <ol style="list-style-type: none"> <li>1. Pengguna memiliki kredensial yang valid dan tersimpan di sistem.</li> </ol> <p><b>Alur:</b></p> <ol style="list-style-type: none"> <li>1. Pengguna membuka aplikasi.</li> <li>2. Form login (loginForm.fxml) ditampilkan.</li> <li>3. Pengguna memasukkan username dan password.</li> <li>4. Pengguna menekan tombol login.</li> <li>5. loginCon memvalidasi kredensial.</li> <li>6. Jika berhasil, dashboard utama (crudCon.fxml) ditampilkan.</li> </ol>

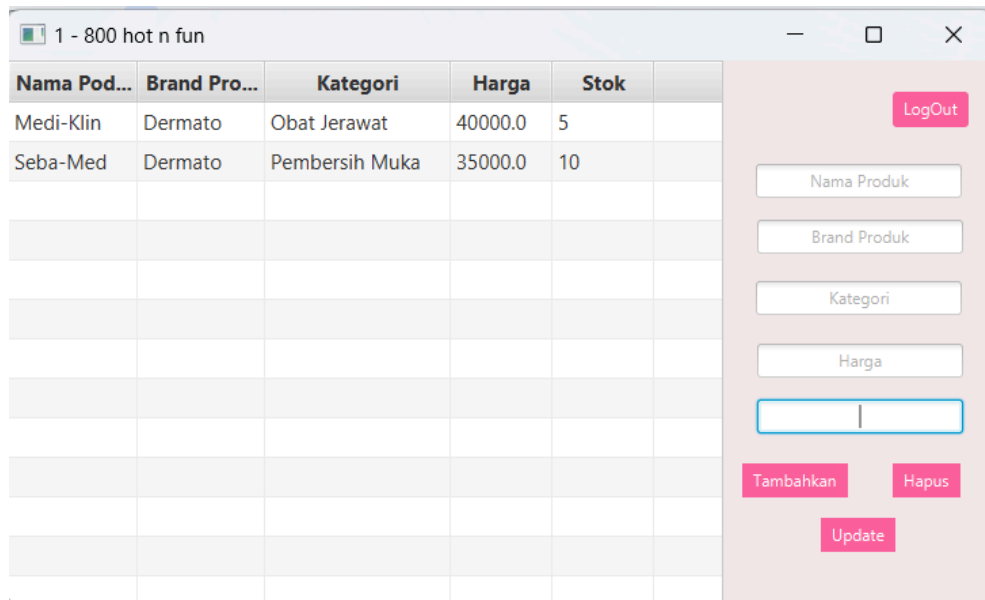
Login ke Aplikasi (1/10)	
	<p><b>Output:</b></p> <ol style="list-style-type: none"> <li>1. Login berhasil akan mengarahkan pengguna ke dashboard utama.</li> <li>2. Login gagal menampilkan pesan kesalahan: "Username atau password salah!"</li> </ol> <p><b>Error Handling:</b></p> <ol style="list-style-type: none"> <li>1. Jika ada field kosong: Alert menampilkan pesan, "Username dan Password tidak boleh kosong!"</li> <li>2. Jika kredensial salah: Alert menampilkan pesan, "Username atau password salah!"</li> <li>3. Jika file/database untuk kredensial hilang: Alert menampilkan pesan, "Sistem login tidak tersedia saat ini."</li> </ol>

### 3.2. Melihat Daftar Produk

Melihat Daftar Produk (2/10)	
Code	<pre> public void initialize() {     taskList = FXCollections.observableArrayList(TaskFileManager.loadTasks());     AllTable.setItems(taskList);      nameCol.setCellValueFactory(new PropertyValueFactory&lt;&gt;("namaProd"));     brandCol.setCellValueFactory(new PropertyValueFactory&lt;&gt;("brandProd"));     kategoriCol.setCellValueFactory(new PropertyValueFactory&lt;&gt;("kategoriProd"));     priceCol.setCellValueFactory(new PropertyValueFactory&lt;&gt;("price"));     stokCol.setCellValueFactory(new PropertyValueFactory&lt;&gt;("stok")); } </pre>

