

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **VI.1 Kesimpulan**

Berdasarkan penelitian yang sudah dilaksanakan dan berdasarkan hasil pembahasan pada bab-bab sebelumnya dapat ditarik kesimpulan yaitu :

1. Aplikasi sistem pendukung keputusan pemilihan kosmetik perawatan kulit telah berhasil dibangun.
2. Skin Care Application dapat memberikan alternatif kosmetik menggunakan metode SAW yang dapat menjadi acuan dalam memilih kosmetik perawatan kulit.

#### **VI.2 Saran**

Saran yang dapat diberikan untuk mengembangkan aplikasi ini adalah mengembangkan aplikasi ini agar tidak hanya berbasis *web* saja, tetapi juga berbasis *mobile* sehingga kelak aplikasi ini akan semakin mudah diakses kapan saja dan di mana saja.

## Daftar Pustaka

- Ajiwerdhi, Anak Agung Gde Putra, Kesiman, Made Windu Antara, Wirawan, I Made Agus, 2012, Pengembangan Sistem Pendukung Keputusan Berbasis Mobile Untuk Pengisian Kartu Rencana Studi Denga Fuzzy Multi-Attribute Decision Making (FMADM) Metode Simpl Additive Weighting (SAW), Jurnal Nasional Pendidikan Teknik Informatika, Volume 2.
- Dickey, Philip, 2006, Cosmetics and Personal-Care Products: Avoiding Bodily Harm, Seattle : Washington Toxics Coalition.
- Herdiyan, 2012, INDUSTRI KOSMETIKA: Pasar Berpotensi Tumbuh 10%, [online], (<http://www.bisnis.com/articles/industri-kosmetika-pasar-berpotensi-tumbuh-10-percent>, diakses tanggal 20 July 2012).
- Janko, W. , 2005, Multi-Criteria Decision Making: An Application Study of ELECTRE & TOPSIS , dalam Fuzzy Multi-Attribute Decision Making (FUZZY MADM), Yogyakarta : Graha Ilmu.
- Juju, Dominikus, Studio, MataMaya, 2008, Teknik Mempercepat Koneksi Internet, Jakarta : Elex Media Komputindo.
- Kusumadewi, S. et al. , 2006, Fuzzy Multi-Attribute Decision Making (FUZZY MADM), Yogyakarta : Graha Ilmu.
- Pourvakhshouri, S.Z., B.M. Shattri, Z.I. Zelina, A. Noordin, 2006, Decision Support System Oil Spill Management, International Archives of

- Photogrammetry, Remote Sensing, and Spatial Information Sciences, Vol. XXXVI.
- Putra, Bambang Eka, 2009, Sistem Pendukung Penilaian Keja Karyawan, Skripsi Unpublished, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.
- Ratnaningsih, Rahajeng, 2006, Implementasi Metode K-means Partitioning Clustering Untuk Sistem Pendukung Keputusan Penentuan Jenis Perawatan Kulit.
- Rudolphi, Wictoria, 2000, Multi Criteria Decision Analysis As A Framework For Integrated Land Use Management In Canadian Natinal Parks.
- Ruebush, Mitch, 2005, Comparing SQL Server 2005 and Oracle 10g as a Database Platform for Microsoft .NET Developers.
- Santoso, Leo Willyanto, Budhi Gregorius Satia, Mappatombong Lydiawaty, 2004, Perancangan dan Pembuatan Aplikasi Sistem Pakar Untuk Penentuan Kosmetik dan Jenis Perawatan Tubuh di Pusat Perawatan "Epiderma".
- Sardi, Irawan, 2004, Solusi Bisnis Berbasis Microsoft Office System 2003, Jakarta : Elex Media Komputindo.
- Supardi, Ir. Yuniar, 2010, Web My Profile Dengan Joomla 1.5x, Jakarta : Elex Media Komputindo.
- Tranggono, SpKK., Dr. Retno Iswari, Latifah Apt., Dra. Fatma, 2007, Buku Pegangan Ilmu Pengetahuan Kosmetik, Jakarta : Gramedia Pustaka Utama.
- Turban, Efraim, 2005, Decision Support and Expert Systems. Englewood Cliffs, N.J.: Prentice-Hall,.
- Wardhani, Anita K, 2012, Pakai Kosmetik Sembarangan Picu Penyakit Kulit, [online],

(<http://www.tribunnews.com/2012/02/29/pakai-kosmetik-sembarangan-picu-penyakit-kulit>, diakses tanggal 20 Juli 2012).

Wibowo, Henry, Amalia, Riska, Fadlun M, Andi, Arivanty, Kurnia, 2009, Sistem Pendukung Keputusan Untuk Menentukan Penerima Beasiswa Bank BRI Menggunakan FMAD (Studi Kasus: Mahasiswa Fakultas Teknologi Industri Universitas Islam Indonesia), Yogyakarta : Seminar Nasional Aplikasi Teknologi Informasi 2009, ISSN: 1907-5022.

Yeh, Chun-Shing, 2002, A Problem-based Selection of Multi-Attribute Decision Making Methods, dalam Fuzzy Multi-Attribute Decision Making (FUZZY MADM), Yogyakarta : Graha Ilmu.

Yuhefizar S.Kom, Mooduto, Ir. HA, Hidayat S.T., Rahmat, 2009, Cara Mudah Membangun Website Interaktif Menggunakan Content Management System Joomla Edisi Revisi, Jakarta : Elex Media Komputindo.

Yuhefizar, 2008, 10 jam Menguasai Internet Teknologi dan Aplikasinya, Jakarta : Elex Media Komputindo.

Zimmermann, 1991, Fuzzy Sets Theory and Its Application, dalam Fuzzy Multi-Attribute Decision Making (FUZZY MADM), Yogyakarta : Graha Ilmu.



**LAMP IRAN**

# SKPL

## SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

### Skin Care Application

(SCA)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Helga Laksita Asastani / 5612

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<b>SKPL-SCA</b>		1/38
Fakultas Teknologi Industri				

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperik sa oleh								
Disetuj ui oleh								

## Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

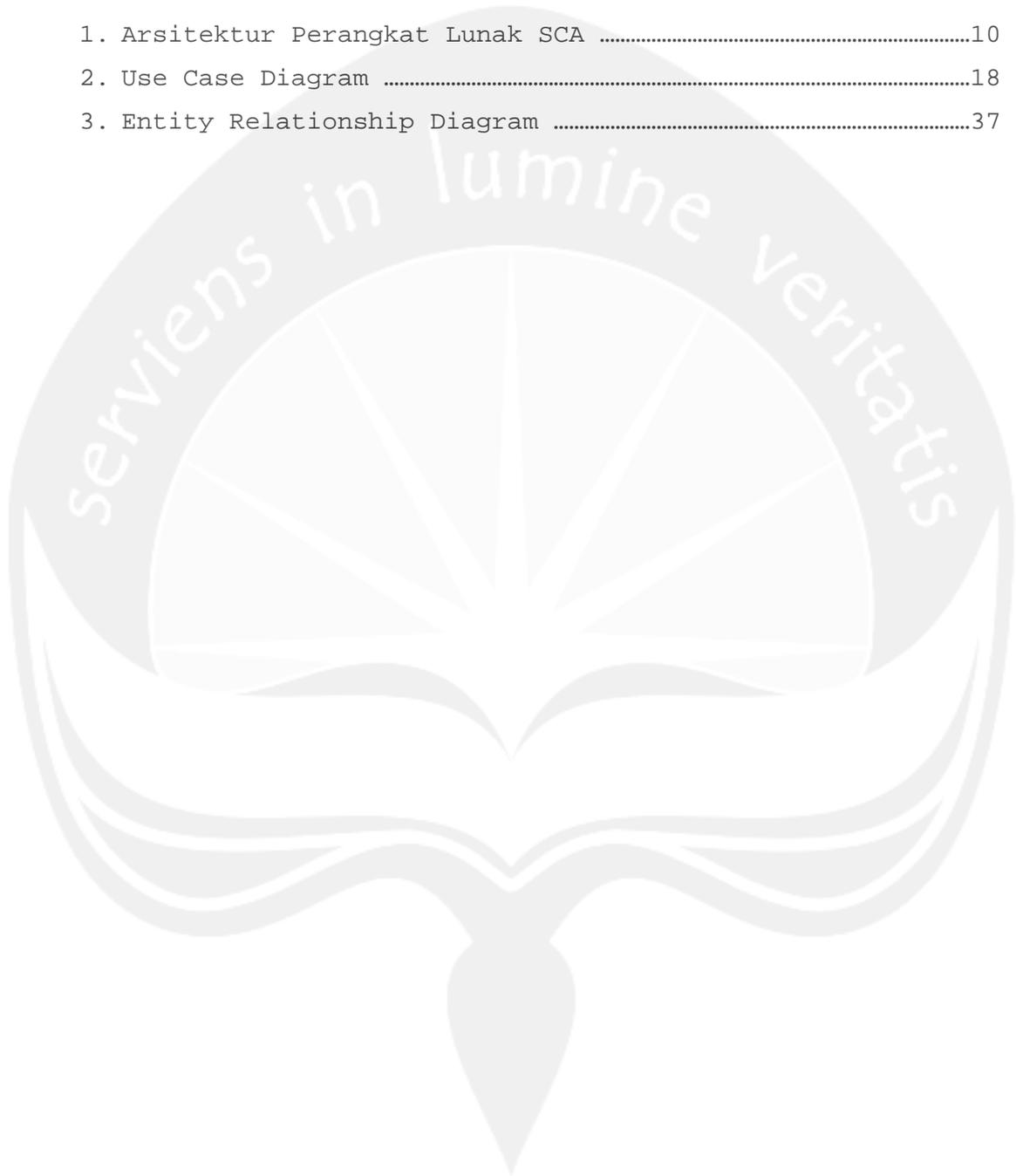


## Daftar Isi

1	Pendahuluan .....	6
1.1	Tujuan .....	6
1.2	Lingkup Masalah .....	6
1.3	Definisi, Akronim dan Singkatan .....	6
1.4	Referensi .....	7
1.5	Deskripsi umum (Overview) .....	8
2	Deskripsi Kebutuhan .....	9
2.1	Perspektif produk .....	9
2.2	Fungsi Produk .....	10
2.3	Karakteristik Pengguna .....	15
2.4	Batasan-batasan .....	16
2.5	Asumsi dan Ketergantungan .....	16
3	Kebutuhan khusus .....	16
3.1	Kebutuhan antarmuka eksternal .....	16
3.2	Kebutuhan fungsionalitas Perangkat Lunak .....	18
4	Spesifikasi Rinci Kebutuhan .....	19
4.1	Spesifikasi Kebutuhan Fungsionalitas .....	19
5	ERD .....	37

## Daftar Gambar

1. Arsitektur Perangkat Lunak SCA .....	10
2. Use Case Diagram .....	18
3. Entity Relationship Diagram .....	37



# 1 Pendahuluan

## 1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SCA (*Skin Care Application*) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras dan pengguna), dan atribut (fitur-fitur tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SCA ini juga mendefinisikan batasan perancangan perangkat lunak.

## 1.2 Lingkup Masalah

Perangkat Lunak SCA dikembangkan dengan tujuan :

1. Menangani pengelolaan data anggota.
2. Menangani pengelolaan data produk.
3. Menangani pengelolaan kategori produk.
4. Menangani pengelolaan pendaftaran anggota.
5. Menangani pengubahan profil anggota.
6. Menangani pengubahan sandi anggota.
7. Menangani pemesanan produk.
8. Menanganai pembayaran pembelian produk.
9. Menangani simulasi pemilihan produk perawatan kulit.

## 1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari

Program Studi Teknik Informatika	SKPL – SCA	6/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

	perangkat lunak yang akan dikembangkan.
SKPL-SCA-XXX	Kode yang merepresentasikan kebutuhan pada SCA ( <i>Skin Care Application</i> ) dimana XXX merupakan nomor fungsi produk.
Anggota	<i>Customer</i> atau pelanggan dari SCA. Untuk menjadi <i>customer</i> , seseorang harus terdaftar sebagai <i>member</i> atau anggota terlebih dahulu.
Administrator	Role yang memberikan hak akses menangani manajemen sistem, termasuk di dalamnya pengelolaan <i>account</i> dari seluruh user yang terdaftar di dalam sistem.
Web based	Menyatakan aplikasi yang dalam penggunaannya pada komputer <i>client</i> dapat diakses menggunakan web browser tanpa memerlukan proses instalasi terlebih dahulu.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

#### 1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Ratriana, Defi, 2011, *Pembangunan Sistem Pendukung Keputusan Wedding Planner*, Skripsi Unpublished, Program Studi Teknik Informatika, Fakultas Teknik Industri, Universitas Atma Jaya, Yogyakarta.
2. Marpaung, Hosanna M.H., 2012, *Pembangunan Aplikasi Online Shopping Berbasis Web*, Skripsi

Unpublished, Program Studi Teknik Informatika,  
Fakultas Teknik Industri, Universitas Atma Jaya,  
Yogyakarta.

### 1.5 Deskripsi umum (Overview)

Dokumen SKPL ini terbagi atas 5 bagian utama. Bagian pertama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SCA yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SCA tersebut.

Bagian ketiga berisi penjelasan tentang kebutuhan khusus perangkat lunak SCA berupa antarmuka eksternal dan fungsionalitas perangkat lunak.

Bagian keempat berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SCA yang akan dikembangkan.

Bagian kelima berisi penggambaran diagram ERD (*Entity Relationship Diagram*) yang merupakan dasar pengembangan perangkat lunak SCA.

Program Studi Teknik Informatika	SKPL – SCA	8/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## 2 Deskripsi Kebutuhan

### 2.1 Perspektif produk

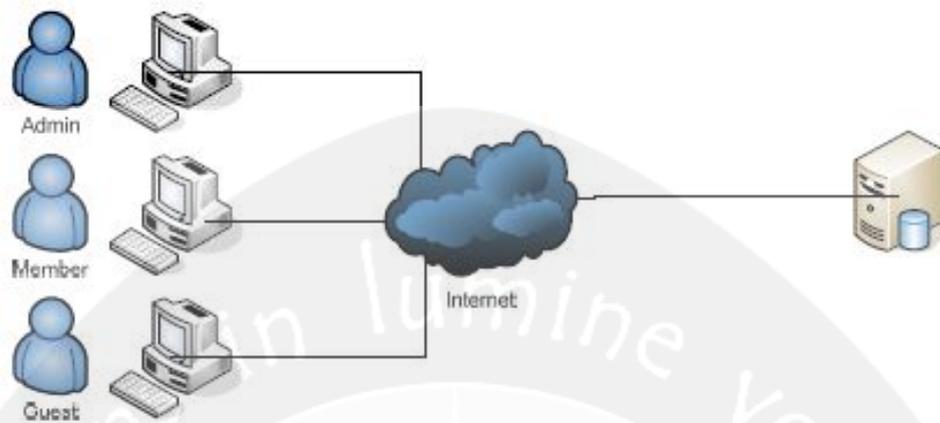
SCA merupakan perangkat lunak yang dikembangkan untuk membantu memutuskan produk perawatan kulit yang tepat dengan menggunakan model *Multiple Attribute Decision Making* yang menerapkan metode *Simple Additive Weighting*. Sistem ini disertakan dalam sebuah penjualan produk kosmetik via Internet (*website*) sehingga diharapkan dapat membantu dalam memilih produk perawatan kulit yang sesuai dengan kebutuhan.

Perangkat lunak SCA ini berjalan pada platform Windows dan dibuat menggunakan bahasa pemrograman Microsoft Visual C#, sedangkan untuk lingkungan pemrogramannya menggunakan Microsoft Visual Studio 2005.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*). Pada sistem ini, seperti terlihat pada gambar 2.1, arsitektur perangkat lunak yang digunakan berupa client server, dimana semua data disimpan di server. User dapat mengakses data yang ada di server tersebut secara online dengan memanggil *web service* pada website yang tersedia di web server arsitektur perangkat lunak yang digunakan berupa 2-tier, dimana semua data disimpan di server.

Inputan data yang dimasukkan akan disimpan dalam database server, sehingga jika ada pencarian data, maka data yang diinginkan akan dicari ke database server yang selanjutnya dikirimkan ke aplikasi *client*.

Program Studi Teknik Informatika	SKPL – SCA	9/37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		



Gambar 2.1 Arsitektur Perangkat lunak SCA

## 2.2 Fungsi Produk

Fungsi produk perangkat lunak SCA untuk anggota sebagai berikut :

### 1. Fungsi *Login* (**SKPL-SCA-001**).

Merupakan fungsi yang digunakan oleh user untuk dapat masuk ke dalam sistem yang akan digunakan.

### 2. Fungsi *Pengelolaan Data Anggota* (**SKPL-SCA-002**).

Merupakan fungsi yang digunakan untuk mengelola data anggota.

Fungsi *Pengelolaan Anggota* mencakup :

#### a. Fungsi *Delete Data Anggota* (**SKPL-SCA-002-01**).

Merupakan fungsi yang digunakan untuk menghapus data anggota yang telah tersimpan dalam database.

#### b. Fungsi *Search Data Anggota* (**SKPL-SCA-002-02**).

Merupakan fungsi yang digunakan untuk mencari

data anggota yang telah tersimpan dalam database.

c. Fungsi *Display Data Anggota* (**SKPL-SCA-002-03**).  
Merupakan fungsi yang digunakan untuk menampilkan data anggota yang telah tersimpan dalam database.

3. Fungsi *Pengelolaan Data Produk* (**SKPL-SCA-003**).

Merupakan fungsi yang digunakan untuk mengelola data produk perawatan yang ada.

a. Fungsi *Entry Data Produk* (**SKPL-SCA-003-01**)

Merupakan fungsi yang digunakan untuk memasukkan data produk baru.

b. Fungsi *Edit Data Produk* (**SKPL-SCA-003-02**).

Merupakan fungsi yang digunakan untuk mengubah data produk yang telah tersimpan dalam database.

c. Fungsi *Delete Data Produk* (**SKPL-SCA-003-03**).

Merupakan fungsi yang digunakan untuk menghapus data produk yang telah tersimpan dalam database.

d. Fungsi *Search Data Produk* (**SKPL-SCA-003-04**).

Merupakan fungsi yang digunakan untuk mencari data produk yang telah tersimpan dalam database.

e. Fungsi *Display Data Produk* (**SKPL-SCA-003-05**).

Merupakan fungsi yang digunakan untuk menampilkan data produk yang telah tersimpan dalam database.

4. Fungsi *Pengelolaan Kategori* (**SKPL-SCA-004**).

Merupakan fungsi yang digunakan untuk mengelola data kategori produk.

a. Fungsi *Entry Kategori* (**SKPL-SCA-004-01**)

Merupakan fungsi yang digunakan untuk memasukkan data kategori baru.

b. Fungsi *Edit Kategori* (**SKPL-SCA-004-02**).

Merupakan fungsi yang digunakan untuk mengubah data kategori produk yang telah tersimpan dalam database.

c. Fungsi *Delete Kategori* (**SKPL-SCA-004-03**).

Merupakan fungsi yang digunakan untuk menghapus data kategori produk yang telah tersimpan dalam database.

d. Fungsi *Display Kategori* (**SKPL-SCA-004-04**).

Merupakan fungsi yang digunakan untuk menampilkan data kategori produk yang telah tersimpan dalam database.

5. Fungsi *Pendaftaran Anggota* (**SKPL-SCA-005**).

Merupakan fungsi yang digunakan untuk mengelola pendaftaran tamu agar dapat menjadi anggota.

6. Fungsi *Profil Anggota* (**SKPL-SCA-006**).

Merupakan fungsi yang digunakan untuk mengelola data atau profil anggota.

a. Fungsi *Lihat Profil* (**SKPL-SCA-006-01**)

Merupakan fungsi yang digunakan untuk menampilkan profil anggota yang tersimpan di database.

b. Fungsi *Ubah Profil* (**SKPL-SCA-006-02**).

Merupakan fungsi yang digunakan untuk mengubah profil anggota.

c. Fungsi *Ubah Password* (**SKPL-SCA-006-03**)

Merupakan fungsi yang digunakan untuk mengubah password anggota.

7. Fungsi *Verifikasi Akun* (**SKPL-SCA-007**).

Merupakan fungsi yang digunakan untuk mengirim email kepada anggota yang berisikan verifikasi kode untuk memverifikasi akun kemudian membuat password yang baru.

8. Fungsi *Simulasi Produk* (**SKPL-SCA-008**).

Merupakan fungsi yang digunakan untuk memilih faktor-faktor dalam membeli produk sesuai jenis kulit dan masalah kulit yang ada.

9. Fungsi *Pemesanan* (**SKPL-SCA-009**).

Merupakan fungsi yang digunakan untuk mengelola data pemesanan.

Fungsi Pengelolaan Pemesanan mencakup :

a. Fungsi *Tambah Pemesanan* (**SKPL-SCA-009-01**).