## Melihat Daftar Produk (2/10)

### Screenshot



### Notes

#### Prasyarat:

1. Pengguna telah login.
2. Data produk tersedia dan berhasil dimuat dari TaskFileManager.

#### Alur:

1. Admin membuka dashboard daftar produk (crudCon.fxml).
2. crudCon memuat produk menggunakan TaskFileManager.
3. Semua produk ditampilkan di TableView dengan kolom seperti nama, merek, kategori, harga, dan stok.

#### Output:

1. Daftar produk ditampilkan di tabel.

#### Error Handling:

1. Jika tidak ada produk yang tersedia, tabel akan kosong.
2. Jika gagal memuat data produk: Alert menampilkan pesan, "Gagal memuat data produk!"

### 3.3. Menambahkan Produk Baru

Menambahkan Produk Baru (3/10)	
Code	<pre>public void saveTask() {     try {         if (namaProdField.getText().isEmpty()    brandProdField.getText().isEmpty()                kategoriField.getText().isEmpty()    hargaField.getText().isEmpty()                    jumlahField.getText().isEmpty()) {             throw new MissingFieldException("Data anda kosong, harap masukkan data anda !");         }          ProductTask task = new ProductTask(             namaProdField.getText(),             brandProdField.getText(),             kategoriField.getText(),             Integer.parseInt(hargaField.getText()),             Integer.parseInt(jumlahField.getText()));          taskList.add(task);         TaskFileManager.saveTasks(new ArrayList&lt;&gt;(taskList));      } catch (MissingFieldException e) {         showError(e.getMessage());     } catch (NumberFormatException e) {         showError("Harga dan jumlah harus berupa angka.");     } }</pre>

## Menambahkan Produk Baru (3/10)

### Screenshot

Nama Pod...	Brand Pro...	Kategori	Harga	Stok
Medi-Klin	Dermato	Obat Jerawat	40000.0	5
Seba-Med	Dermato	Pembersih Muka	35000.0	10
Cetaphil	Cetaphil	Pembersih Muka	42000.0	20

LogOut

Cetaphil

Cetaphil

Pembersih Muka

42000

20

Tambahkan Hapus

Update

### Notes

#### Prasyarat:

1. Pengguna telah login.
2. Semua field data produk telah diisi dengan benar.

#### Alur:

1. Admin mengisi detail produk di field (Nama, Merek, Kategori, Harga, Stok).
2. Admin menekan tombol "Simpan".
3. saveTask() memvalidasi input:
  - a. Memastikan semua field terisi.
  - b. Memastikan harga dan stok berupa angka.
4. Produk ditambahkan ke taskList dan disimpan melalui TaskFileManager.
5. TableView diperbarui untuk menampilkan produk baru.

#### Output:

1. Produk baru berhasil ditambahkan ke tabel.

#### Error Handling:

1. Jika ada field kosong: Alert, "Data anda kosong, harap masukkan data anda!"
2. Jika harga atau stok tidak valid: Alert, "Harga dan jumlah harus berupa angka."

### 3.4. Memperbarui Produk yang Ada

Memperbarui Produk yang Ada (4/10)	
Code	<pre>@FXML private void updateData() {     try {         ProductTask selected = AllTable.getSelectionModel().getSelectedItem();         if (selected == null) {             throw new TaskNotSelectedException("Mohon pilih data yang ingin anda rubah!");         }         if(namaProdField.getText().isEmpty()    brandProdField.getText().isEmpty()    kategoriField.getText().isEmpty()    hargaField.getText().isEmpty()    jumlahField.getText().isEmpty()){             throw new MissingFieldException("Data anda kosong tolong masukkan data anda terlebih dahulu!");         }         selected.setNamaProd(namaProdField.getText());         selected.setBrandProd(brandProdField.getText());         selected.setKategori(brandProdField.getText());          selected.setPrice(Integer.parseInt(hargaField.getText()));          selected.setStok(Integer.parseInt(jumlahField.getText()));          TaskFileManager.saveTasks(new ArrayList&lt;&gt;(taskList));         AllTable.refresh();     } catch (MissingFieldException e) {         showError(e.getMessage());     } catch (NumberFormatException e) {         showError("Harga dan jumlah harus berupa angka.");     } catch (TaskNotSelectedException e) {         showError(e.getMessage());     } }</pre>

## Memperbarui Produk yang Ada (4/10)

### Screenshot

The screenshot shows a window titled "Login Form" with a table of products and a form for updating a product. The table has columns: Nama Produk, Brand Produk, Kategori, Harga, and Qty. The form on the right has fields for the same columns, with values from the selected row (Cetaphil) pre-filled. There are buttons for "LogOut", "Tambahkan", "Hapus", and "Update".

Nama Produk	Brand Produk	Kategori	Harga	Qty
Medi-Klin	Dermato	Obat Jera...	40000.0	5
Seba-Med	Dermato	Pembersih...	35000.0	10
Cetaphil	Dermato	Dermato	35000.0	15

Form fields and buttons:

- LogOut
- Cetaphil
- Dermato
- Pembersih Jerawat
- 35000
- 15
- Tambahkan
- Hapus
- Update

### Notes

#### Prasyarat:

1. Pengguna telah login.
2. Produk dipilih dari tabel.
3. Semua field untuk pembaruan telah diisi.

#### Alur:

1. Admin memilih produk dari TableView.
2. Admin mengedit detail produk di field.
3. Admin menekan tombol "Update".
4. updateData() memvalidasi produk yang dipilih dan input field:
  - a. Memastikan semua field terisi.
  - b. Memvalidasi harga dan stok berupa angka.
5. Produk yang dipilih diperbarui dengan nilai baru.
6. TaskFileManager menyimpan daftar yang diperbarui.
7. TableView diperbarui untuk menampilkan perubahan.



### Memperbarui Produk yang Ada (4/10)

**Output:**

1. Detail produk diperbarui di tabel.

**Error Handling:**

1. Jika tidak ada produk yang dipilih: Alert, "Mohon pilih data yang ingin anda rubah!"
2. Jika field kosong: Alert, "Data anda kosong tolong masukkan data anda terlebih dahulu!"
3. Jika harga atau stok tidak valid: Alert, "Harga dan jumlah harus berupa angka."

### 3.5. Menghapus Produk

### Menghapus Produk (5/10)

**Code**

```
@FXML
private void deleteData() {
    try {
        ProductTask selected =
AllTable.getSelectionModel().getSelectedItem();
        if (selected == null) {
            throw new TaskNotSelectedException("Tolong pilih
data yang akan di hapus");
        }
        taskList.remove(selected);
        TaskFileManager.saveTasks(new ArrayList<>());
    } catch (TaskNotSelectedException e) {
        showError(e.getMessage());
    }
}
```

## Menghapus Produk (5/10)

### Screenshot

The screenshot shows a mobile application window titled "1 - 800 hot n fun". It features a table with the following data:

Nama Pod...	Brand Pro...	Kategori	Harga	Stok
Medi-Klin	Dermato	Obat Jerawat	40000.0	5
Seba-Med	Dermato	Pembersih Muka	35000.0	10

To the right of the table is a sidebar containing the following elements:

- A "LogOut" button.
- An empty text input field.
- Input fields for "Brand Produk", "Kategori", "Harga", and "Jumlah Stok".
- Buttons for "Tambahkan", "Hapus", and "Update".

### Notes

#### Prasyarat:

1. Pengguna telah login.
2. Produk dipilih dari tabel.

#### Alur:

1. Admin memilih produk dari TableView.
2. Admin menekan tombol "Delete".
3. deleteData() memvalidasi produk yang dipilih:
  - a. Memastikan sebuah produk dipilih.
4. Produk yang dipilih dihapus dari taskList.
5. TaskFileManager menyimpan daftar yang diperbarui.
6. TableView diperbarui untuk mencerminkan penghapusan.