Merupakan fungsi yang digunakan untuk menambah pemesanan.

b. Fungsi *Edit Pemesanan* (**SKPL-SCA-009-02**).

Program Studi Teknik Informatika	SKPL – SCA	13/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan untuk mengubah pemesanan yang telah tersimpan dalam database.

c. Fungsi *Pembatalan Pemesanan* (**SKPL-SCA-009-03**).

Merupakan fungsi yang digunakan untuk membatalkan pemesanan yang tidak diperlukan.

d. Fungsi *Display Pemesanan* (**SKPL-SCA-009-04**).

Merupakan fungsi yang digunakan untuk menampilkan data pemesanan yang telah tersimpan dalam database.

10. Fungsi *Pengelolaan Pemesanan* (**SKPL-SCA-010**).

Merupakan fungsi yang digunakan untuk mengelola data pesanan.

Fungsi Pengelolaan Pemesanan mencakup :

a. Fungsi *Cari Pesanan* (**SKPL-SCA-010-01**).

Merupakan fungsi yang digunakan untuk mencari data pesanan yang telah tersimpan dalam database.

b. Fungsi *Display Pesanan* (**SKPL-SCA-010-02**).

Merupakan fungsi yang digunakan untuk menampilkan data pesanan yang telah tersimpan dalam database.

c. Fungsi *Delete Pesanan* (**SKPL-SCA-010-03**).

Merupakan fungsi yang digunakan untuk menghapus data pesanan yang telah tersimpan dalam database.

11. Fungsi *Pengelolaan Pembayaran* (**SKPL-SCA-011**).

Merupakan fungsi yang digunakan untuk mengelola data pembayaran.

Fungsi Pengelolaan Pembayaran mencakup :

- a. Fungsi *Entry Pembayaran* (**SKPL-SCA-011-01**).  
Merupakan fungsi yang digunakan untuk menambah data pembayaran suatu barang.
  - b. Fungsi *Delete Pembayaran* (**SKPL-SCA-011-02**).  
Merupakan fungsi yang digunakan untuk menghapus data pembayaran suatu barang.
  - c. Fungsi *Cari Pembayaran* (**SKPL-SCA-011-03**).  
Merupakan fungsi yang digunakan untuk mencari data pembayaran suatu barang.
  - d. Fungsi *Display Pembayaran* (**SKPL-SCA-011-04**).  
Merupakan fungsi yang digunakan untuk menampilkan data pembayaran yang telah tersimpan dalam database.
12. Fungsi *Konfirmasi Pembayaran* (**SKPL-SCA-012**).  
Merupakan fungsi yang digunakan untuk melakukan konfirmasi pembayaran setelah melakukan pembayaran pesanan.

### 2.3 Karakteristik Pengguna

Karakteristik dari pengguna SCA ini adalah sebagai berikut :

1. Administrator
  1. Memahami pengoperasian komputer secara aktif.
  2. Memahami sistem komputer tempat perangkat lunak dijalankan.
  3. Mengerti dan memahami perangkat lunak yang digunakan.
2. Anggota

Program Studi Teknik Informatika	SKPL – SCA	15/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Memahami pengoperasian komputer.
  2. Mengerti penggunaan internet.
3. User
1. Memahami pengoperasian komputer.
  2. Mengerti penggunaan internet.

#### **2.4 Batasan-batasan**

Batasan-batasan dalam pengembangan perangkat lunak SCA tersebut adalah :

1. Kebijakan Umum  
Berpedoman pada tujuan dari pengembangan perangkat lunak SCA.
2. Keterbatasan perangkat keras  
Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

#### **2.5 Asumsi dan Ketergantungan**

Untuk dapat menggunakan aplikasi ini dibutuhkan sistem operasi berplatform Microsoft Windows.

### **3 Kebutuhan khusus**

#### **3.1 Kebutuhan antarmuka eksternal**

Kebutuhan antar muka eksternal pada perangkat lunak SCA meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak.

### 3.1.1 Antarmuka pemakai

Pengguna akan berinteraksi dengan SCA melalui antarmuka yang ditampilkan dalam bentuk web page dalam browser.

### 3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SCA adalah:

1. Perangkat komputer (PC desktop, laptop, netbook, dll).
2. Keyboard
3. Mouse

### 3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SCA adalah sebagai berikut :

1. Nama : SQL Server 2005  
Sumber : Microsoft

Sebagai database management system (DBMS) yang digunakan untuk penyimpanan data di sisi server.

2. Nama : Windows  
Sumber : Microsoft  
Sebagai sistem operasi yang digunakan.

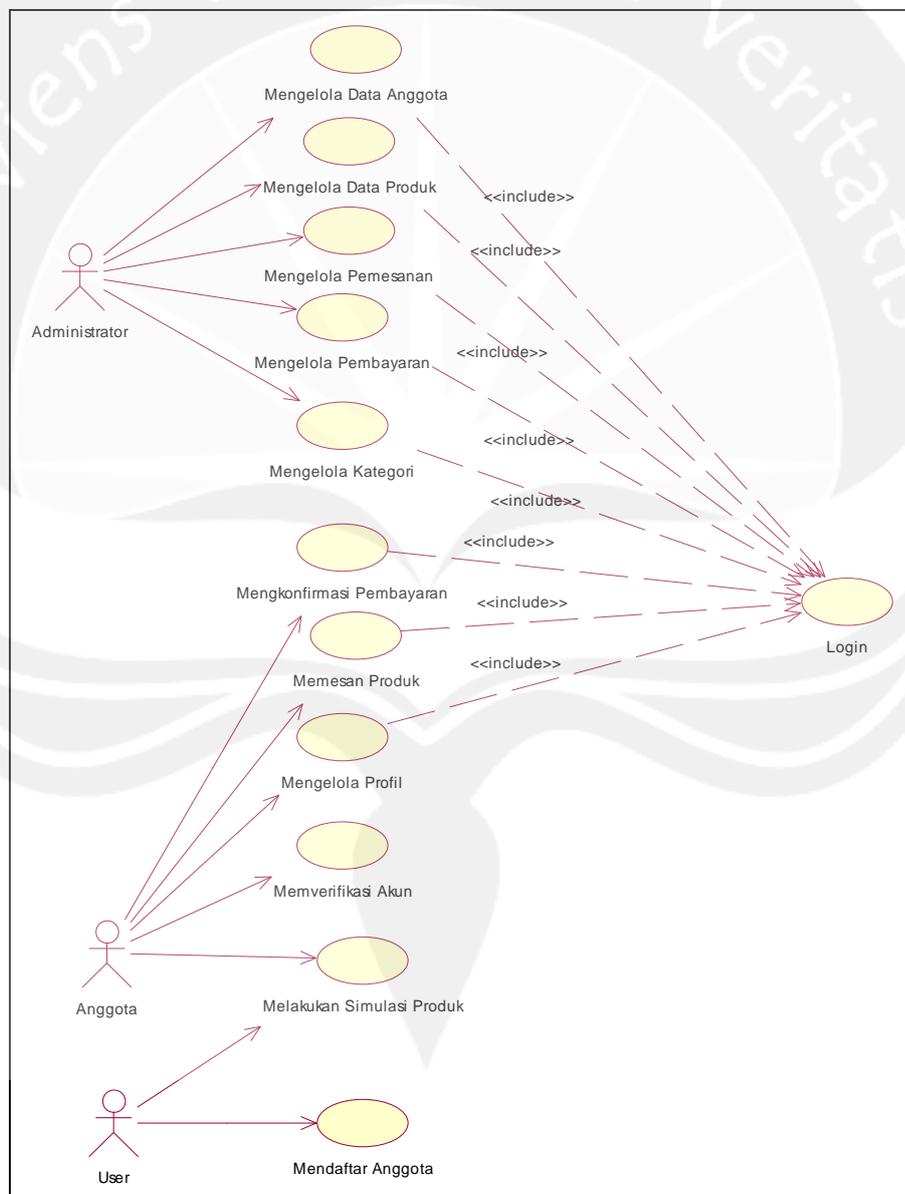
3. Nama : IIS  
Sumber : Microsoft  
Sebagai web server.

4. Nama : Internet Explorer, Mozilla FireFox, Google Chrome, Opera  
Sebagai browser internet.

5. Nama : .NET Compact Framework 2.0  
 Sumber : Microsoft  
 Sebagai framework untuk menjalankan aplikasi.

### 3.2 Kebutuhan fungsionalitas Perangkat Lunak

#### 3.2.1 Use Case Diagram



Gambar 3.1 Use Case Diagram

## **4 Spesifikasi Rinci Kebutuhan**

### **4.1 Spesifikasi Kebutuhan Fungsionalitas**

#### **4.1.1 Use case Spesification : Login**

##### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada nama pengguna dan sandi yang berupa rangkaian karakter.

##### **2. Primary Actor**

1. Administrator

##### **3. Supporting Actor**

1. Anggota

##### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login
2. Sistem menampilkan antarmuka untuk login
3. Aktor memasukkan nama pengguna dan sandi
4. Sistem memeriksa nama pengguna dan sandi yang diinputkan aktor

E-1 Sandi atau nama pengguna tidak sesuai

5. Sistem memberikan akses ke aktor
6. Use Case ini selesai

##### **5. Alternative Flow**

none

##### **6. Error Flow**

E-1 Sandi atau nama pengguna tidak sesuai

1. Sistem menampilkan peringatan bahwa sandi atau nama pengguna tidak sesuai
2. Kembali ke Basic Flow langkah ke 3

##### **7. PreConditions**

1. Aktor telah memiliki nama pengguna dan sandi untuk masuk ke dalam sistem.

Program Studi Teknik Informatika	SKPL – SCA	19/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## 8. PostConditions

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.

### 4.1.2 Use case Spesification : Pengelolaan Data Anggota

#### 1. Brief Description

Use Case ini memungkinkan user untuk melakukan pengelolaan data anggota. Termasuk di dalamnya menghapus anggota, mencari anggota, dan menampilkan data anggota.

#### 2. Primary Actor

1. Administrator

#### 3. Supporting Actor

none

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data anggota.
2. Sistem memberikan pilihan untuk Display Data Anggota, Delete Data Anggota, Search Data Anggota.
3. Aktor memilih untuk melakukan Display Data Anggota
  - A-1 Aktor memilih untuk melakukan Delete Data Anggota
  - A-2 Aktor memilih untuk Search Data Anggota
4. Sistem menampilkan semua data anggota yang terdapat pada database.
5. Use Case selesai.

#### 5. Alternative Flow

- A-1 Aktor memilih untuk melakukan Delete Data Anggota

Program Studi Teknik Informatika	SKPL – SCA	20/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Aktor memasukkan nama atau email atau alamat atau nomor hp anggota yang akan dihapus.

E-1 Nama atau email atau alamat atau nomor hp belum tersimpan dalam database.

2. Sistem menampilkan semua data anggota yang tersimpan pada database yang memiliki nama atau email atau alamat atau nomor hp yang sesuai.
3. Aktor memilih salah satu data anggota yang akan dihapus.
4. Sistem meminta konfirmasi penghapusan.
5. Aktor memasukkan konfirmasi penghapusan.
6. Sistem menghapus data tersebut dari database.
7. Kembali ke Basic Flow langkah ke 7 .

A-2 Aktor memilih untuk melakukan Search Data Anggota

1. Aktor memasukkan nama atau email atau alamat atau nomor hp anggota.
2. Sistem akan melakukan pencarian nama atau email atau alamat atau nomor hp ke dalam database.

E-1 Nama atau email atau alamat atau nomor hp belum tersimpan dalam database.

3. Sistem menampilkan nama atau email atau alamat atau nomor hp yang sesuai.
4. Kembali ke Basic Flow langkah 7.

## 6. Error Flow

E-1 Nama atau email atau alamat atau nomor hp belum tersimpan dalam database

1. Sistem memberikan peringatan bahwa nama atau email atau alamat atau nomor hp anggota yang diinputkan tidak ditemukan pada database.
2. Kembali ke Alternative Flow A-1 langkah ke 1 atau A-2 langkah ke 1.

## 7. PreConditions

Program Studi Teknik Informatika	SKPL – SCA	21/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

#### **8. PostConditions**

1. Data anggota telah terupdate.

#### **4.1.3 Use case Spesification : Pengelolaan Data Produk**

##### **1. Brief Description**

Use Case ini digunakan user untuk mengelola data produk perawatan. Termasuk di dalamnya yaitu menambah data produk, mengubah data produk, menghapus data produk, mencari data produk, menampilkan data produk.

##### **2. Primary Actor**

1. Administrator

##### **3. Supporting Actor**

None

##### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data produk.
2. Sistem memberikan pilihan untuk melakukan Entry Data Produk, Edit Data Produk, Delete Data Produk, Search Data Produk, dan Display Data Produk.
3. Aktor memilih untuk melakukan Entry Data Produk .
  - A-1 Aktor memilih untuk melakukan Edit Data Produk.
  - A-2 Aktor Memilih untuk melakukan Delete Data Produk.
  - A-3 Aktor Memilih untuk melakukan Search Data Produk.
  - A-4 Aktor Memilih untuk melakukan Display Data Produk.

Program Studi Teknik Informatika	SKPL – SCA	22/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Aktor memasukkan data produk baru.
  - E-1 Data Produk telah tersimpan dalam database.
5. Sistem menyimpan data ke dalam database.
6. Use case selesai.

#### 5. Alternative Flow

A-1 Aktor memilih untuk melakukan Edit Data Produk

1. Aktor memilih data produk yang diubah.
2. Sistem menampilkan data produk.
3. Aktor mengubah data produk sesuai dengan kebutuhan.
4. Sistem menyimpan perubahan data ke dalam database.
5. Kembali ke Basic Flow langkah ke 6.

A-2 Aktor memilih untuk melakukan Delete Data Produk

1. Aktor memilih data produk yang dihapus.
2. Sistem menampilkan data produk.
3. Aktor menghapus data produk.
4. Sistem menyimpan perubahan data ke dalam database.
5. Kembali ke Basic Flow langkah ke 6.

A-3 Aktor memilih untuk melakukan Search Data Produk

1. Aktor memasukkan nama produk atau merk.
2. Sistem melakukan pencarian pada database yang sesuai dengan data produk tersebut.

E-2 Data produk tidak terdapat pada database.

3. Sistem menampilkan data produk yang sesuai dengan ketentuan.
4. Kembali ke Basic Flow langkah ke 6.

A-4 Aktor memilih untuk melakukan Display Data Produk

1. Sistem menampilkan semua data produk yang tersimpan dalam database.
2. Kembali ke Basic Flow Langkah ke 6.

## 6. Error Flow

E-1 Data Produk telah tersimpan

1. Sistem memberikan peringatan bahwa data produk telah terdapat pada database.
2. Kembali ke Basic Flow A-3 langkah ke 4.

E-2 Data produk tidak ditemukan

1. Sistem memberikan peringatan bahwa data produk tidak terdapat pada database.
2. Kembali ke Alternative Flow A-1 langkah ke 1, A-2 langkah ke 1, A-3 langkah ke 1.

## 7. PreConditions

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

## 8. PostConditions

1. Data Produk terupdate

### 4.1.4 Use case Spesification : Pengelolaan Kategori

#### 1. Brief Description

Use Case ini digunakan user untuk mengelola data kategori produk. Termasuk di dalamnya yaitu menambah data kategori, mengubah data kategori, menghapus data kategori, dan menampilkan data kategori.

#### 2. Primary Actor

1. Administrator

#### 3. Supporting Actor

None

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data kategori.
2. Sistem memberikan pilihan untuk melakukan Entry Kategori, Edit Kategori, Delete Kategori, dan Display Kategori.

Program Studi Teknik Informatika	SKPL – SCA	24/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Aktor memilih untuk melakukan Entry kategori.
  - A-1 Aktor memilih untuk melakukan Edit Kategori.
  - A-2 Aktor Memilih untuk melakukan Delete Kategori.
  - A-3 Aktor Memilih untuk melakukan Display Kategori.
4. Aktor memasukkan data kategori produk.
  - E-1 Data kategori telah tersimpan dalam database.
5. Sistem menyimpan data ke dalam database.
6. Use case selesai.

#### **5. Alternative Flow**

- A-1 Aktor memilih untuk melakukan Edit Kategori
1. Aktor memilih data kategori yang diubah.
  2. Sistem menampilkan data kategori yang sesuai.
  3. Aktor mengubah data kategori sesuai dengan kebutuhan.
  4. Sistem menyimpan perubahan data ke dalam database.
  5. Kembali ke Basic Flow langkah ke 6.
- A-2 Aktor memilih untuk melakukan Delete Kategori
1. Aktor memilih data kategori yang dihapus.
  2. Sistem menampilkan nama kategori yang sesuai.
  3. Aktor menghapus data kategori.
  4. Sistem menyimpan perubahan data ke dalam database.
  5. Kembali ke Basic Flow langkah ke 6.
- A-3 Aktor memilih untuk melakukan Display Kategori
1. Sistem menampilkan semua data kategori yang tersimpan dalam database.
  2. Kembali ke Basic Flow Langkah ke 6.

#### **6. Error Flow**

- E-1 Data kategori telah tersimpan

Program Studi Teknik Informatika	SKPL – SCA	25/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem memberikan peringatan bahwa data kategori telah terdapat pada database.
2. Kembali ke Basic Flow A-3 langkah ke 4.

**7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

**8. PostConditions**

1. Data kategori terupdate

**4.1.5 Use case Spesification : Pendaftaran Anggota**

**1. Brief Description**

Use Case ini digunakan oleh aktor untuk menjadi anggota.

**2. Primary Actor**

1. User

**3. Supporting Actor**

None

**4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk menjadi anggota.
2. Sistem menampilkan form yang harus diisi aktor.
3. Aktor memasukkan data pribadi.  
E-1 Data user telah tersimpan pada database.
4. Sistem menyimpan data baru tersebut ke dalam database.
5. Use case selesai.

**5. Alternative Flow**

None

**6. Error Flow**

E-1 Data user telah tersimpan dalam database

1. Sistem memberikan peringatan bahwa data user telah ada dalam database.

Program Studi Teknik Informatika	SKPL – SCA	26/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Kembali ke Basic Flow langkah ke 3.

#### **7. PreConditions**

None

#### **8. PostConditions**

1. Data user terupdate dalam database

### **4.1.6 Use case Spesification : Profil Anggota**

#### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk mengelola profil anggota. Termasuk di dalamnya yaitu mengubah profil dan mengubah password.

#### **2. Primary Actor**

1. Anggota

#### **3. Supporting Actor**

none

#### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk mengelola profil.
2. Sistem memberikan pilihan untuk melakukan Ubah Profile dan Ubah Password.
3. Aktor memilih untuk melakukan Ubah Profil.  
A-1 Aktor memilih untuk melakukan Ubah Password.
4. Aktor memasukkan data pribadi baru.
5. Sistem menyimpan data baru tersebut ke dalam database.
6. Use case selesai.

#### **5. Alternative Flow**

- A-1 Aktor memilih untuk melakukan Ubah Password
1. Sistem meminta password yang sedang digunakan.
  2. Aktor memasukkan password.
  3. Sistem melakukan pengecekan username dan password ke dalam database.

Program Studi Teknik Informatika	SKPL – SCA	27/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

E-1 Password salah.

4. Aktor memasukkan password yang baru.
5. Sistem menyimpan data baru tersebut ke dalam database.
6. Berlanjut ke Basic Flow langkah ke-6.

#### **6. Error Flow**

E-1 Password salah

1. Sistem memberikan peringatan bahwa password berbeda dengan yang tersimpan dalam database.
2. Kembali ke Alternative Flow langkah ke-2.

#### **7. PreConditions**

1. Use Case Login sudah dilakukan.
2. Aktor telah memasuki sistem.

#### **8. PostConditions**

1. Data user terupdate dalam database

### **4.1.7 Use case Spesification : Verifikasi Akun**

#### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk melakukan verifikasi akan akunnnya karena lupa password.

#### **2. Primary Actor**

1. Anggota

#### **3. Supporting Actor**

None

#### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor lupa dengan passwordnya dan memilih menu lupa password.
  2. Aktor memasukkan username.
  3. Sistem memeriksa username yang merupakan email aktor
- E-1 Username atau email tidak sesuai

Program Studi Teknik Informatika	SKPL – SCA	28/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Sistem mengirimkan email verifikasi ke email aktor.
5. Aktor melakukan konfirmasi dengan memasukkan verifikasi kode.
6. Sistem mengecek verifikasi kode yang telah dimasukkan aktor.
- E-2 Verifikasi kode salah
7. Sistem menampilkan form pembuatan password baru.
8. Aktor memasukkan password yang baru.
9. Sistem mengganti password lama dengan password baru pada database.
10. Use case selesai.