#### Output:

1. Produk berhasil dihapus dari tabel.

#### Error Handling:

1. Jika tidak ada produk yang dipilih: Alert, "Tolong pilih data yang akan dihapus."

### 3.6. Logout dari Aplikasi

Logout dari Aplikasi (6/10)	
Code	<pre>@FXML private void logOut(){     Stage stage = (Stage) AllTable.getScene().getWindow();     stage.close();     try {         FXMLLoader loader = new FXMLLoader(getClass().getResource("loginForm.fxml"));         Parent root = loader.load();         Stage loginStage = new Stage();         loginStage.setTitle("Login Form");         loginStage.setScene(new Scene(root));         loginStage.show();     } catch (IOException e) {         showError("Gagal untuk logOut dari aplikasi !");     } }</pre>
Screenshot	

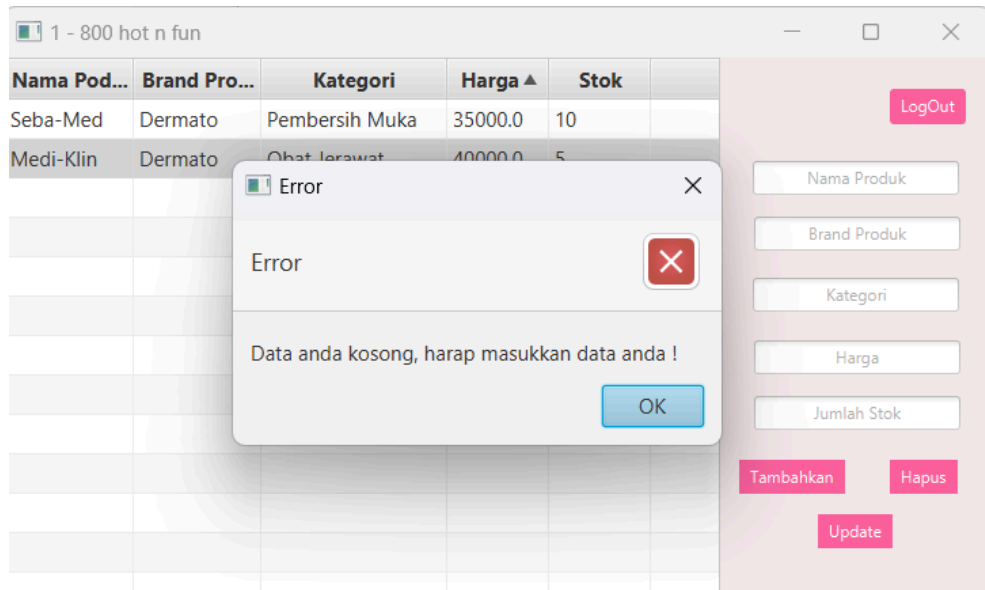
Logout dari Aplikasi (6/10)	
Notes	<p><b>Prasyarat:</b></p> <ol style="list-style-type: none"> <li>1. Pengguna telah login.</li> </ol> <p><b>Alur:</b></p> <ol style="list-style-type: none"> <li>1. Admin menekan tombol "Logout".</li> <li>2. logOut() menutup jendela saat ini.</li> <li>3. Form login (loginForm.fxml) dimuat dan ditampilkan.</li> </ol> <p><b>Output:</b></p> <ol style="list-style-type: none"> <li>1. Pengguna diarahkan kembali ke form login.</li> </ol> <p><b>Error Handling:</b></p> <ol style="list-style-type: none"> <li>1. Jika loginForm.fxml gagal dimuat: Alert, "Gagal untuk logOut dari aplikasi!"</li> </ol>