#### **5. Alternative Flow**

None

#### **6. Error Flow**

E-1 Username atau email tidak sesuai

1. Sistem menampilkan peringatan bahwa username atau email tidak sesuai
2. Kembali ke Basic Flow langkah ke-3

E-2 Verifikasi kode salah

1. Sistem menampilkan peringatan bahwa kode salah
2. Kembali ke Basic Flow langkah ke-5

#### **7. PreConditions**

1. Aktor telah terdaftar sebagai anggota.

#### **8. PostConditions**

1. Sandi aktor telah diganti dengan yang baru setelah melakukan verifikasi akun.

### **4.1.8 Use case Spesification : Simulasi Produk**

#### **1. Brief Description**

Program Studi Teknik Informatika	SKPL – SCA	29/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case ini memungkinkan aktor untuk melakukan simulasi terhadap produk yang sesuai dengan kebutuhan aktor yaitu jenis kulit atau masalah kulit yang dimiliki.

## **2. Primary Actor**

1. User

## **3. Supporting Actor**

none

## **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan simulasi produk.
2. Sistem menampilkan form yang harus diisi aktor.
3. Aktor memasukkan data kriteria.
4. Aktor meminta sistem untuk menghitung dan menampilkan produk yang disarankan.
5. Sistem mengecek inputan user dan melakukan perhitungan.
6. Sistem menampilkan produk yang ditawarkan.
7. Use case selesai.

## **5. Alternative Flow**

None

## **6. Error Flow**

None

## **7. PreConditions**

None

## **8. PostConditions**

1. Aktor mengetahui produk yang sesuai dengan kriteria kulit yang dimiliki.

### **4.1.9 Use case Spesification : Pemesanan**

#### **1. Brief Description**

Use Case ini digunakan user untuk melakukan pemesanan. Termasuk di dalamnya yaitu menambah pesanan,

Program Studi Teknik Informatika	SKPL – SCA	30/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

mengubah pesanan, membatalkan pesanan, dan menampilkan pesanan.

## **2. Primary Actor**

1. Anggota

## **3. Supporting Actor**

None

## **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pemesanan.
2. Sistem memberikan pilihan untuk melakukan Entry Pemesanan, Edit Pemesanan, Pembatalan Pemesanan, dan Display Pemesanan.
3. Aktor memilih untuk melakukan Entry Pemesanan.
  - A-1 Aktor memilih untuk melakukan Edit Pemesanan
  - A-2 Aktor memilih untuk melakukan Pembatalan Pemesanan
  - A-3 Aktor Memilih untuk melakukan Display Pemesanan.
4. Aktor memilih produk yang ingin dipesan.
5. Sistem memasukkan data produk ke dalam database pemesanan.
6. Use case selesai.

## **5. Alternative Flow**

A-1 Aktor memilih untuk melakukan Edit Pemesanan

1. Sistem menampilkan data pemesanan yang akan telah dilakukan.
2. Aktor memasukkan data yang ingin dirubah.
3. Sistem menyimpan data baru ke dalam database.
4. Berlanjut ke Basic Flow langkah ke-6

A-2 Aktor memilih untuk melakukan Pembatalan Pemesanan

1. Sistem menampilkan data pemesanan yang akan telah dilakukan.

Program Studi Teknik Informatika	SKPL – SCA	31/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Aktor meminta sistem untuk membatalkan pemesanan yang ditampilkan sistem.

3. Berlanjut ke Basic Flow langkah ke-6

A-3 Aktor memilih untuk melakukan Display Pemesanan

1. Sistem menampilkan data pemesanan yang ada di database.

2. Berlanjut ke Basic Flow langkah ke-6

#### **6. Error Flow**

None

#### **7. PreConditions**

1. Use Case Login sudah dilakukan

2. Aktor telah memasuki sistem

#### **8. PostConditions**

1. Data Pemesanan telah terupdate

### **4.1.10 Use case Spesification : Pengelolaan Pemesanan**

#### **1. Brief Description**

Use Case ini digunakan user untuk mengelola data pemesanan. Termasuk di dalamnya yaitu menghapus pemesanan, mencari pemesanan, dan menampilkan pemesanan.

#### **2. Primary Actor**

1. Adminstrator

#### **3. Supporting Actor**

None

#### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan pemesanan.

2. Sistem memberikan pilihan untuk melakukan Hapus Pemesanan, Cari Pemesanan, dan Display Pemesanan.

3. Aktor memilih untuk melakukan Hapus Pemesanan.

A-1 Aktor memilih untuk melakukan Cari Pemesanan.

Program Studi Teknik Informatika	SKPL – SCA	32/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

A-2 Aktor Memilih untuk melakukan Display Pemesanan.

4. Sistem menghapus data pemesanan pada database yang tidak valid.
5. Use case selesai.

#### **5. Alternative Flow**

A-1 Aktor memilih untuk melakukan Cari Pemesanan

1. Aktor memasukkan data pemesanan yang ingin dicari.
2. Sistem melakukan pengecekan terhadap data yang diinputkan aktor.

E-1 Data yang diinputkan aktor tidak ditemukan.

3. Sistem menampilkan data pemesanan yang ditemukan berdasarkan inputan aktor.
4. Berlanjut ke Basic Flow langkah ke-6

A-2 Aktor memilih untuk melakukan Display Pemesanan

1. Sistem menampilkan data pemesanan yang ada di database.
2. Berlanjut ke Basic Flow langkah ke-6

#### **6. Error Flow**

E-1 Data yang diinputkan aktor tidak ditemukan

1. Sistem menampilkan peringatan bahwa data yang diinputkan tidak ditemukan
2. Kembali ke Alternatif Flow A-1 langkah ke-1.

#### **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

#### **8. PostConditions**

1. Data Pemesanan telah terupdate

Program Studi Teknik Informatika	SKPL – SCA	33/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

#### 4.1.11 Use case Spesification : Pengelolaan Pembayaran

##### 1. Brief Description

Use Case ini digunakan aktor untuk melakukan pengelolaan pembayaran. Fasilitas yang ada yaitu memasukkan data pembayaran, menghapus data pembayaran, dan menampilkan data pembayaran.

##### 2. Primary Actor

1. Administrator

##### 3. Supporting Actor

None

##### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data pembayaran.
2. Sistem memberikan pilihan untuk melakukan Entry Pembayaran, Edit Pembayaran, Delete Pembayaran, Search Pembayaran, dan Display Pembayaran.
3. Aktor memilih untuk melakukan Entry Pembayaran.
  - A-1 Aktor Memilih untuk melakukan Delete Pembayaran.
  - A-2 Aktor Memilih untuk melakukan Search Pembayaran.
  - A-3 Aktor Memilih untuk melakukan Display Pembayaran.
4. Aktor memasukkan pembayaran baru.
  - E-1 Data pembayaran telah tersimpan dalam database.
5. Sistem menyimpan data ke dalam database.
6. Use case selesai.

##### 5. Alternative Flow

A-1 Aktor memilih untuk melakukan Delete Data Pembayaran

1. Aktor memasukkan nama anggota.

E-2 Nama anggota tidak terdapat pada database.

Program Studi Teknik Informatika	SKPL – SCA	34/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan nama anggota yang sesuai.
  3. Aktor menghapus data pembayaran dengan nama anggota tersebut.
  4. Sistem menyimpan perubahan data ke dalam database.
  5. Kembali ke Basic Flow langkah ke 6.
- A-2 Aktor memilih untuk melakukan Search Data Pembayaran
1. Aktor memasukkan data pembayaran.
  2. Sistem melakukan pencarian pada database yang sesuai dengan data pembayaran tersebut.
- E-2 Data pembayaran tidak terdapat pada database.
3. Sistem menampilkan data pembayaran yang sesuai dengan ketentuan.
  4. Kembali ke Basic Flow langkah ke 6.
- A-3 Aktor memilih untuk melakukan Display Data Pembayaran
1. Sistem menampilkan semua data pembayaran yang tersimpan dalam database.
  2. Kembali ke Basic Flow Langkah ke 6.

## 6. Error Flow

- E-1 Data Pembayaran telah tersimpan
1. Sistem memberikan peringatan bahwa data pembayaran telah terdapat pada database.
  2. Kembali ke Basic Flow langkah ke 4.
- E-2 Data pembayaran tidak ditemukan
1. Sistem memberikan peringatan bahwa data pembayaran tidak terdapat pada database.
  2. Kembali ke Alternative Flow A-1 langkah ke 1, A-2 langkah ke 2.

## 7. PreConditions

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

Program Studi Teknik Informatika	SKPL – SCA	35/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## 8. PostConditions

1. Data Pembayaran terupdate

### 4.1.12 Use case Spesification : Konfirmasi Pembayaran

#### 1. Brief Description

Use Case ini digunakan aktor untuk melakukan konfirmasi pembayaran setelah user melakukan pemesanan.

#### 2. Primary Actor

1. Anggota

#### 3. Supporting Actor

None

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan konfirmasi pembayaran.
2. Sistem menampilkan form untuk mengisi data pembayaran.
3. Aktor memasukkan data pembayaran yang telah dilakukan.
4. Sistem menyimpan data ke dalam database.
5. Use case selesai.

#### 5. Alternative Flow

None

#### 6. Error Flow

None

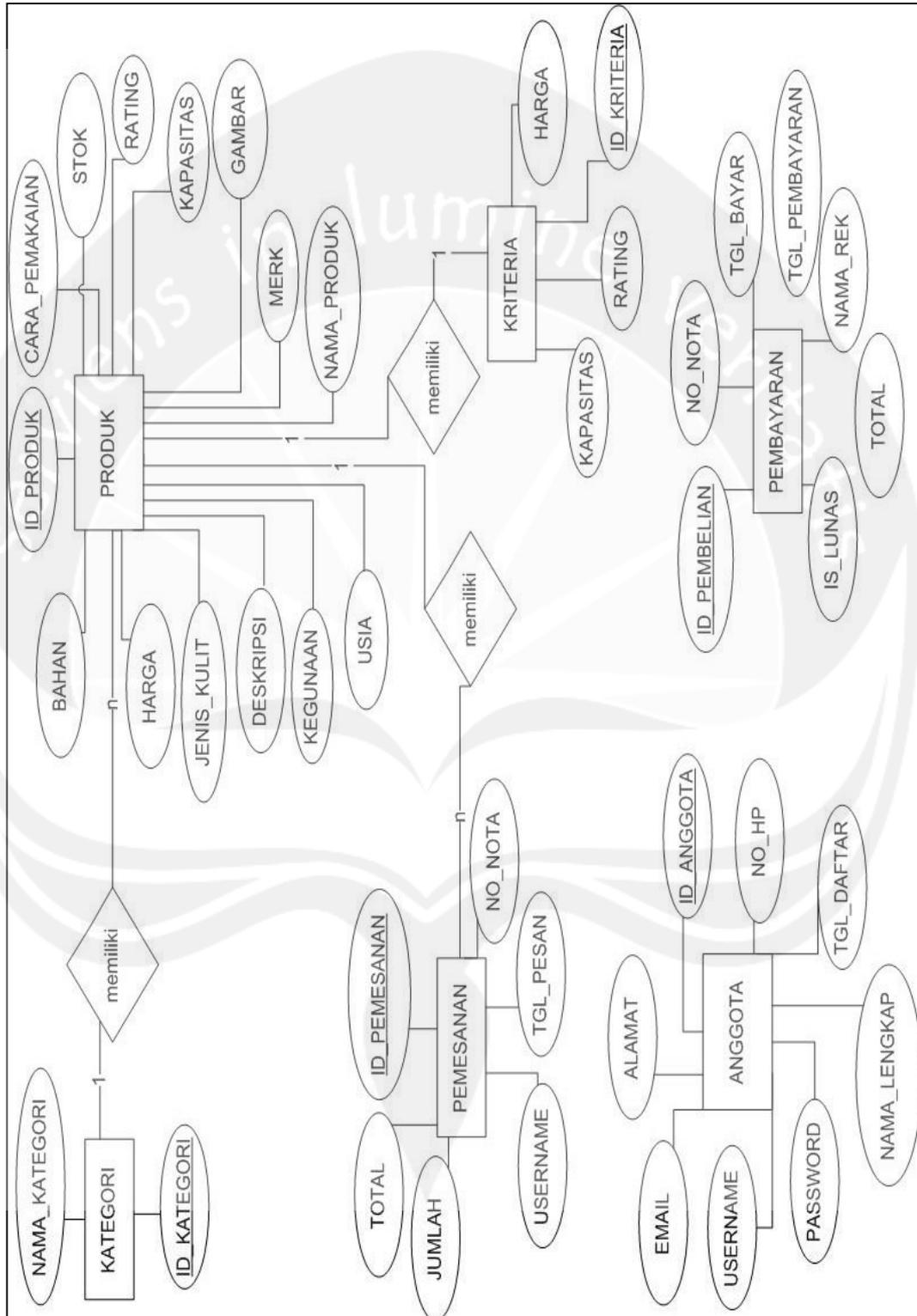
#### 7. PreConditions

1. Aktor telah melakukan pemesanan

#### 8. PostConditions

1. Data Pembayaran terupdate

## 5 Entity Relationship Diagram (ERD)



Gambar 5.1 Entity Relationship Diagram

Program Studi Teknik Informatika	SKPL – SCA	37/ 37
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

# DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SCA

(Skin Care Application)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Helga Laksita Asastani / 5612

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<b>DPPL-SCA</b>		1/70
Fakultas Teknologi Industri				

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperik sa oleh								
Disetuj ui oleh								

### Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

## Daftar Isi

1	Pendahuluan .....	10
1.1	Tujuan.....	10
1.2	Ruang Lingkup.....	10
1.3	Definisi dan Akronim.....	10
1.4	Referensi.....	11
2	Perancangan Sistem .....	12
2.1	Perancangan Arsitektur.....	12
2.2	Perancangan Rinci.....	13
2.2.1	Sequence Diagram.....	13
2.2.1.1	Login.....	13
2.2.1.2	Pengelolaan Anggota .....	13
2.2.1.2.1	Delete Data Anggota .....	13
2.2.1.2.2	Search Data Anggota .....	14
2.2.1.2.3	Tampil Data Anggota .....	14
2.2.1.3	Register Anggota .....	15
2.2.1.4	Profil Anggota.....	15
2.2.1.5	Lupa Sandi.....	<b>Error! Bookmark not defined.</b>
2.2.1.6	Pengelolaan Produk.....	17
2.2.1.6.1	Entry Data Produk .....	17
2.2.1.6.2	Edit Data Produk .....	18
2.2.1.6.3	Hapus Data Produk .....	18
2.2.1.6.4	Tampil Data Produk .....	19
2.2.1.6.5	Cari Data Produk .....	19
2.2.1.7	Skin Care.....	20
2.2.1.8	Pengelolaan Kategori.....	20
2.2.1.8.1	Entry Kategori .....	20
2.2.1.8.2	Edit Kategori .....	21
2.2.1.8.3	Hapus Kategori .....	21

Program Studi Teknik Informatika	DPPL – SCA	4/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2.2.1.8.4	Tampil Kategori .....	22
2.2.1.9	Pemesanan .....	22
2.2.1.9.1	Entry Pemesanan .....	22
2.2.1.9.2	Edit Pemesanan .....	23
2.2.1.9.3	Hapus Pemesanan .....	23
2.2.1.9.4	Tampil Pemesanan .....	24
2.2.1.10	Pengelolaan Pemesanan .....	24
2.2.1.10.1	Hapus Data Pemesanan .....	24
2.2.1.11	Pengelolaan Pembayaran .....	26
2.2.1.11.1	Edit Pembayaran	<b>Error! Bookmark not defined.</b>
2.2.1.11.2	Hapus Pembayaran .....	26
2.2.1.11.3	Tampil Pembayaran .....	26
2.2.1.11.4	Cari Pembayaran .....	27
2.2.1.12	Konfirmasi Pembayaran .....	27
2.2.2	Class Diagram .....	28
2.2.3	Class Diagram Specific Descriptions .....	29
2.2.3.1	Specific Design Class LoginUI .....	29
2.2.3.2	Specific Design Class PengelolaanAnggota .....	29
2.2.3.3	Specific Design Class ProfilAnggota .....	30
2.2.3.4	Specific Design Class LupaSandi .....	30
2.2.3.5	Specific Design Class RegisterAnggota .....	30
2.2.3.6	Specific Design Class SkinCare .....	31
2.2.3.7	Specific Design Class PengelolaanKategori .....	31
2.2.3.8	Specific Design Class PengelolaanProduk .....	32
2.2.3.9	Specific Design Class PemesananUI .....	32
2.2.3.10	Specific Design Class PengelolaanPemesanan .....	33
2.2.3.11	Specific Design Class PengelolaanPembayaran .....	33
2.2.3.12	Specific Design Class KonfirmasiPembayaran .....	34
2.2.3.13	Specific Design Class AnggotaManager .....	34
2.2.3.14	Specific Design Class KategoriManager .....	35
2.2.3.15	Specific Design Class KriteriaManager .....	36

2.2.3.16	Specific Design Class ProdukManager	37
2.2.3.17	Specific Design Class LoginManager	38
2.2.3.18	Specific Design Class PemesananManager	38
2.2.3.19	Specific Design Class PembayaranManager	39
2.2.3.20	Specific Design Class UserDB	40
2.2.3.21	Specific Design Class KategoriDB	41
2.2.3.22	Specific Design Class KriteriaDB	42
2.2.3.23	Specific Design Class ProdukDB	43
2.2.3.24	Specific Design Class PemesananDB	45
2.2.3.25	Specific Design Class PembayaranDB	46
3	Perancangan Data	48
3.1	Dekomposisi Data	48
3.1.1	Deskripsi Entitas Data ANGGOTA	48
3.1.2	Deskripsi Entitas Data PRODUK	48
3.1.3	Deskripsi Entitas Data KATEGORI	49
3.1.4	Deskripsi Entitas Data KRITERIA	49
3.1.5	Deskripsi Entitas Data PEMESANAN	49
3.1.6	Deskripsi Entitas Data PEMBAYARAN	50
3.2	Physical Data Model	51
4	Perancangan Antarmuka	52
4.1	Login	52
4.2	Pengelolaan Anggota	53
4.3	Register Anggota	54
4.4	Profil Anggota	55
4.5	Lupa Sandi	<b>Error! Bookmark not defined.</b>
4.6	Pengelolaan Produk	58
4.6.1	Entry Data Produk	58
4.6.2	Lihat Data Produk 1	59
4.6.3	Lihat Data Produk 2	60
4.6.4	Edit Data Produk	61
4.6.5	Detail Produk	62
4.7	Skin Care	63

4.8	Pengelolaan Kategori.....	65
4.9	Pemesanan.....	66
4.9.1	Edit Pemesanan.....	66
4.9.2	Konfirmasi Pemesanan.....	67
4.10	Pengelolaan Pembayaran.....	69
4.11	Konfirmasi Pembayaran.....	70



## DAFTAR GAMBAR

Gambar 2.1 Arsitektur SCA.....	12
Gambar 2.2 Sequence Diagram : Login.....	13
Gambar 2.3 Sequence Diagram : Delete Data Anggota.....	13
Gambar 2.4 Sequence Diagram : Search Data Anggota.....	14
Gambar 2.4 Sequence Diagram : Display Data Anggota.....	14
Gambar 2.5 Sequence Diagram : Register Anggota.....	15
Gambar 2.6 Sequence Diagram : Profil Anggota.....	16
Gambar 2.7 Sequence Diagram : Lupa Sandi.....	17
Gambar 2.8 Sequence Diagram : Entry Data Produk.....	17
Gambar 2.9 Sequence Diagram : Edit Data Produk.....	18
Gambar 2.10 Sequence Diagram : Hapus Data Produk.....	18
Gambar 2.11 Sequence Diagram : Tampil Data Produk.....	19
Gambar 2.12 Sequence Diagram : Cari Data Produk.....	19
Gambar 2.13 Sequence Diagram : Skin Care.....	20
Gambar 2.14 Sequence Diagram : Entry Kategori.....	20
Gambar 2.15 Sequence Diagram : Edit Kategori.....	21
Gambar 2.16 Sequence Diagram : Hapus Kategori.....	21
Gambar 2.17 Sequence Diagram : Tampil Kategori.....	22
Gambar 2.18 Sequence Diagram : Entry Pemesanan.....	22
Gambar 2.19 Sequence Diagram : Edit Pemesanan.....	23
Gambar 2.20 Sequence Diagram : Hapus Pemesanan.....	23
Gambar 2.21 Sequence Diagram : Tampil Pemesanan.....	24
Gambar 2.22 Sequence Diagram : Hapus Data Pemesanan.....	25
Gambar 2.23 Sequence Diagram : Edit Pembayaran.....	<b>Error!</b>
<b>Bookmark not defined.</b>	
Gambar 2.24 Sequence Diagram : Hapus Pembayaran.....	26
Gambar 2.25 Sequence Diagram : Tampil Pembayaran.....	26
Gambar 2.26 Sequence Diagram : Cari Pembayaran.....	27
Gambar 2.27 Sequence Diagram : Konfirmasi Pembayaran....	27
Gambar 2.28 Class Diagram.....	28

Program Studi Teknik Informatika	DPPL – SCA	8/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Gambar 3.1 Physical Data Model.....	51
Gambar 4.1 Rancangan Antarmuka Login.....	52
Gambar 4.2 Rancangan Antarmuka Pengelolaan Anggota.....	53
Gambar 4.3 Rancangan Antarmuka Register Anggota.....	54
Gambar 4.4 Rancangan Antarmuka Lihat Profil Anggota.....	55
Gambar 4.5 Rancangan Antarmuka Ubah Profile.....	55
Gambar 4.6 Rancangan Antarmuka Ubah Password.....	56
Gambar 4.7 Rancangan Antarmuka Lupa Sandi.....	57
Gambar 4.8 Rancangan Antarmuka Entry Data Produk.....	58
Gambar 4.9 Rancangan Antarmuka Lihat Data Produk.....	59
Gambar 4.10 Rancangan Antarmuka Lihat Data Produk.....	60
Gambar 4.11 Rancangan Antarmuka Edit Data Produk.....	61
Gambar 4.12 Rancangan Antarmuka Detail Produk.....	62
Gambar 4.13 Rancangan Antarmuka Skin Care 1.....	63
Gambar 4.14 Rancangan Antarmuka Skin Care 2	<b>Error! Bookmark not defined.</b>
Gambar 4.15 Rancangan Antarmuka Skin Care 3.....	64
Gambar 4.16 Rancangan Antarmuka Pengelolaan Kategori.....	65
Gambar 4.17 Rancangan Antarmuka Edit Pemesanan.....	66
Gambar 4.18 Rancangan Antarmuka Konfirmasi Pemesanan.....	67
Gambar 4.19 Rancangan Antarmuka Edit Data Pembayaran.....	69
Gambar 4.20 Rancangan Antarmuka Konfirmasi Pembayaran...	70

## 1 Pendahuluan

### 1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

### 1.2 Ruang Lingkup

Perangkat Lunak SCA dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan data anggota.
2. Menangani pengelolaan data produk.
3. Menangani pengelolaan kategori produk.
4. Menangani pengelolaan pendaftaran anggota.
5. Menangani perubahan profil anggota.
6. Menangani perubahan sandi anggota.
7. Menangani pemesanan produk.
8. Menanganai pembayaran pembelian produk.
9. Menangani simulasi pemilihan produk perawatan kulit.