### 3.7. Data Kosong saat Menyimpan Produk

Data Kosong saat Menyimpan Produk (7/10)	
Code	<pre> if(namaProdField.getText().isEmpty()    brandProdField.getText().isEmpty()    kategoriField.getText().isEmpty()    hargaField.getText().isEmpty()    jumlahField.getText().isEmpty()){     throw new MissingFieldException("Data anda kosong tolong masukkan data anda terlebih dahulu!"); } </pre>

### Data Kosong saat Menyimpan Produk (7/10)

#### Screenshot



#### Notes

##### Prasyarat:

1. Admin mengakses form untuk menambahkan produk.
2. Admin belum mengisi salah satu atau semua field input.

##### Alur:

1. Admin mengklik tombol Simpan tanpa mengisi field input secara lengkap.
2. Sistem memvalidasi form input dan mendeteksi ada field yang kosong.
3. Sistem menampilkan pesan error berupa alert:
4. "Data anda kosong, harap masukkan data anda!"

##### Output:

1. Alert tampil.
2. Data tidak disimpan ke daftar produk.

### 3.8. Format Angka Tidak Valid

#### Format Angka Tidak Valid (8/10)

#### Code

```
if(namaProdField.getText().isEmpty() ||
brandProdField.getText().isEmpty())
```

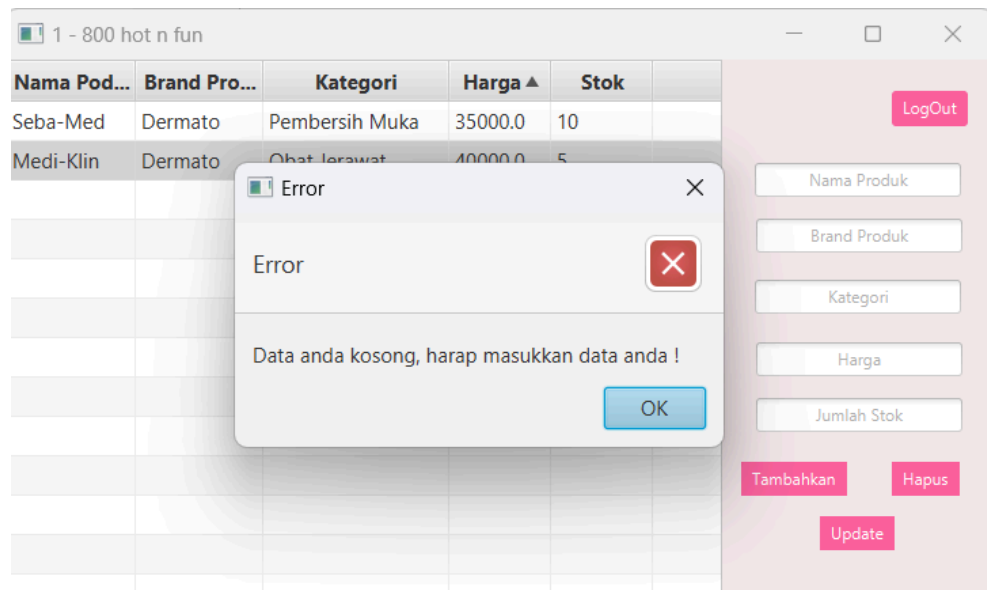
## Format Angka Tidak Valid (8/10)

```

        || kategoriField.getText().isEmpty() ||
hargaField.getText().isEmpty() ||
jumlahField.getText().isEmpty()){
            throw new MissingFieldException("Data anda kosong
tolong masukkan data anda terlebih dahulu!");
        }
    }

```

### Screenshot



### Notes

#### Prasyarat:

1. Admin mengakses form untuk menambahkan atau memperbarui produk.
2. Admin mengisi field Harga atau Jumlah dengan nilai non-numerik.

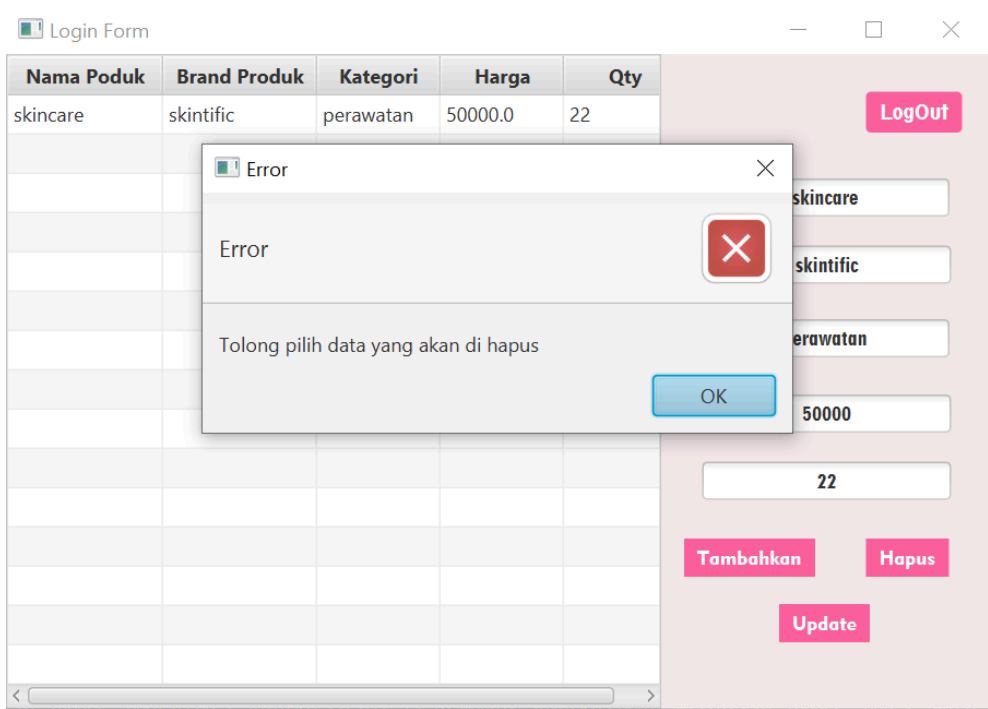
#### Alur:

1. Admin memasukkan teks atau simbol non-angka pada field Harga atau Jumlah.
2. Admin mengklik tombol Simpan atau Update.
3. Sistem mencoba memarsing nilai input sebagai angka, tetapi gagal.
4. Sistem menampilkan pesan error berupa alert:
5. "Harga dan jumlah harus berupa angka."

#### Output:

Format Angka Tidak Valid (8/10)	
	<ol style="list-style-type: none"> <li>1. Alert tampil.</li> <li>2. Data tidak disimpan atau diperbarui.</li> </ol>

### 3.9. Tidak Ada Data yang Dipilih untuk Dihapus atau Diperbarui

Tidak Ada Data yang Dipilih untuk Dihapus (9/10)	
Code	<pre> ProductTask selected = AllTable.getSelectionModel().getSelectedItem();  if (selected == null) {     throw new TaskNotSelectedException("Tolong pilih data yang akan di hapus"); } </pre>
Screenshot	 <p>The screenshot shows a Java application window titled "Login Form". It contains a table with the following columns: Nama Produk, Brand Produk, Kategori, Harga, and Qty. The first row of data is: skincare, skintific, perawatan, 50000.0, 22. To the right of the table are input fields for each column and buttons: LogOut, Tambahkan, Hapus, and Update. An error dialog box is displayed in the foreground with the title "Error" and the message "Tolong pilih data yang akan di hapus".</p>
Notes	<p><b>Prasyarat:</b></p> <ol style="list-style-type: none"> <li>1. Daftar produk tersedia di tabel.</li> <li>2. Admin belum memilih baris data apapun di tabel.</li> </ol>