Aplikasi yang berupa web ini dapat berjalan pada semua platform yang dilengkapi dengan web browser.

### 1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
SCA	Perangkat lunak aplikasi pemilihan produk perawatan muka.

Program Studi Teknik Informatika	DPPL – SCA	10/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Administrator	Role yang memberikan hak akses menangani manajemen sistem, termasuk di dalamnya pengelolaan account dari seluruh user yang terdaftar di dalam sistem.
---------------	---

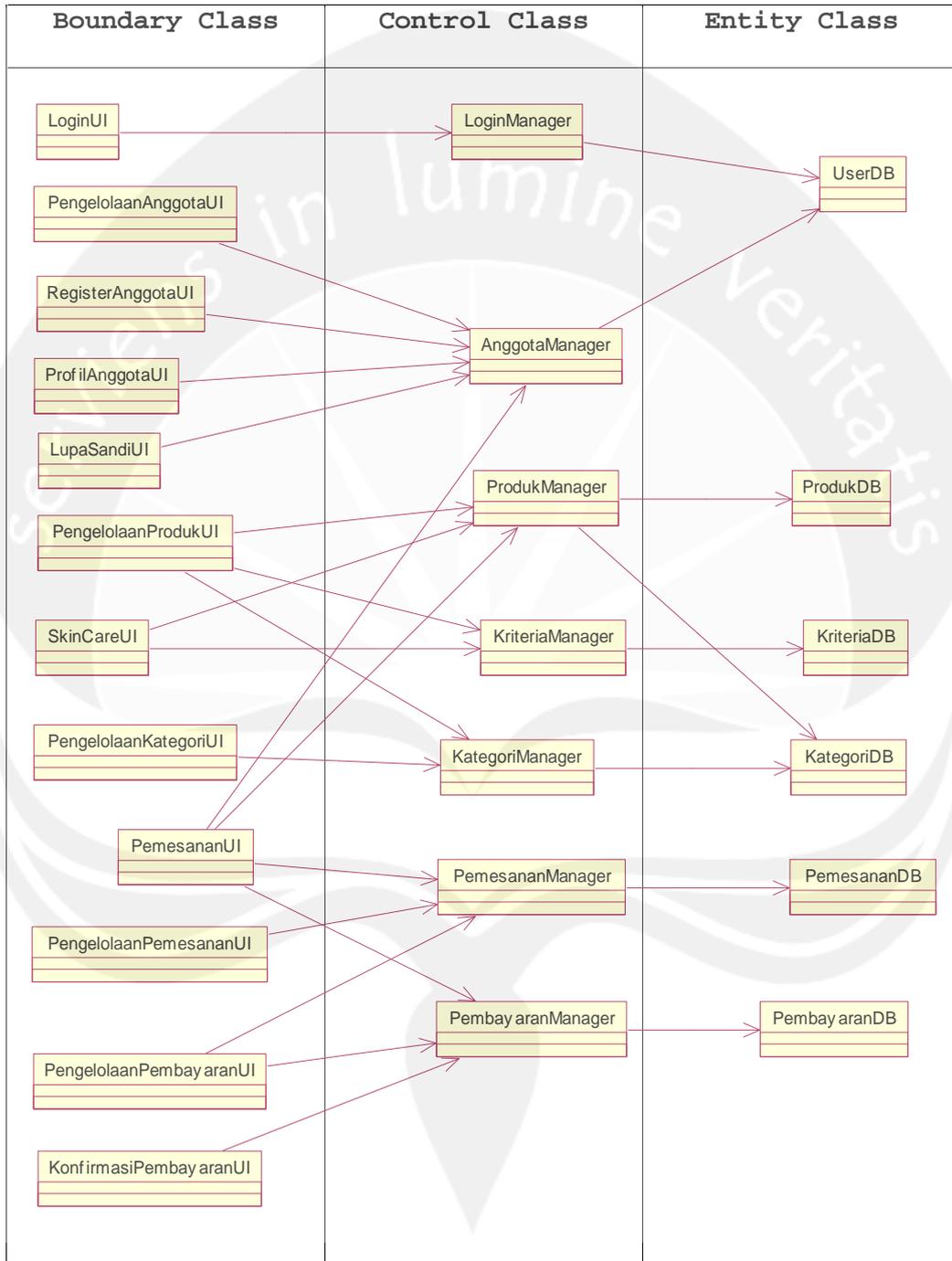
#### 1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Marpaung, Hosanna M.H., 2012, *Pembangunan Aplikasi Online Shopping Berbasis Web*, Skripsi Unpublished, Program Studi Teknik Informatika, Fakultas Teknik Industri, Universitas Atma Jaya, Yogyakarta.

## 2 Perancangan Sistem

### 2.1 Perancangan Arsitektur



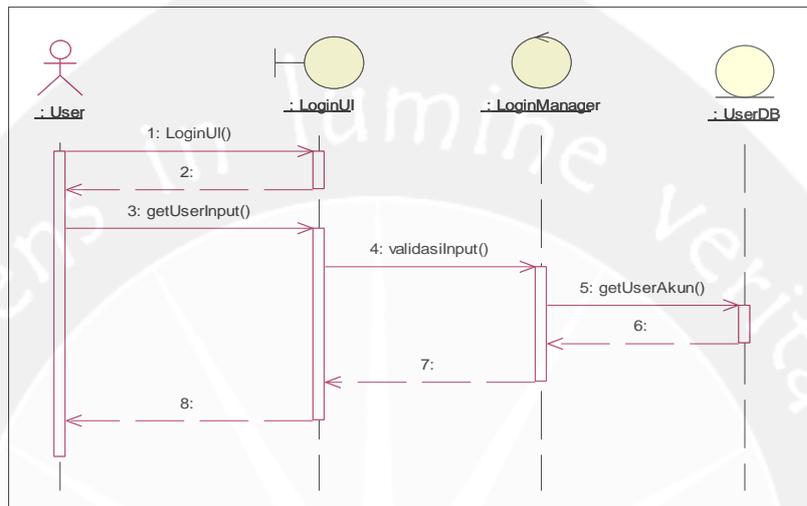
Gambar 2.1 Arsitektur SCA

Program Studi Teknik Informatika	DPPL – SCA	12/ 71
<p>Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

## 2.2 Perancangan Rinci

### 2.2.1 Sequence Diagram

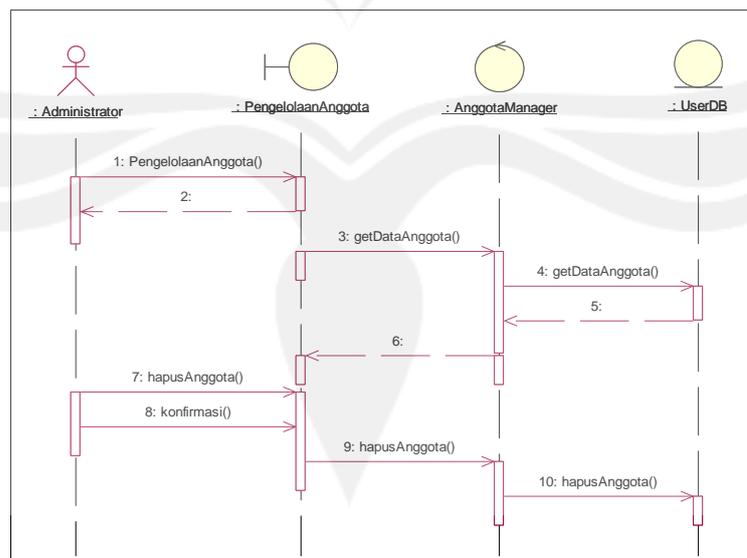
#### 2.2.1.1 Login



Gambar 2.2 Sequence Diagram : Login

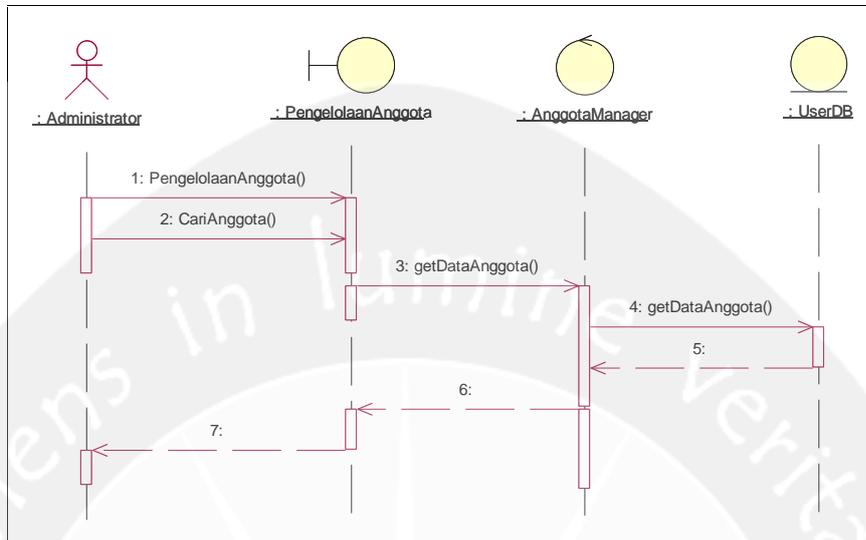
#### 2.2.1.2 Pengelolaan Anggota

##### 2.2.1.2.1 Delete Data Anggota



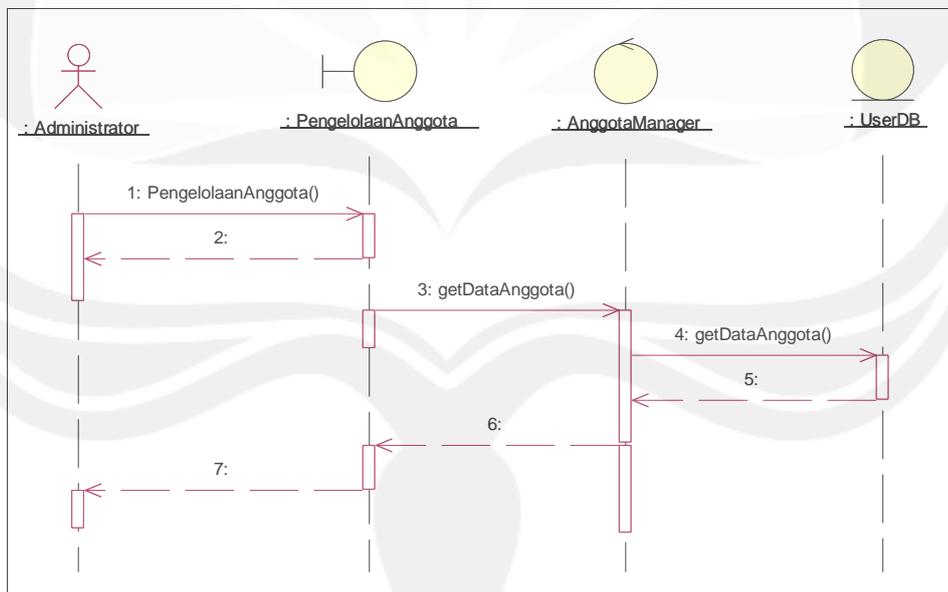
Gambar 2.3 Sequence Diagram : Delete Data Anggota

### 2.2.1.2.2 Search Data Anggota



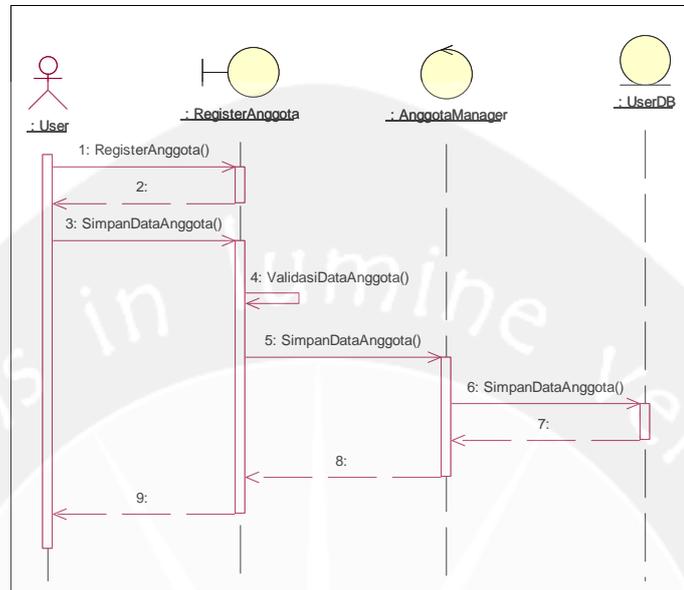
Gambar 2.4 Sequence Diagram : Search Data Anggota

### 2.2.1.2.3 Display Data Anggota



Gambar 2.5 Sequence Diagram : Display Data Anggota

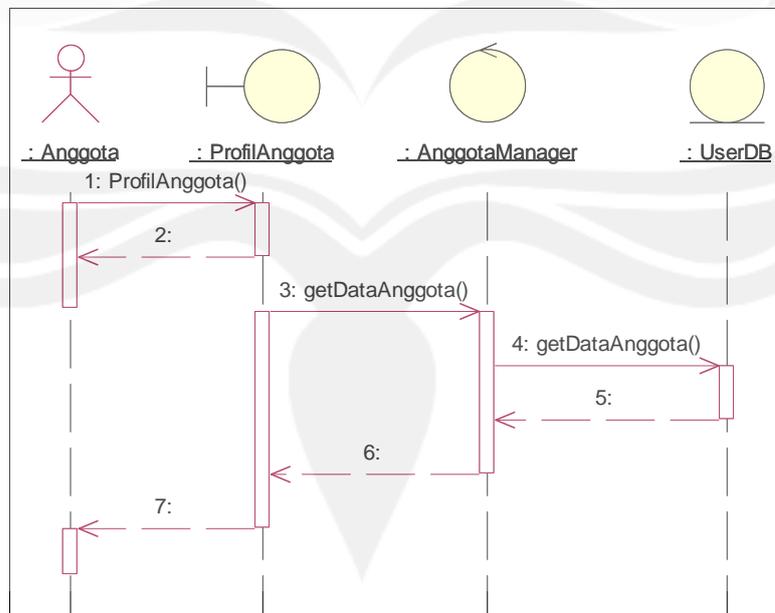
### 2.2.1.3 Register Anggota



Gambar 2.6 Sequence Diagram : Register Anggota

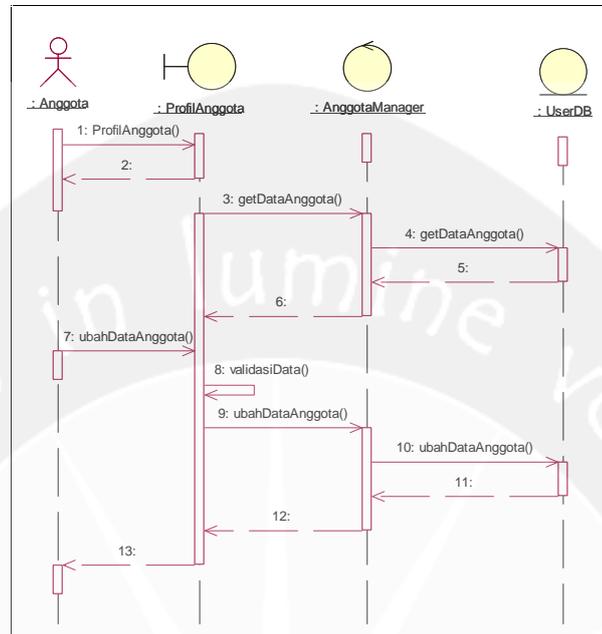
### 2.2.1.4 Profil Anggota

#### 2.2.1.4.1 Lihat Profil



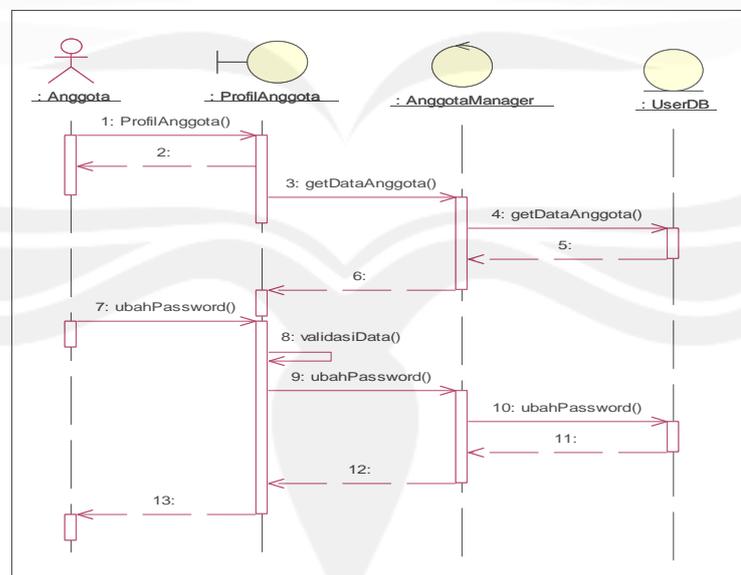
Gambar 2.7 Sequence Diagram : Lihat Profil

### 2.2.1.4.2 Ubah Profil



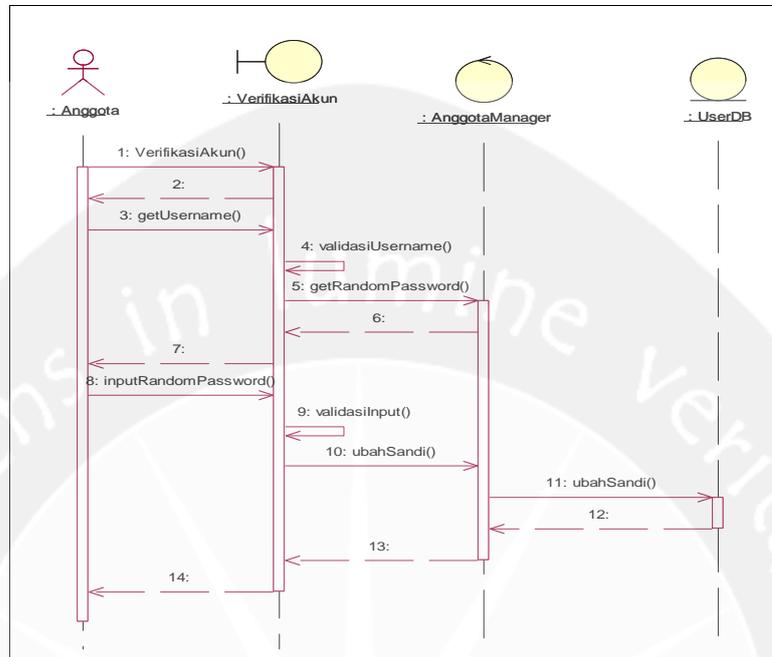
Gambar 2.8 Sequence Diagram : Ubah Profil

### 2.2.1.4.3 Ubah Password



Gambar 2.9 Sequence Diagram : Ubah Password

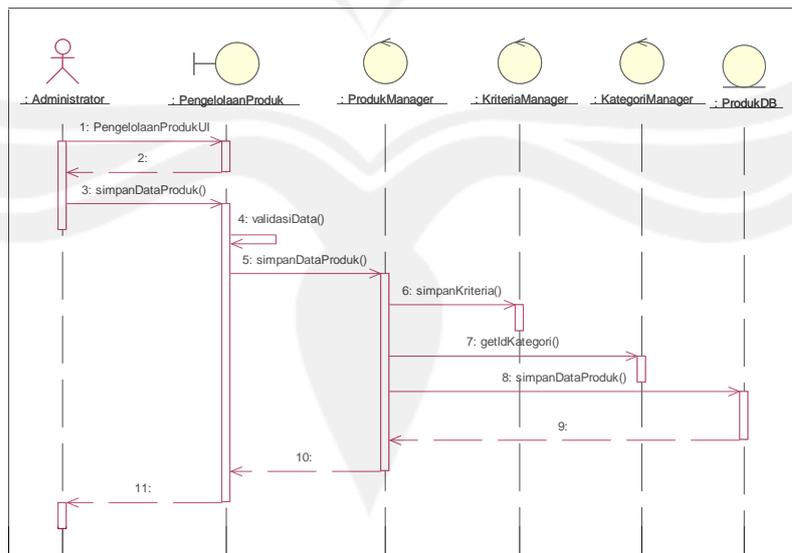
### 2.2.1.5 Verifikasi Akun



Gambar 2.10 Sequence Diagram : Verifikasi Akun

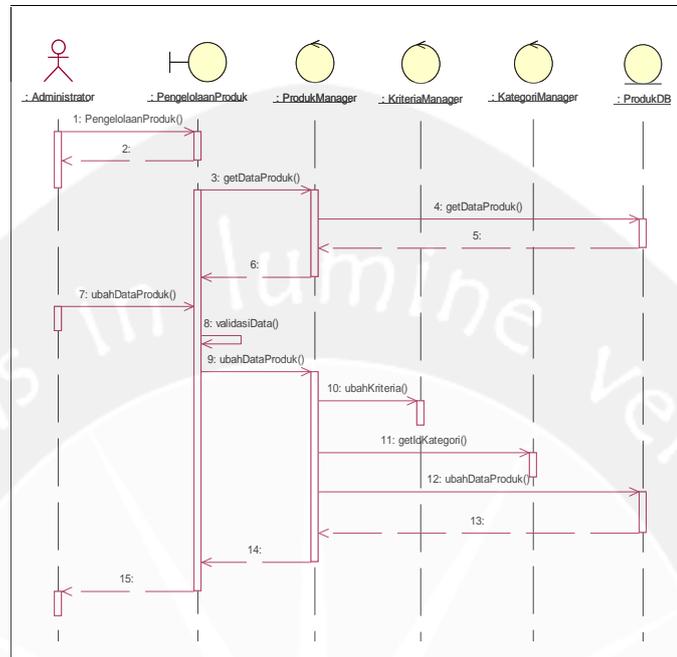
### 2.2.1.6 Pengelolaan Produk

#### 2.2.1.6.1 Entry Data Produk



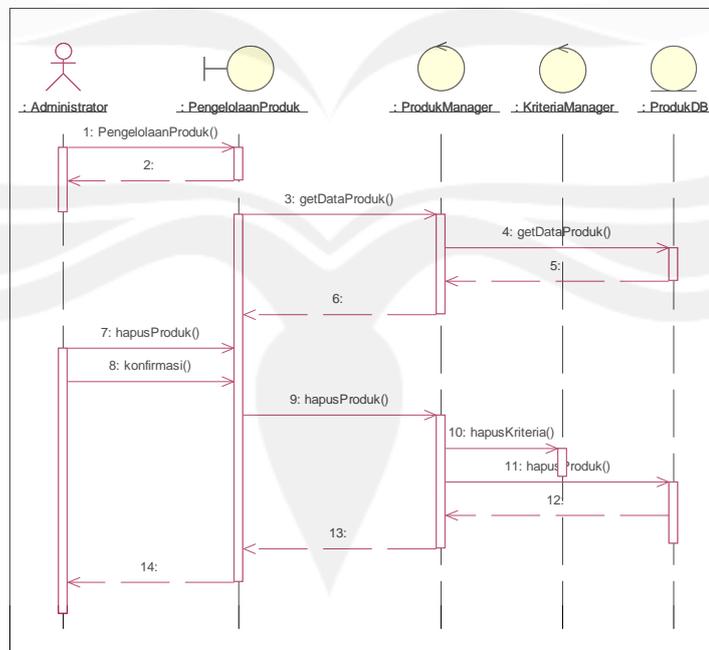
Gambar 2.11 Sequence Diagram : Entry Data Produk

### 2.2.1.6.2 Edit Data Produk



Gambar 2.12 Sequence Diagram : Edit Data Produk

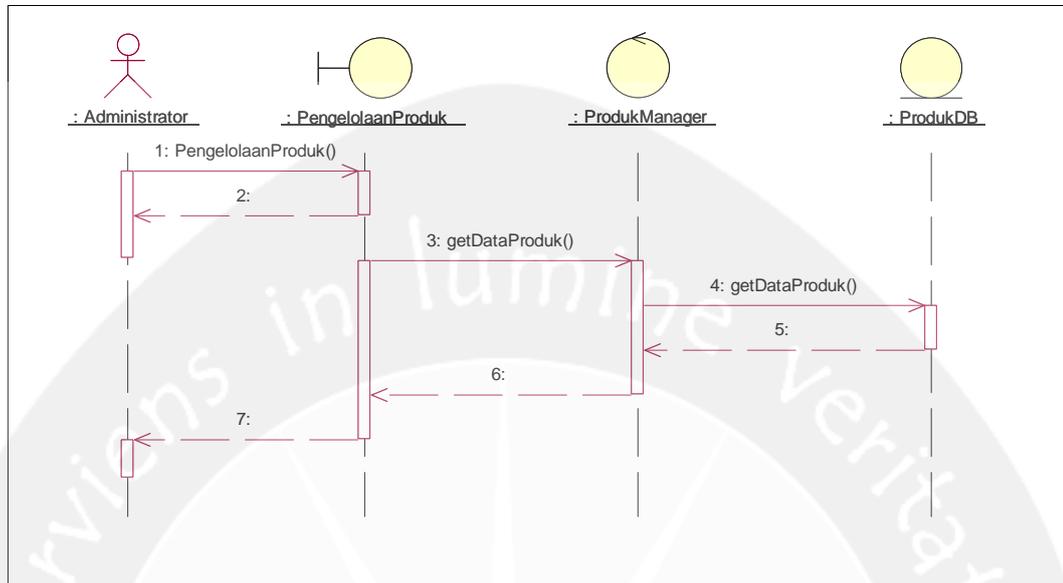
### 2.2.1.6.3 Hapus Data Produk



Gambar 2.13 Sequence Diagram : Hapus Data Produk

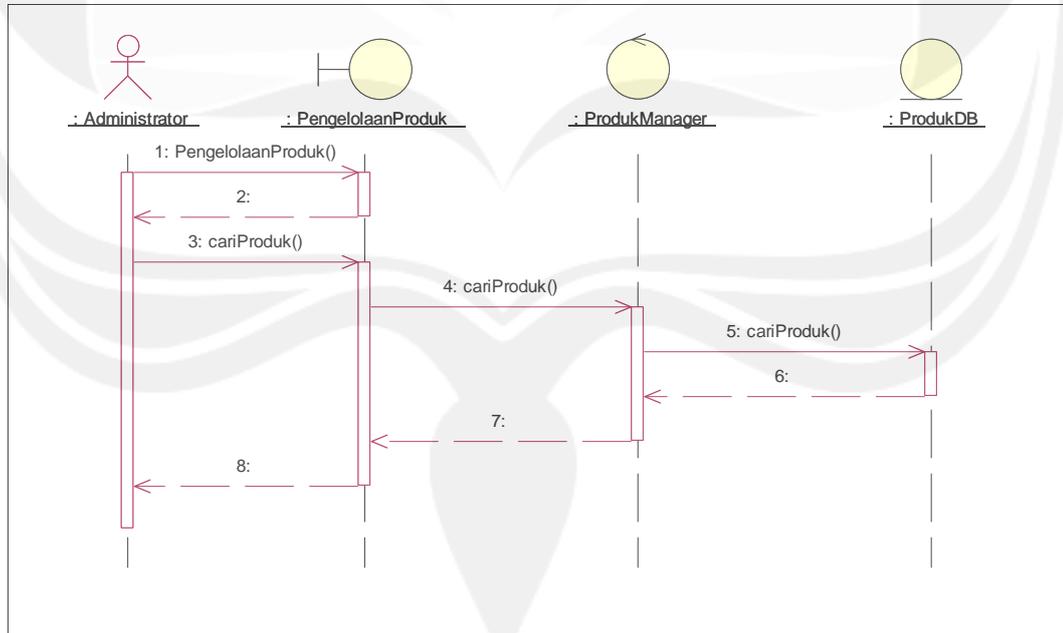
Program Studi Teknik Informatika	DPPL – SCA	18/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

#### 2.2.1.6.4 Tampil Data Produk



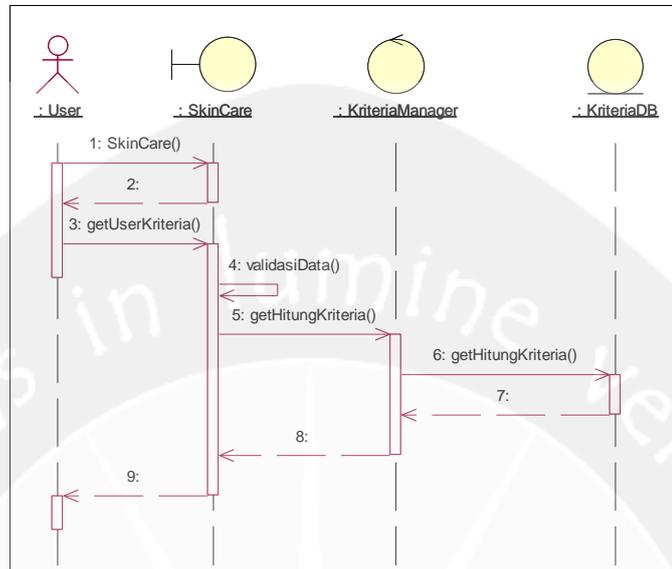
Gambar 2.14 Sequence Diagram : Tampil Data Produk

#### 2.2.1.6.5 Cari Data Produk



Gambar 2.15 Sequence Diagram : Cari Data Produk

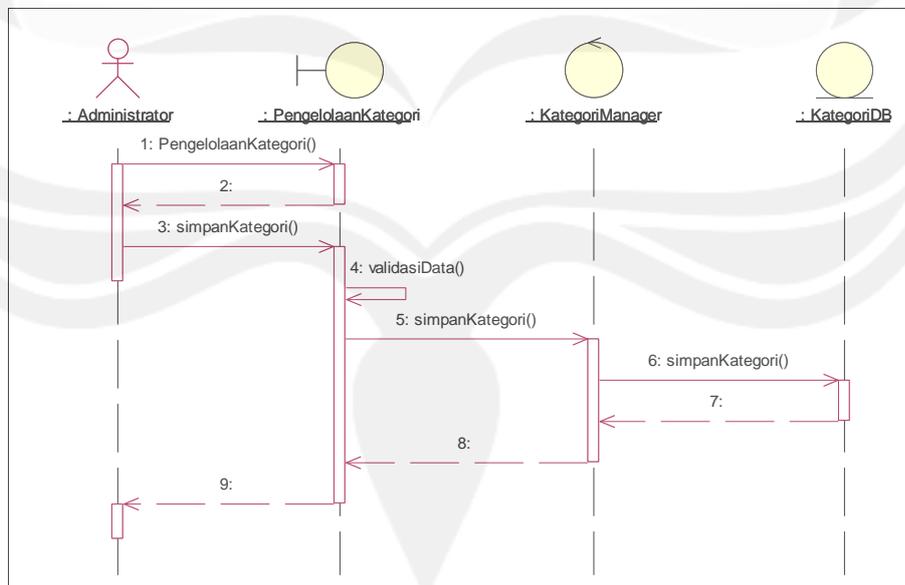
### 2.2.1.7 Simulasi Produk



Gambar 2.16 Sequence Diagram : Simulasi Produk

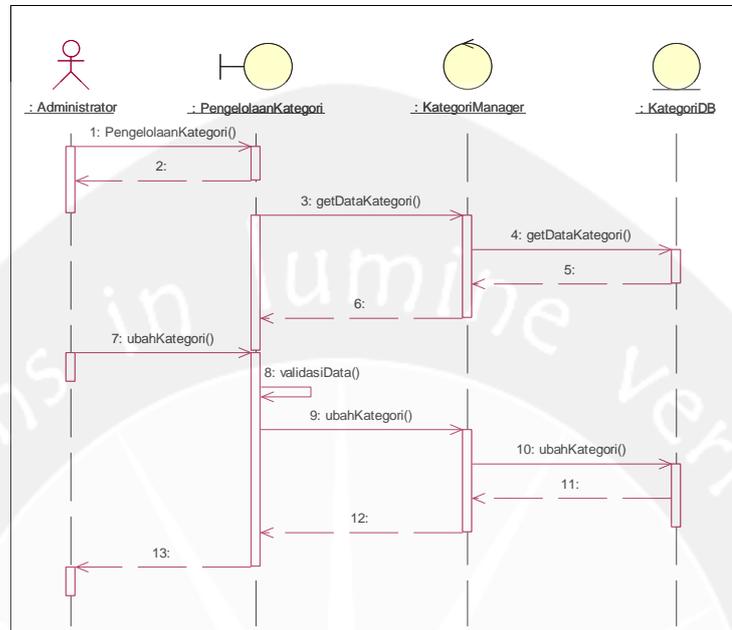
### 2.2.1.8 Pengelolaan Kategori

#### 2.2.1.8.1 Entry Kategori



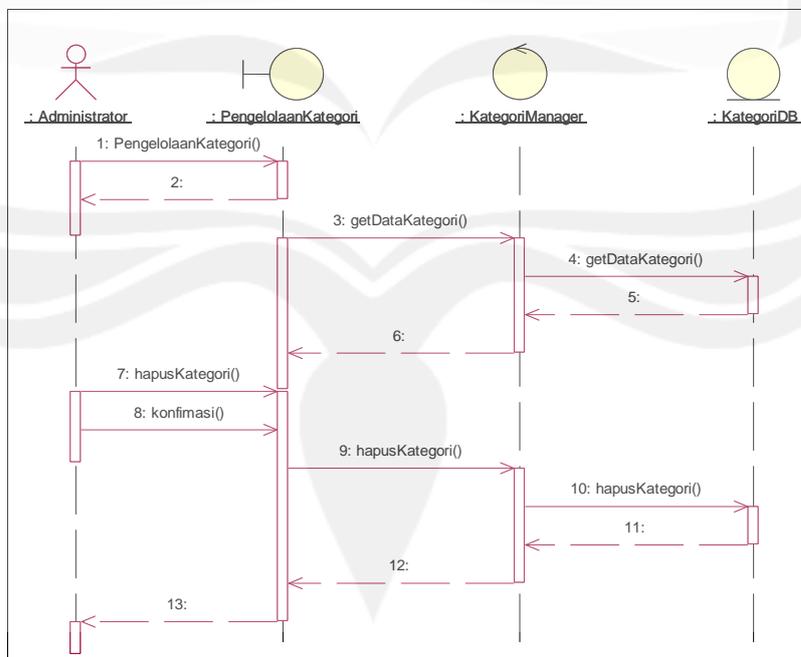
Gambar 2.17 Sequence Diagram : Entry Kategori

### 2.2.1.8.2 Edit Kategori



Gambar 2.18 Sequence Diagram : Edit Kategori

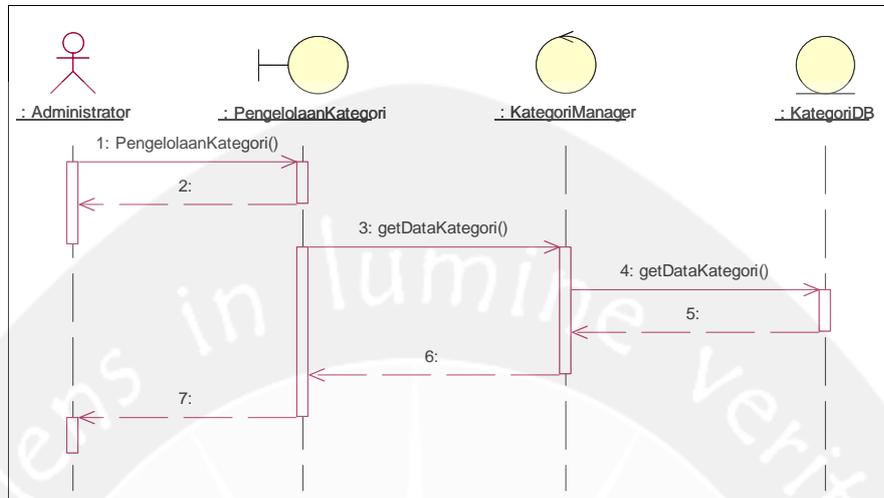
### 2.2.1.8.3 Hapus Kategori



Gambar 2.19 Sequence Diagram : Hapus Kategori

Program Studi Teknik Informatika	DPPL – SCA	21/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

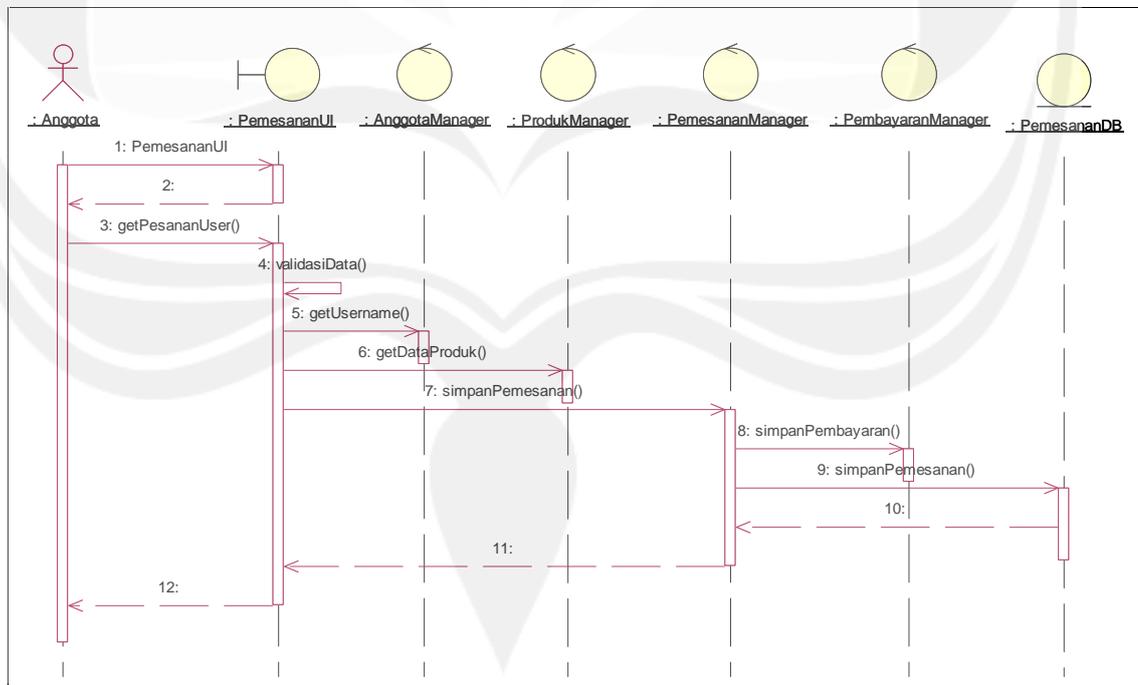
#### 2.2.1.8.4 Tampil Kategori



Gambar 2.20 Sequence Diagram : Tampil Kategori

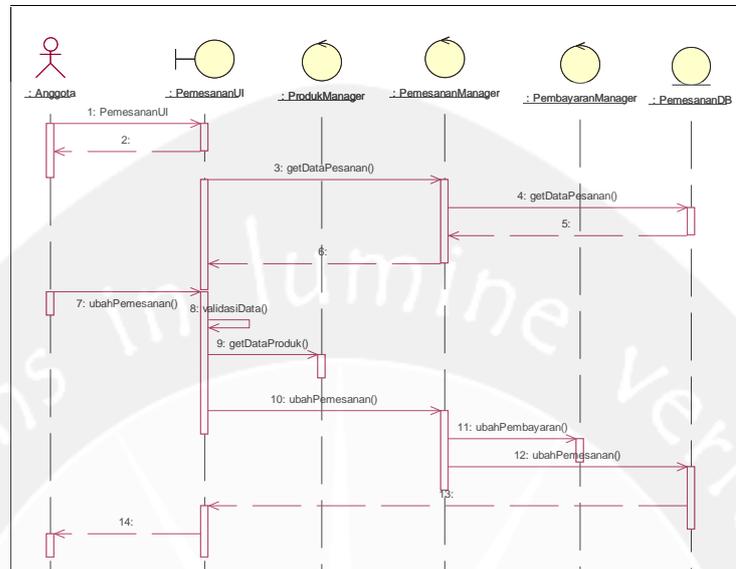
#### 2.2.1.9 Pemesanan

##### 2.2.1.9.1 Entry Pemesanan



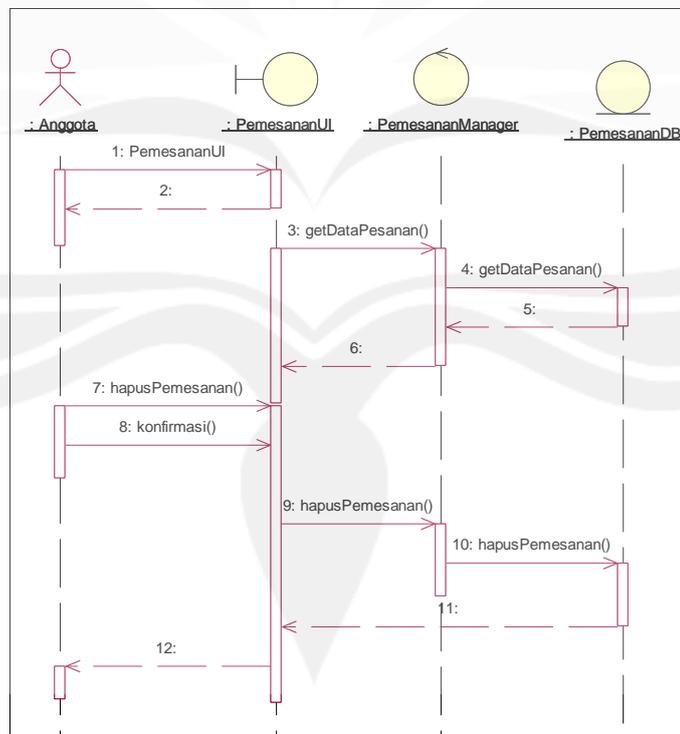
Gambar 2.21 Sequence Diagram : Entry Pemesanan