Tidak Ada Data yang Dipilih untuk Dihapus (9/10)	
	<p><b>Alur:</b></p> <ol style="list-style-type: none"> <li>1. Admin mengklik tombol Hapus atau Update tanpa memilih data pada tabel.</li> <li>2. Sistem memvalidasi pilihan di tabel dan mendeteksi tidak ada data yang dipilih.</li> <li>3. Sistem menampilkan pesan error berupa alert: "Tolong pilih data yang akan dihapus."</li> </ol> <p><b>Output:</b></p> <ol style="list-style-type: none"> <li>1. Alert tampil.</li> <li>2. Tidak ada data yang dihapus dari daftar.</li> </ol>

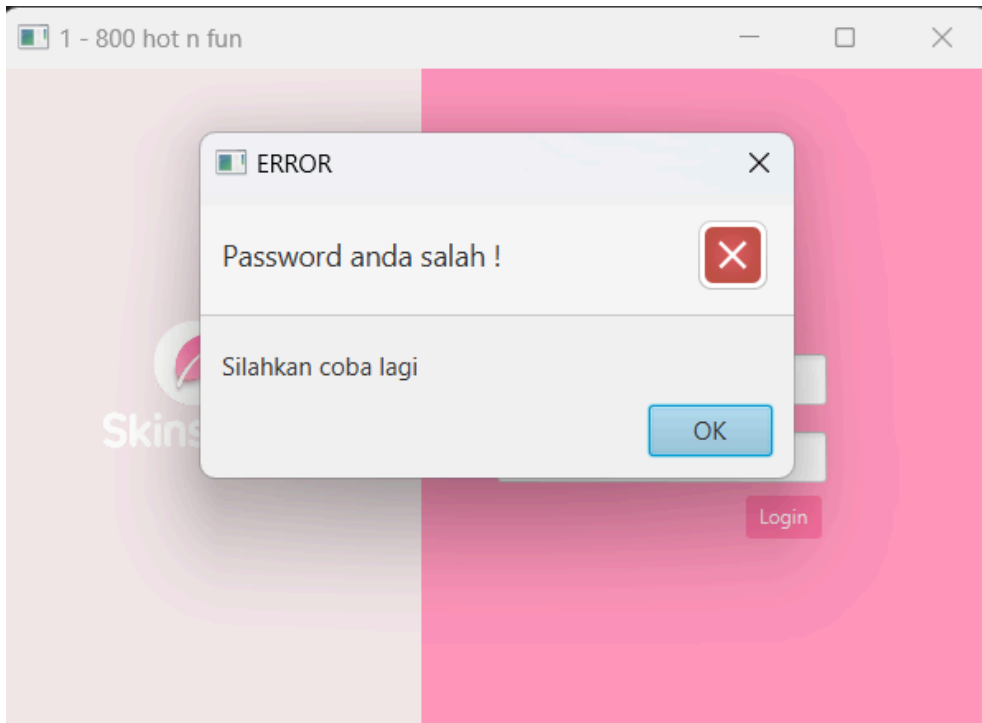
### 3.10. Password Salah saat Login

Password Salah saat Login (10/10)	
Code	<pre> else {     Alert alert = new Alert(AlertType.ERROR);     alert.setTitle("ERROR");     alert.setHeaderText("Password anda salah !");     alert.setContentText("Silahkan coba lagi");     alert.showAndWait(); } </pre>



## Password Salah saat Login (10/10)

### Screenshot



### Notes

#### Prasyarat:

1. Admin sudah berada di halaman login.
2. Admin memasukkan username yang valid tetapi password yang salah.

#### Alur:

1. Admin mengakses halaman login.
2. Admin memasukkan username yang benar tetapi password salah di form login.
3. Admin mengklik tombol Login.
4. Sistem memvalidasi data login dan mendeteksi bahwa kombinasi username dan password tidak sesuai.
5. Sistem menampilkan pesan error berupa alert: "Password salah, silakan coba lagi!"

#### Output:

1. Alert tampil.
2. Admin tetap berada di halaman login.



## **LAMPIRAN**

Berikut ini *link* video presentasi *Administrator of SkinSense* pada YouTube:  
<https://youtu.be/wrbETicz0zM>.