### 2.2.1.9.2 Edit Pemesanan



Gambar 2.22 Sequence Diagram : Edit Pemesanan

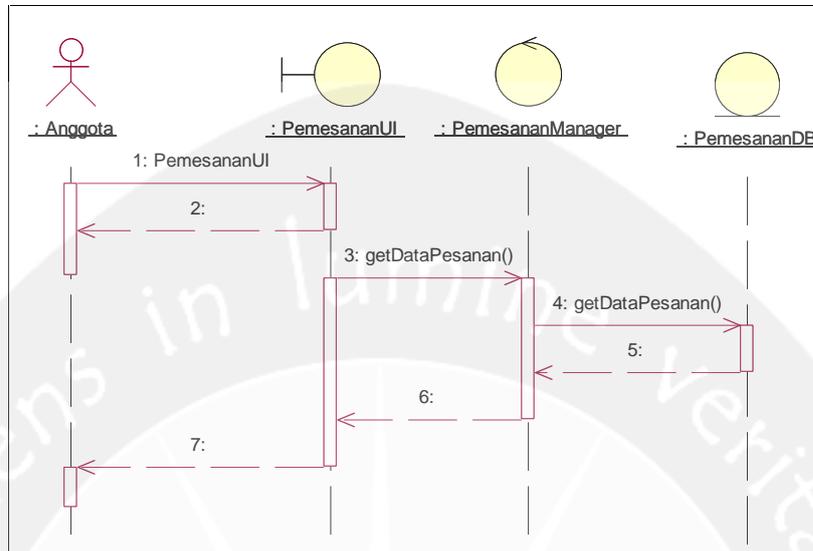
### 2.2.1.9.3 Pembatalan Pemesanan



Gambar 2.23 Sequence Diagram : Pembatalan Pemesanan

Program Studi Teknik Informatika	DPPL – SCA	23/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

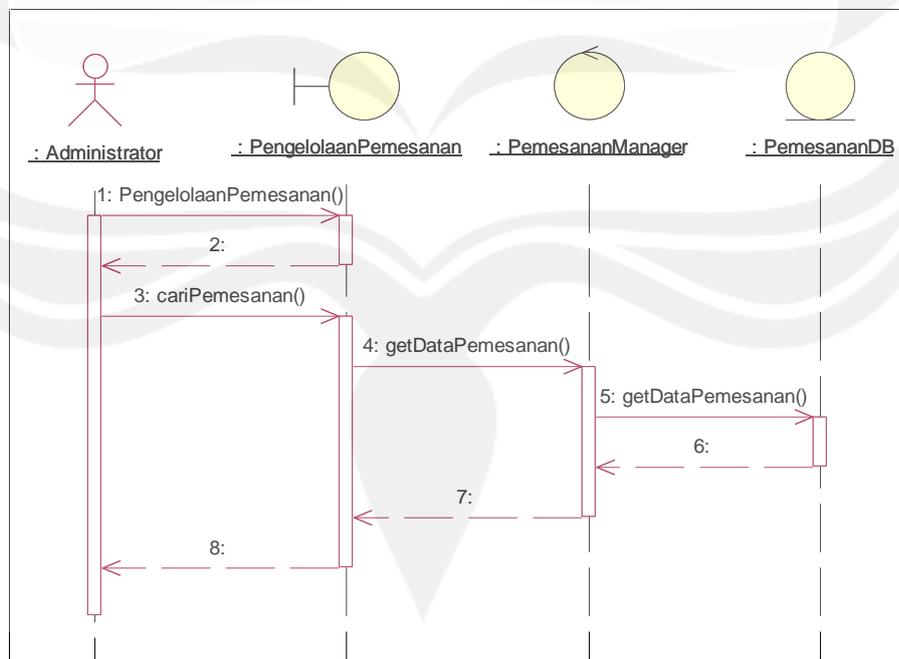
### 2.2.1.9.4 Tampil Pemesanan



Gambar 2.24 Sequence Diagram : Tampil Pemesanan

### 2.2.1.10 Pengelolaan Pemesanan

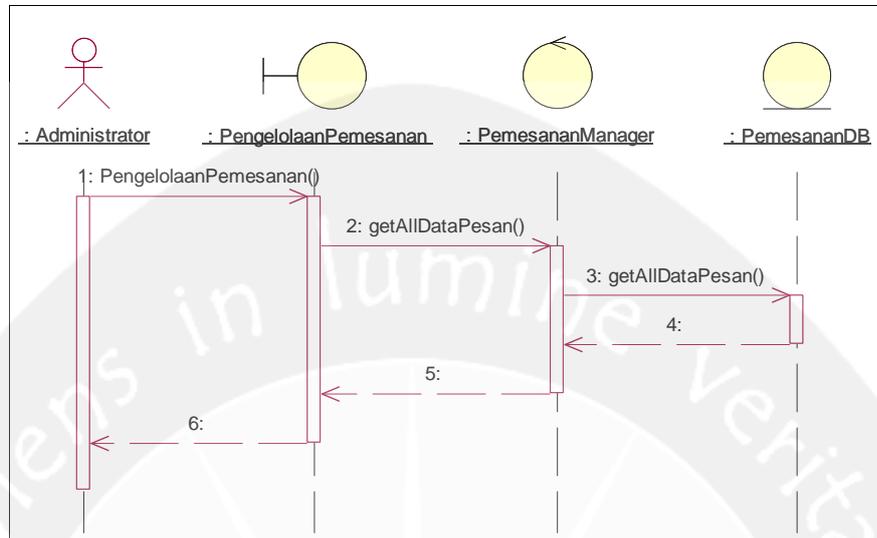
#### 2.2.1.10.1 Cari Data Pesanan



Gambar 2.25 Sequence Diagram : Cari Data Pesanan

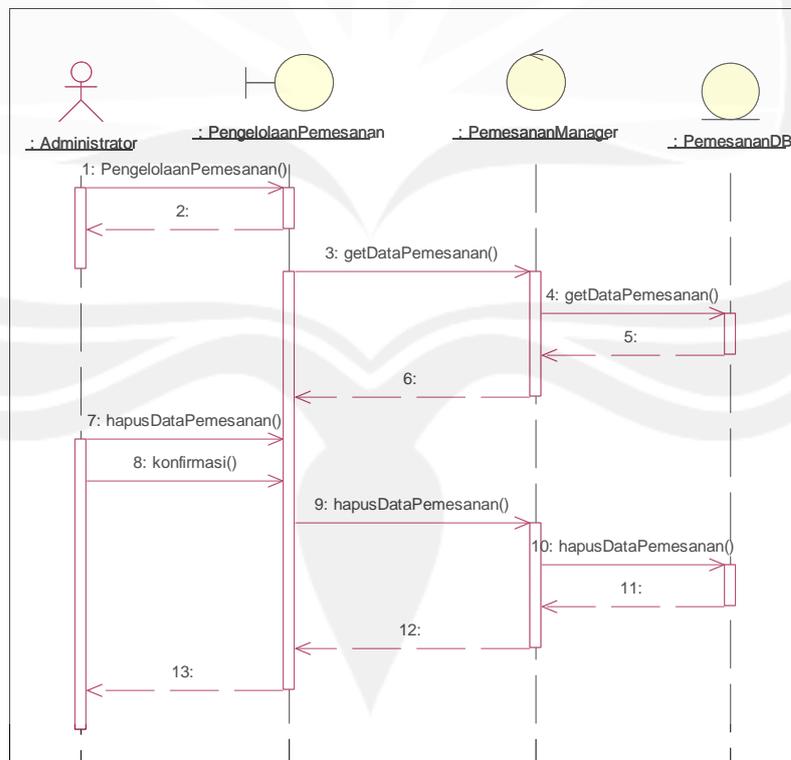
Program Studi Teknik Informatika	DPPL – SCA	24/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 2.2.1.10.2 Tampil Data Pesanan



Gambar 2.26 Sequence Diagram : Tampil Data Pesanan

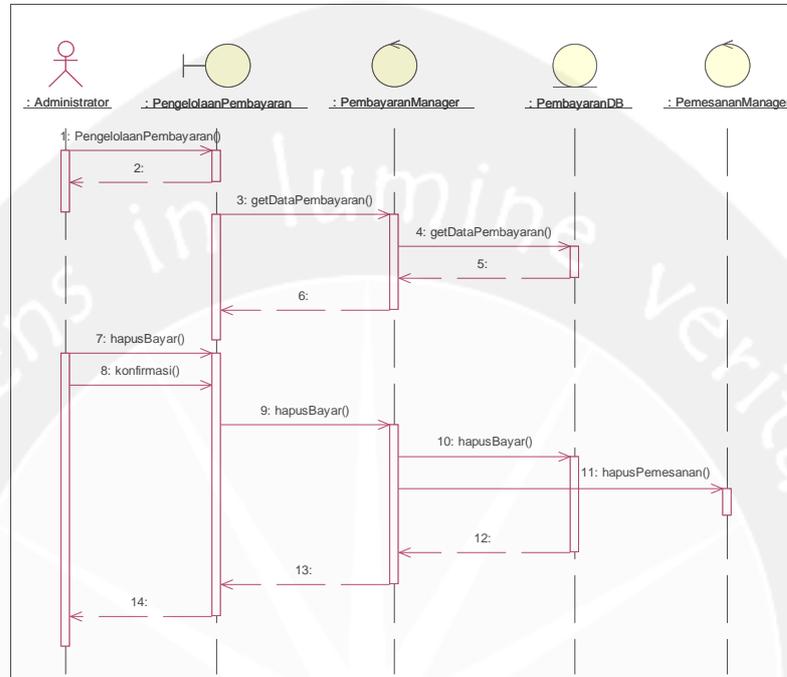
### 2.2.1.10.3 Hapus Data Pesanan



Gambar 2.27 Sequence Diagram : Hapus Data Pesanan

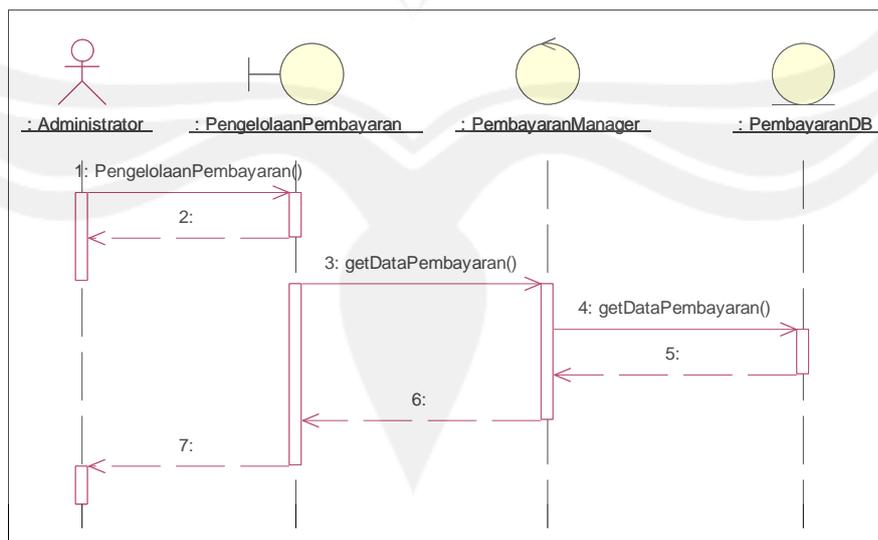
## 2.2.1.11 Pengelolaan Pembayaran

### 2.2.1.11.1 Hapus Pembayaran



Gambar 2.28 Sequence Diagram : Hapus Pembayaran

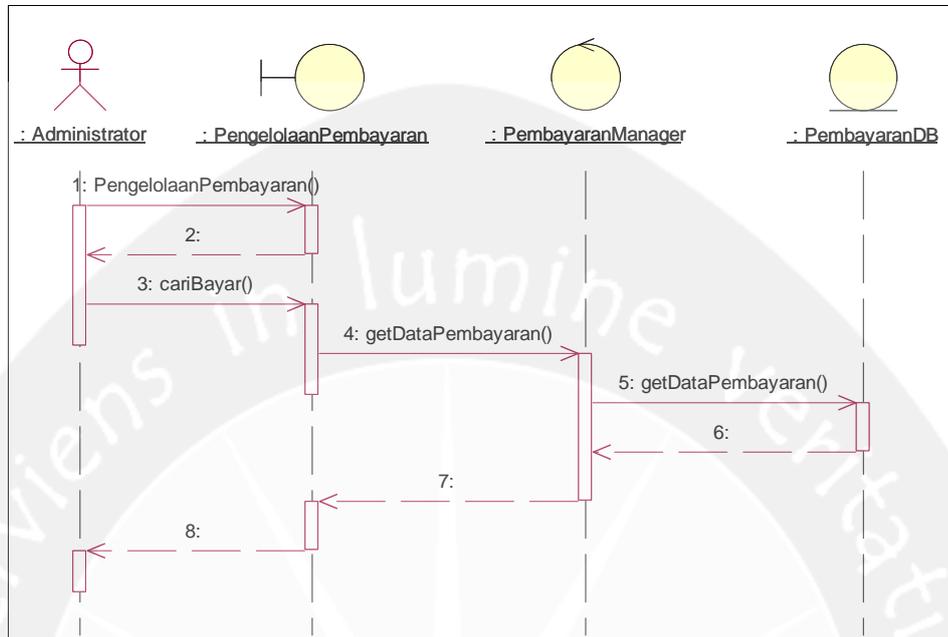
### 2.2.1.11.2 Tampil Pembayaran



Gambar 2.29 Sequence Diagram : Tampil Pembayaran

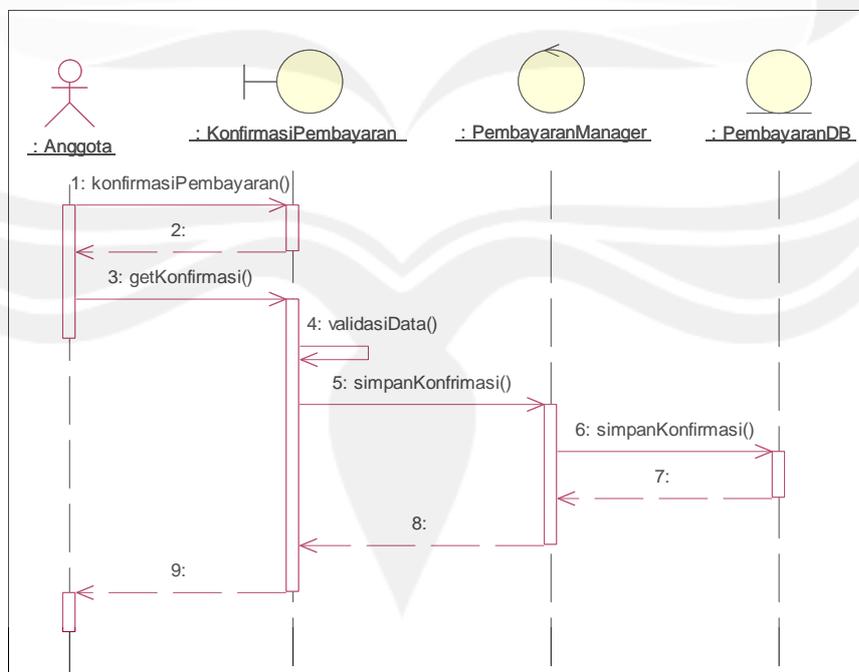
Program Studi Teknik Informatika	DPPL – SCA	26/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 2.2.1.11.3 Cari Pembayaran



Gambar 2.30 Sequence Diagram : Cari Pembayaran

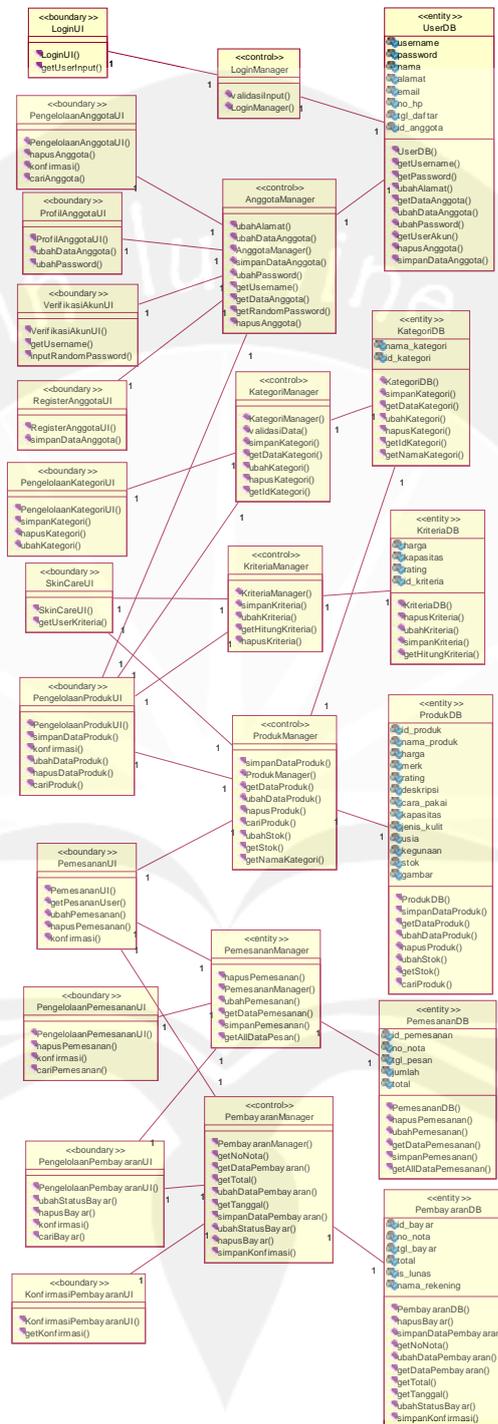
### 2.2.1.12 Konfirmasi Pembayaran



Gambar 2.31 Sequence Diagram : Konfirmasi Pembayaran

Program Studi Teknik Informatika	DPPL – SCA	27/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 2.2.2 Class Diagram



Gambar 2.32 Class Diagram

Program Studi Teknik Informatika	DPPL – SCA	28/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 2.2.3 Class Diagram Specific Descriptions

#### 2.2.3.1 Specific Design Class LoginUI

LoginUI	<<boundary>>
+LoginUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getUserInput() : String Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.	

#### 2.2.3.2 Specific Design Class PengelolaanAnggotaUI

PengelolaanAnggotaUI	<<boundary>>
+PengelolaanAnggotaUI () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+hapusAngota(username) Operasi ini digunakan untuk menghapus data anggota tertentu yang ada pada database.	
+cariAnggota(username) Operasi ini digunakan untuk mencari data anggota tertentu yang ada pada database.	
+konfirmasi(bool) Operasi ini digunakan untuk meminta konfirmasi dari user sebelum melakukan penghapusan.	

Program Studi Teknik Informatika	DPPL – SCA	29/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 2.2.3.3 Specific Design Class ProfilAnggotaUI

<b>ProfilAnggotaUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre> +ProfilAnggotaUI () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +ubahDataAnggota(username) Operasi ini digunakan untuk mengubah data anggota pada database. +ubahPassword(username) Operasi ini digunakan untuk mengubah password anggota pada database. </pre>	

### 2.2.3.4 Specific Design Class VerifikasiAkunUI

<b>VerifikasiAkunUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre> +VerifikasiAkunUI () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +inputRandomPassword() Operasi ini digunakan untuk mengambil dan membandingkan password yang diberikan secara acak oleh sistem dengan input user. +getUsername() Operasi ini digunakan untuk mengambil username yang user. </pre>	

### 2.2.3.5 Specific Design Class RegisterAnggotaUI

<b>RegisterAnggotaUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre> + RegisterAnggotaUI () Default konstruktor, digunakan untuk inisialisasi semua </pre>	

Program Studi Teknik Informatika	DPPL – SCA	30/ 71
<p>Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

attribute dari kelas ini.

+simpanDataAnggota(UserDB)

Operasi ini digunakan untuk menyimpan data anggota baru ke dalam database.

#### 2.2.3.6 Specific Design Class SkinCareUI

<b>SkinCareUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+SkinCareUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+ getUserKriteria() Operasi ini digunakan untuk mengambil kriteria yang dimasukkan oleh user.	

#### 2.2.3.7 Specific Design Class PengelolaanKategoriUI

<b>PengelolaanKategoriUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+ PengelolaanKategoriUI () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+simpanKategori(KategoriDB) Operasi ini digunakan untuk menyimpan data kategori baru ke dalam database.	
+ubahKategori(id_kategori) Operasi ini digunakan untuk mengubah data kategori tertentu pada database.	
+hapusKategori(id_kategori) Operasi ini digunakan untuk menghapus data kategori tertentu pada database.	

### 2.2.3.8 Specific Design Class PengelolaanProdukUI

<b>PengelolaanProdukUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre> + PengelolaanProdukUI () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +simpanDataProduk(ProdukDB) Operasi ini digunakan untuk menyimpan data produk baru ke dalam database. +ubahDataProduk(id_produk) Operasi ini digunakan untuk mengubah data produk tertentu pada database. +cariProduk(id_produk) Operasi ini digunakan untuk mencari data produk tertentu pada database. +hapusDataProduk(id_produk) Operasi ini digunakan untuk menghapus data produk tertentu pada database. +konfirmasi(bool) Operasi ini digunakan untuk meminta konfirmasi dari user sebelum melakukan penghapusan. </pre>	

### 2.2.3.9 Specific Design Class PemesananUI

<b>PemesananUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre> + PemesananUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getPesananUser(PemesananDB) Operasi ini digunakan untuk mengambil dan menyimpan data pemesanan ke dalam database. </pre>	

Program Studi Teknik Informatika	DPPL – SCA	32/ 71
<p>Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

+ubahPemesanan(no\_nota,username)

Operasi ini digunakan untuk mengubah data pemesanan tertentu pada database.

+hapusPemesanan(no\_nota,username)

Operasi ini digunakan untuk menghapus data pemesanan tertentu pada database.

+konfirmasi(bool)

Operasi ini digunakan untuk meminta konfirmasi dari user sebelum melakukan penghapusan.

+cariPemesanan(no\_nota)

Operasi ini digunakan untuk mencari data pemesanan tertentu pada database.

#### 2.2.3.10 Specific Design Class PengelolaanPemesananUI

<b>PengelolaanPemesananUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+ PengelolaanPemesananUI () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+hapusPemesanan(no_nota) Operasi ini digunakan untuk menghapus data pemesanan yang batal pada database.	
+konfirmasi(bool) Operasi ini digunakan untuk meminta konfirmasi dari user sebelum melakukan penghapusan.	

#### 2.2.3.11 Specific Design Class PengelolaanPembayaranUI

<b>PengelolaanPembayaranUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+ PengelolaanPembayaranUI () Default konstruktor, digunakan untuk inisialisasi semua	

attribute dari kelas ini.

+cariBayar(no\_nota)

Operasi ini digunakan untuk mencari data pembayaran tertentu ke dalam database.

+ubahStatusBayar(no\_nota,username)

Operasi ini digunakan untuk mengubah status bayar (is\_lunas) pada database.

+hapusBayar(no\_nota)

Operasi ini digunakan untuk menghapus data pembayaran tertentu pada database.

+konfirmasi(bool)

Operasi ini digunakan untuk meminta konfirmasi dari user sebelum melakukan penghapusan.

#### 2.2.3.12 Specific Design Class KonfirmasiPembayaranUI

<b>KonfirmasiPembayaranUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+ KonfirmasiPembayaranUI () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getKonfirmasi(username,no_nota) Operasi ini digunakan untuk mengubah status bayar (is_lunas) user tertentu pada database.	

#### 2.2.3.13 Specific Design Class AnggotaManager

<b>AnggotaManager</b>	<b>&lt;&lt;control&gt;&gt;</b>
+ AnggotaManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+simpanDataAnggota(AnggotaDB)	

Program Studi Teknik Informatika	DPPL – SCA	34/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Operasi ini digunakan untuk menyimpan data anggota baru ke dalam database.

+ubahDataAnggota(username)

Operasi ini digunakan untuk mengubah data anggota tertentu pada database.

+ubahAlamat(username)

Operasi ini digunakan untuk mengubah alamat seorang user pada database.

+ubahPassword(username)

Operasi ini digunakan untuk mengubah password user pada database.

+getDataAnggota()

Operasi ini digunakan untuk menampilkan data anggota yang ada pada database.

+getRandomPassword()

Operasi ini digunakan untuk membuat password acak.

+hapusAnggota(username)

Operasi ini digunakan untuk menghapus data anggota tertentu yang ada pada database.

+getUsername(username)

Operasi ini digunakan untuk mengambil username tertentu dari database.

#### 2.2.3.14 Specific Design Class KategoriManager

<b>KategoriManager</b>	<b>&lt;&lt;control&gt;&gt;</b>
<p>+ KategoriManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+simpanKategori(KategoriDB) Operasi ini digunakan untuk menyimpan data kategori baru ke dalam database.</p>	

Program Studi Teknik Informatika	DPPL – SCA	35/ 71
----------------------------------	------------	--------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

```

+ubahKategori(id_kategori)
Operasi ini digunakan untuk mengubah data kategori pada
database.
+hapusKategori(id_kategori)
Operasi ini digunakan untuk menghapus kategori tertentu pada
database.
+getIdKategori(nama_kategori): id_kategori
Operasi ini digunakan untuk mengambil id_kategori
berdasarkan nama_kategori.
+getDataKategori(): KategoriDB
Operasi ini digunakan untuk mengambil semua data kategori
dari database.
+validasiData()
Operasi ini digunakan untuk mengecek data.

```

#### 2.2.3.15 Specific Design Class KriteriaManager

<b>KriteriaManager</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<pre> + KriteriaManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +simpanKriteria(KriteriaDB) Operasi ini digunakan untuk menyimpan data kriteria ke dalam database. +ubahKriteria(id_produk) Operasi ini digunakan untuk mengubah data kriteria tertentu pada database. +hapusKriteria(id_produk) Operasi ini digunakan untuk menghapus data kriteria tertentu dari database. +getHitungKriteria(id_produk) Operasi ini digunakan untuk mengambil data kriteria tertentu </pre>	

dari database.

### 2.2.3.16 Specific Design Class ProdukManager

ProdukManager	<<control>>
<pre>+ ProdukManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +simpanDataProduk(ProdukDB) Operasi ini digunakan untuk menyimpan data produk ke dalam database. +ubahDataProduk(id_produk) Operasi ini digunakan untuk mengubah data produk tertentu pada database. +hapusProduk(id_produk) Operasi ini digunakan untuk menghapus data poduk tertentu pada database. +getDataProduk(): ProdukDB Operasi ini digunakan untuk mengambil semua data produk dari database. +cariProduk(id_produk) Operasi ini digunakan untuk mengambil data tertentu dari database. +ubahStok(id_produk) Operasi ini digunakan untuk mengubah stok produk tertent dari database. +getStok(id_produk): stok Operasi ini digunakan untuk mengambil data stok produk tertentu dari database. +getNamaKategori(id_kategori): nama_kategori Operasi ini digunakan untuk mengambil nama kategori tertentu dari database.</pre>	

### 2.2.3.17 Specific Design Class LoginManager

<b>LoginManager</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+LoginManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +validasiInput() Operasi ini digunakan untuk mengecek input user yaitu username dan password dengan database.</pre>	

### 2.2.3.18 Specific Design Class PemesananManager

<b>PemesananManager</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ PemesananManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +simpanPemesanan(PemesananDB) Operasi ini digunakan untuk menyimpan data pemesanan ke dalam database. +ubahPemesanan(no_nota,username) Operasi ini digunakan untuk mengubah data pemesanan tertentu pada database. +hapusPemesanan(no_nota,username) Operasi ini digunakan untuk menghapus data pemesanan tertentu pada database. +getDataPemesanan(username): PemesananDB Operasi ini digunakan untuk mengambil semua data pemesanan tertentu dari database. +getAllDataPesan(): PemesananDB Operasi ini digunakan untuk mengambil semua data pemesanan dari database.</pre>	

Program Studi Teknik Informatika	DPPL – SCA	38/ 71
<p>Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

### 2.2.3.19 Specific Design Class PembayaranManager

<b>PembayaranManager</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre> + PembayaranManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +simpanDataPembayaran(PembayaranDB) Operasi ini digunakan untuk menyimpan data pembayaran ke dalam database. +ubahDataPembayaran(no_nota) Operasi ini digunakan untuk mengubah data pembayaran tertentu pada database. +ubahStatusBayar(no_nota) Operasi ini digunakan untuk mengubah data pembayaran tertentu pada database. +hapusBayar(no_nota) Operasi ini digunakan untuk menghapus data pembayaran tertentu pada database. +getDataPembayaran(username, no_nota): PembayaranDB Operasi ini digunakan untuk mengambil semua data pembayaran tertentu dari database. +getNoNota(username): no_nota Operasi ini digunakan untuk mengambil no_nota dari database. +getTotal(no_nota): total Operasi ini digunakan untuk mengambil data total tertentu dari database. +getTanggal(username, no_nota): tgl_bayar Operasi ini digunakan untuk mengambil data tanggal tertentu dari database. +simpanKonfirmasi(PembayaranDB) Operasi ini digunakan untuk menyimpan data konfirmasi </pre>	

Program Studi Teknik Informatika	DPPL – SCA	39/ 71
<p>Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

pembayaran ke dalam database.

### 2.2.3.20 Specific Design Class UserDB

UserDB	<<entity>>
<p>-id_anggota: int Atribut ini digunakan untuk menyimpan nomor identitas dari masing-masing user.</p> <p>-username: string Atribut ini digunakan untuk menyimpan username dari user.</p> <p>-password: string Atribut ini digunakan untuk menyimpan password dari user.</p> <p>-nama: string Atribut ini digunakan untuk menyimpan nama dari user.</p> <p>-alamat: string Atribut ini digunakan untuk menyimpan alamat dari user.</p> <p>-email: string Atribut ini digunakan untuk menyimpan email dari user.</p> <p>-no_hp: string Atribut ini digunakan untuk menyimpan nomor handphone dari user.</p> <p>-tgl_daftar: string Atribut ini digunakan untuk menyimpan tanggal pendaftaran dari user.</p>	
<p>+UserDB() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+simpanDataAnggota(UserDB) Operasi ini digunakan untuk menyimpan data user ke dalam database.</p> <p>+hapusAnggota(id_user) Operasi ini digunakan untuk menghapus data user di dalam database.</p>	

+ubahDataAnggota(username)

Operasi ini digunakan untuk menghapus data user tertentu di dalam database.

+getDataAnggota(): UserDB

Operasi ini digunakan untuk mengambil semua data user di dalam database.

+getUsername(username): username

Operasi ini digunakan untuk mengambil data user tertentu di dalam database.

+getPassword(username): password

Operasi ini digunakan untuk mengambil password user tertentu di dalam database.

+ubahAlamat(username)

Operasi ini digunakan untuk mengubah data alamat user tertentu di dalam database.

+ubahPassword(username)

Operasi ini digunakan untuk mengubah password user tertentu di dalam database.

+getUserAkun()

Operasi ini digunakan untuk untuk mengambil username dan password user tertentu di dalam database.

### 2.2.3.21 Specific Design Class KategoriDB

KategoriDB	<<entity>>
<p>-id_kategori: int Atribut ini digunakan untuk menyimpan nomor identitas dari kategori.</p> <p>-nama_kategori: string Atribut ini digunakan untuk menyimpan nama kategori.</p>	
<p>+ KategoriDB() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p>	

+simpanKategori(KategoriDB)

Operasi ini digunakan untuk menyimpan data kategori baru ke dalam database.

+ubahKategori(id\_kategori)

Operasi ini digunakan untuk mengubah data kategori pada database.

+hapusKategori(id\_kategori)

Operasi ini digunakan untuk menghapus kategori tertentu pada database.

+getIdKategori(nama\_kategori): id\_kategori

Operasi ini digunakan untuk mengambil id\_kategori berdasarkan nama\_kategori.

+getDataKategori(): KategoriDB

Operasi ini digunakan untuk mengambil semua data kategori dari database.

+getNamaKategori(id\_kategori): nama\_kategori

Operasi ini digunakan untuk mengambil nama kategori tertentu dari database.

#### 2.2.3.22 Specific Design Class KriteriaDB

KriteriaDB	<<entity>>
<p>-id_kriteria: int Atribut ini digunakan untuk menyimpan nomor identitas dari desk.</p> <p>-harga: float Atribut ini digunakan untuk menyimpan kriteria dari harga.</p> <p>-kapasitas: float Atribut ini digunakan untuk menyimpan kriteria dari kapasitas.</p> <p>-rating: float Atribut ini digunakan untuk menyimpan kriteria dari rating.</p>	
+ KriteriaDB()	

Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.

+simpanKriteria(KriteriaDB)  
Operasi ini digunakan untuk menyimpan data kriteria ke dalam database.

+ubahKriteria(id\_produk)  
Operasi ini digunakan untuk mengubah data kriteria tertentu pada database.

+hapusKriteria(id\_produk)  
Operasi ini digunakan untuk menghapus data kriteria tertentu dari database.

+getHitungKriteria(id\_produk)  
Operasi ini digunakan untuk mengambil data kriteria tertentu dari database.

### 2.2.3.23 Specific Design Class ProdukDB

ProdukDB	<<entity>>
<p>-id_produk: int Atribut ini digunakan untuk menyimpan nomor identitas dari produk.</p> <p>-nama_produk: string Atribut ini digunakan untuk menyimpan nama produk.</p> <p>-harga: int Atribut ini digunakan untuk menyimpan harga produk.</p> <p>-merk: string Atribut ini digunakan untuk menyimpan merk produk.</p> <p>-rating: int Atribut ini digunakan untuk menyimpan rating produk.</p> <p>-deskripsi: string Atribut ini digunakan untuk menyimpan deskripsi produk.</p> <p>-cara_pakai: string Atribut ini digunakan untuk menyimpan cara pemakaian produk.</p>	

-kapasitas: int

Atribut ini digunakan untuk menyimpan kapasitas produk.

-jenis\_kulit: string

Atribut ini digunakan untuk menyimpan kecocokan produk untuk jenis kulit tertentu.

-usia: int

Atribut ini digunakan untuk menyimpan kecocokan produk untuk usia tertentu.

-kegunaan: string

Atribut ini digunakan untuk menyimpan kegunaan produk.

-stok: int

Atribut ini digunakan untuk menyimpan stok dari produk.

-gambar: image

Atribut ini digunakan untuk menyimpan gambar produk.

+ ProdukDB()

Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.

+simpanDataProduk(ProdukDB)

Operasi ini digunakan untuk menyimpan data produk ke dalam database.

+ubahDataProduk(id\_produk)

Operasi ini digunakan untuk mengubah data produk tertentu pada database.

+hapusProduk(id\_produk)

Operasi ini digunakan untuk menghapus data produk tertentu pada database.

+getDataProduk(): ProdukDB

Operasi ini digunakan untuk mengambil semua data produk dari database.

+cariProduk(id\_produk)

Operasi ini digunakan untuk mengambil data tertentu dari database.

+ubahStok(id\_produk)

Operasi ini digunakan untuk mengubah stok produk tertentu dari database.

+getStok(id\_produk): stok

Operasi ini digunakan untuk mengambil data stok produk tertentu dari database.

#### 2.2.3.24 Specific Design Class PemesananDB

PemesananDB	<<entity>>
<p>-id_pemesanan: int Atribut ini digunakan untuk menyimpan nomor identitas dari pemesanan.</p> <p>-no_nota: string Atribut ini digunakan untuk menyimpan nomor nota tiap pemesanan.</p> <p>-tgl_pesan: string Atribut ini digunakan untuk menyimpan tanggal pemesanan terjadi.</p> <p>-jumlah: int Atribut ini digunakan untuk menyimpan jumlah produk tiap pemesanan.</p> <p>-total: int Atribut ini digunakan untuk menyimpan total tiap pemesanan.</p>	
<p>+ PemesananDB() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+simpanPemesanan(PemesananDB) Operasi ini digunakan untuk menyimpan data pemesanan ke dalam database.</p> <p>+ubahPemesanan(no_nota,username) Operasi ini digunakan untuk mengubah data pemesanan tertentu pada database.</p>	

```
+hapusPemesanan(no_nota,username)
Operasi ini digunakan untuk menghapus data pemesanan
tertentu pada database.
+getDataPemesanan(username): PemesananDB
Operasi ini digunakan untuk mengambil data pemesanan
tertentu dari database.
+getAllDataPesan(): PemesananDB
Operasi ini digunakan untuk mengambil semua data pemesanan
dari database.
```

### 2.2.3.25 Specific Design Class PembayaranDB

<b>PembayaranDB</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>-id_bayar: int Atribut ini digunakan untuk menyimpan nomor identitas dari pembayaran. -no_nota: string Atribut ini digunakan untuk menyimpan nomor nota tiap pembayaran. -tgl_bayar: string Atribut ini digunakan untuk menyimpan tanggal jatuh pembayaran. -total: string Atribut ini digunakan untuk menyimpan total tiap pembayaran. -is_lunas: string Atribut ini digunakan untuk menyimpan status pembayaran. -nama_rek: string Atribut ini digunakan untuk menyimpan nama rekening yang digunakan dalam pembayaran.</pre>	
<pre>+ PembayaranDB() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +simpanPembayaran(PembayaranDB)</pre>	

Operasi ini digunakan untuk menyimpan data pembayaran ke dalam database.

+ubahPembayaran(no\_nota)

Operasi ini digunakan untuk mengubah data pembayaran tertentu pada database.

+ubahStatusBayar(no\_nota)

Operasi ini digunakan untuk mengubah data pembayaran tertentu pada database.

+hapusPembayaran(no\_nota)

Operasi ini digunakan untuk menghapus data pembayaran tertentu pada database.

+getDataPembayaran(username, no\_nota): PembayaranDB

Operasi ini digunakan untuk mengambil semua data pembayaran tertentu dari database.

+getNoNota(username): no\_nota

Operasi ini digunakan untuk mengambil no\_nota dari database.

+getTotal(no\_nota): total

Operasi ini digunakan untuk mengambil data total tertentu dari database.

+getTanggal(username, no\_nota): tgl\_bayar

Operasi ini digunakan untuk mengambil data tanggal tertentu dari database.

+simpanKonfirmasi(PembayaranDB)

Operasi ini digunakan untuk menyimpan data konfirmasi pembayaran ke dalam database.

Program Studi Teknik Informatika	DPPL – SCA	47/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 3 Perancangan Data

#### 3.1 Dekomposisi Data

##### 3.1.1 Deskripsi Entitas Data ANGGOTA

Nama	Tipe	Panjang	Keterangan
ID_ANGGOTA	Integer	-	Id User, Primary key
NAMA	Varchar	30	Nama dari user
EMAIL	Varchar	50	Email dari user
ALAMAT	Varchar	100	Alamat dari user
NO_HP	Varchar	20	Nomor handphone dari user
TGL_DAFTAR	Date	-	Tanggal pendaftaran dari user
USERNAME	Varchar	50	Username dari user
PASSWORD	Varchar	15	Password dari user

##### 3.1.2 Deskripsi Entitas Data PRODUK

Nama	Tipe	Panjang	Keterangan
ID_PRODUK	Integer	-	Id Produk, Primary key
ID_KATEGORI	Integer	-	Id Kategori, Foreign Key
NAMA_PRODUK	Varchar	100	Nama produk
HARGA	Integer	-	Harga produk
CARA_PEMAKAIAN	Varchar	1000	Cara pemakaian produk
MERK	Varchar	50	Merk produk
DESKRIPSI	Varchar	1000	Deskripsi produk
STOK	Integer	-	Stok produk
RATING	Integer	-	Rating produk
KAPASITAS	Integer	-	Kapasitas produk
JENIS_KULIT	Varchar	30	Kecocokan produk bagi jenis kulit
USIA	Integer	-	Kecocokan produk bagi usia
KEGUNAAN	Varchar	30	Kegunaan produk

Program Studi Teknik Informatika	DPPL – SCA	48/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

GAMBAR	Image	-	Gambar produk
--------	-------	---	---------------

### 3.1.3 Deskripsi Entitas Data KATEGORI

Nama	Tipe	Panjang	Keterangan
ID_KATEGORI	Integer	-	Id Kategori, Primary key
NAMA_KATEGORI	Varchar	30	Nama kategori

### 3.1.4 Deskripsi Entitas Data KRITERIA

Nama	Tipe	Panjang	Keterangan
ID_KRITERIA	Integer	-	Id Kriteria, Primary key
ID_PRODUK	Integer	-	Id Produk, Foreign key
HARGA	Float	-	Kriteria harga produk
KAPASITAS	Float	-	Kriteria kapasitas produk
RATING	Float	-	Kriteria rating produk

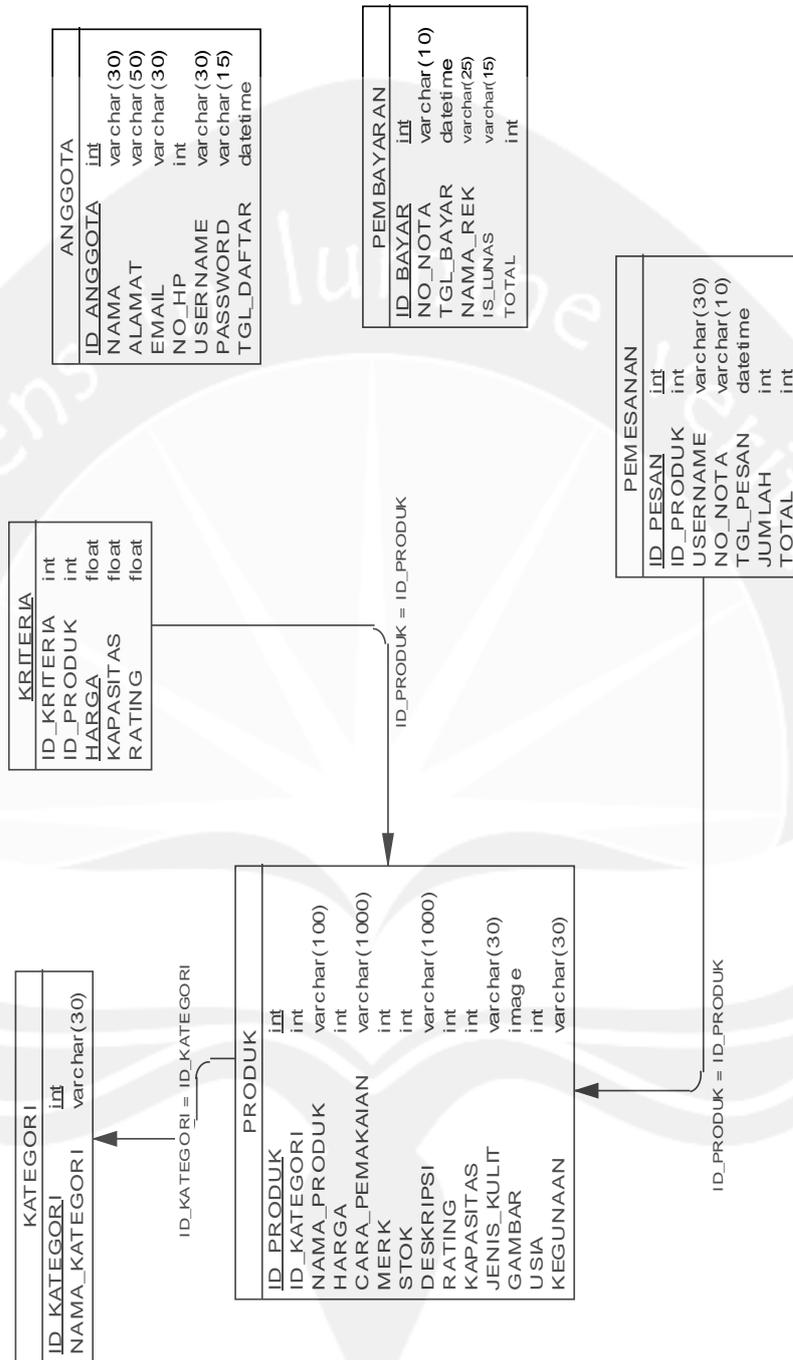
### 3.1.5 Deskripsi Entitas Data PEMESANAN

Nama	Tipe	Panjang	Keterangan
ID_PEMESANAN	Integer	-	Id Pemesanan, Primary key
ID_PRODUK	Integer	-	Id Produk, Foreign Key
USERNAME	Varchar	50	Username, Foreign Key
NO_NOTA	Varchar	10	Nomor nota pemesanan
TGL_PESAN	Date	-	Tanggal pemesanan
JUMLAH	Integer	-	Jumlah pemesanan
TOTAL	Integer	-	Total pemesanan

### 3.1.6 Deskripsi Entitas Data PEMBAYARAN

Nama	Tipe	Panjang	Keterangan
ID_BAYAR	Integer	-	Id Pembayaran, Primary key
NO_NOTA	Varchar	10	Nomor nota pembayaran
TGL_BAYAR	Date	-	Tanggal jatuh pembayaran
NAMA_REK	Varchar	25	Nama rekening pembayaran
TOTAL	Integer	-	Total pembayaran
IS_LUNAS	Varchar	15	Status pembayaran

### 3.2 Physical Data Model



Gambar 3.1 Physical Data Model

## 4 Perancangan Antarmuka

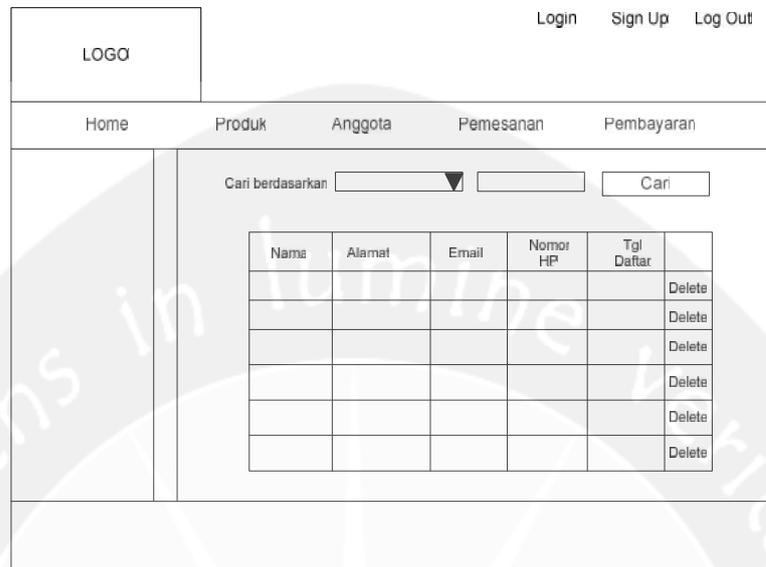
### 4.1 Login

LOGO	Login Sign Up Log Out					
Home	Profile	Produk	Skin Care	Cara Pemesanan	Pemesanan	Konfirmasi Pembayaran
Username <input type="text"/>						
Password <input type="text"/>						
Login					Lupa Sandi	

**Gambar 4.1 Rancangan Antarmuka Login**

Rancangan antarmuka pada gambar 4.1 digunakan untuk melakukan proses login ke dalam sistem. Untuk mendapat akses masuk ke dalam sistem, user harus menginputkan Username dan Password yang benar pada *textbox*. Pada saat tombol Login ditekan, sistem akan mengecek Username dan Password yang diinputkan dengan data Username dan Password yang terdapat di database. Jika data Username dan Password benar atau cocok maka user dapat memesan barang serta mengubah profilnya, sedangkan bila salah atau tidak cocok maka akan muncul peringatan login gagal.

## 4.2 Pengelolaan Anggota



LOGO

Login Sign Up Log Out

Home Produk Anggota Pemesanan Pembayaran

Cari berdasarkan

Name	Alamat	Email	Nomor HP	Tgl Daftar	
					Delete

Gambar 4.2 Rancangan Antarmuka Pengelolaan Anggota

User dapat menghapus data anggota dengan cara mencari data terlebih dahulu pada rancangan antarmuka pada gambar 4.2 kemudian memilih data anggota yang ingin dihapus atau langsung memilih data yang ingin dihapus dengan menekan tombol Delete pada table.

### 4.3 Register Anggota

The image shows a web registration form with the following elements:

- Top right: Login Sign Up Log Out
- Navigation bar: Home Produk User Pembayaran
- Form fields:
  - Nama Lengkap
  - Alamat
  - Email
  - Nomor HP
  - Tanggal Lahir: Tanggal  Bulan  Tahun
  - Password
  - Konfirmasi Password
- Buttons:

Gambar 4.3 Rancangan Antarmuka Register Anggota

Rancangan antarmuka pada gambar 4.3 merupakan form untuk mendaftar menjadi anggota. User mengisi form di atas sesuai dengan data diri dan dengan menekan tombol Daftar untuk menyimpan data. Maka secara otomatis data akan tersimpan dan muncul tanda bahwa user telah berhasil mendaftar. Untuk membatalkan maka user dapat menekan tombol Batalan.

#### 4.4 Profil Anggota

LOGO	Login Sign Up Log Out
Home Profile Produk Skin Care Cara Pemesanan Pemesanan Konfirmasi Pembayaran	
	Nama Nomor HP Alamat Ubah Profile Ubah Password

**Gambar 4.4 Rancangan Antarmuka Lihat Profil Anggota**

Rancangan antarmuka pada gambar 4.4 merupakan form untuk menampilkan profil anggota. User harus login terlebih dahulu supaya dapat mengakses menu profile. Untuk mengubah profile user dapat menekan tombol ubah profile, sedangkan untuk mengubah password user dapat menekan tombol ubah password.

LOGO	Login Sign Up Log Out
Home Profile Produk Skin Care Cara Pemesanan Pemesanan Konfirmasi Pembayaran	
	Nama Nomor HP Alamat Simpan Batal

**Gambar 4.5 Rancangan Antarmuka Ubah Profile**

Antarmuka Ubah Profile digunakan untuk mengubah data user yaitu nama, nomor handphone, dan alamat. Apabila user telah selesai mengubah data maka dapat menekan tombol simpan

yang akan menyimpan perubahan yang dilakukan, sedangkan untuk membatalkan pengubahan data user dapat menekan tombol batal.

LOGG	Login	Sign Up	Log Out
Home	Profile	Produk	Skin Care
			Cara Pemesanan
			Pemesanan
			Konfirmasi Pembayaran
Nama Nomor HP Alamat Password lama <input type="password"/> Password baru <input type="password"/> Konfirmasi password <input type="password"/> <input type="button" value="Simpan"/> <input type="button" value="Batal"/>			

**Gambar 4.6 Rancangan Antarmuka Ubah Password**

Antarmuka Ubah Password digunakan untuk password dengan memasukkan password lama sebagai validasi user. Kemudian user memasukkan password baru dan konfirmasi password yang baru. Untuk menyimpan perubahan password maka user dapat menekan tombol simpan. Apabila password valid maka perubahan akan tersimpan, bila tidak maka sistem akan menampilkan peringatan. Bila user ingin membatalkan perubahan maka dapat menekan tombol batal.

#### 4.5 Verifikasi Akun

LOGO	Login	Sign Up	Log Out			
Home	Profile	Produk	Skin Care	Cara Pemesanan	Pemesanan	Konfirmasi Pembayaran
Kami akan mengirimkan verifikasi password melalui email. Silakan cek email.						
Verifikasi password <input type="text"/>						
<input type="button" value="Submit"/>						
Password baru <input type="text"/>						
Konfirmasi password <input type="text"/>						
<input type="button" value="Simpan"/>						

**Gambar 4.7 Rancangan Antarmuka Verifikasi Akun**

Rancangan antarmuka pada gambar 4.7 di atas akan mengirimkan email secara otomatis ke email user saat user menekan tombol lupa sandi di antarmuka login. Apabila user telah mendapatkan verifikasi kode di dalam email maka user dapat memasukkan verifikasi kode tersebut dan menekan tombol submit. Kemudian sistem akan mengecek kebenaran kode tersebut. Bila benar maka user dapat membuat password baru, tetapi bila salah user harus memasukkan verifikasi kode yang benar.

## 4.6 Pengelolaan Produk

### 4.6.1 Entry Data Produk

The screenshot shows a web application interface for product entry. At the top right, there are links for 'Login', 'Sign Up', and 'Log Out'. Below these is a navigation bar with tabs for 'Home', 'Produk', 'Anggota', 'Pemesanan', and 'Pembayaran'. The main content area is divided into two columns. The left column contains 'Input Produk' and 'Lihat Produk'. The right column contains the product entry form with the following fields and options:

- Nama:
- Gambar:
- Harga:
- Merk:
- Stok:
- Kategori:
- Rating:
- Deskripsi:
- Cara pakai:
- Kapasitas:
- Jenis kulit:  Kering  Normal  Berminyak  
 Sensitif  Kombinasi  Semua jenis kulit
- Usia:
- Masalah:  Jerawat  Komedo  Whitening

At the bottom of the form are two buttons: 'Simpan' and 'Batalan'.

Gambar 4.8 Rancangan Antarmuka Entry Data Produk

Rancangan antarmuka pada gambar 4.8 merupakan form untuk menyimpan data produk. User dapat menyimpan data produk yaitu nama produk, gambar produk, harga, merk, stok produk, kategori, rating, deskripsi, cara pemakaian, kapasitas, jenis kulit, usia dan masalah yang cocok bagi produk dengan menekan tombol simpan untuk menyimpan data. Bila data telah berhasil disimpan maka sistem akan menampilkan pemberitahuan bahwa data telah tersimpan. Untuk membatalkan entry data produk maka dengan menekan tombol batalan.

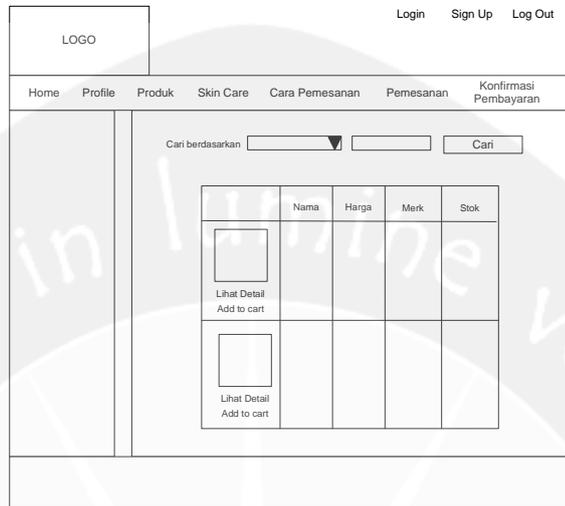
#### 4.6.2 Lihat Data Produk 1

LOGO		Login Sign Up Log Out								
Home		Produk	Anggota	Pemesanan	Pembayaran					
Input Produk	Cari berdasarkan <input type="text"/> <input type="text"/> <input type="button" value="Cari"/>									
Lihat Produk		Nama	Harga	Merk	Stok	Kategori	Cara Pemakaian	Deskripsi	Usia	Kegunaan
	Edit Delete									

**Gambar 4.9 Rancangan Antarmuka Lihat Data Produk**

Antarmuka gambar 4.9 digunakan untuk menampilkan data produk bagi administrator. Untuk mencari data tertentu maka dengan memilih kategori pencarian dan kata kuncinya. Apabila ingin mengubah data maka user dapat menekan kata edit pada table di atas sehingga akan muncul halaman baru yaitu halaman edit data produk.

### 4.6.3 Lihat Data Produk 2



**Gambar 4.10 Rancangan Antarmuka Lihat Data Produk**

Antarmuka gambar 4.10 digunakan untuk menampilkan data produk bagi user. Untuk mencari data tertentu maka dengan memilih kategori pencarian dan kata kuncinya. Apabila ingin melakukan pemesanan user dapat menekan kata add to cart dan secara otomatis sistem akan menambahkan pesanan user ke menu pemesanan. Untuk melihat secara detail produk yang ada maka user dapat menekan kata lihat detail dan user akan mengakses halaman detail produk.

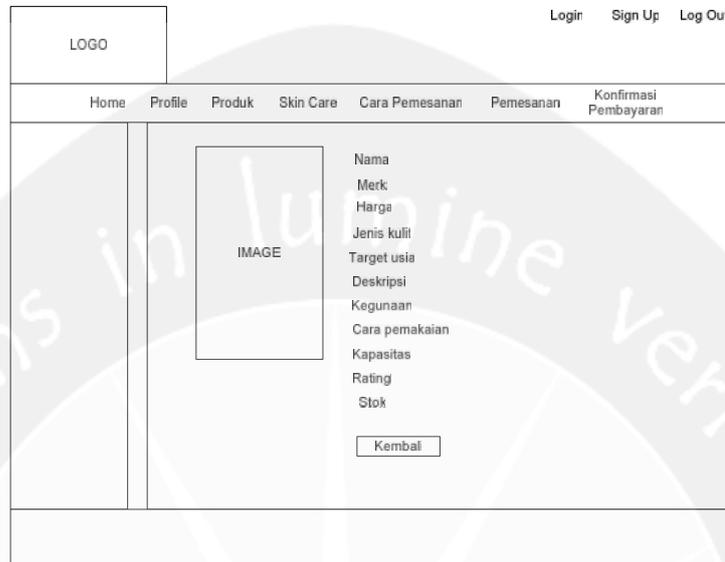
#### 4.6.4 Edit Data Produk

LOGO	Login	Sign Up	Log Out	
Home	Produk	Anggota	Pemesanan	Pembayaran
Input Produk	Nama	<input type="text"/>		
Lihat Produk	Gambar	<input type="text"/>	<input type="button" value="Browse"/>	
	Harga	<input type="text"/>		
	Merk	<input type="text"/>	▼	
	Stok	<input type="text"/>		
	Kategori	<input type="text"/>	▼	
	Rating	<input type="text"/>		
	Deskripsi	<input type="text"/>		
	Cara pakai	<input type="text"/>		
	Kapasitas	<input type="text"/>		
	Jenis kulit	<input type="checkbox"/> Kering	<input type="checkbox"/> Normal	<input type="checkbox"/> Berminyak
		<input type="checkbox"/> Sensitif	<input type="checkbox"/> Kombinasi	<input type="checkbox"/> Semua jenis kulit
	Usia	<input type="text"/>	▼	
	Masalah	<input type="radio"/> Jerawat	<input type="radio"/> Komedo	<input type="radio"/> Whitening
		<input type="button" value="Simpan"/>	<input type="button" value="Batal"/>	

Gambar 4.11 Rancangan Antarmuka Edit Data Produk

User dapat mengedit data produk pada rancangan antarmuka pada gambar 4.11 di atas. Apabila halaman ini terbuka maka sistem secara otomatis telah mengisi form di atas dengan data awal. Untuk mengubah data produk, user dapat langsung mengganti datanya kemudian menekan tombol simpan untuk menyimpan data. Tetapi bila user ingin membatalkan perubahan maka user dapat menekan tombol batal.

#### 4.6.5 Detail Produk



**Gambar 4.12 Rancangan Antarmuka Detail Produk**

Rancangan antarmuka gambar 4.12 digunakan untuk menampilkan data produk tertentu secara detail. Saat user menekan kata lihat detail pada halaman lihat produk 2 maka halaman ini akan muncul dengan data produk yang lebih detail. Untuk kembali ke halaman awal maka user dapat menekan tombol kembali.

#### 4.7 Simulasi Produk

LOGO Login Sign Up Log Out

Home Profile Produk Skin Care Cara Pemesanan Pemesanan Konfirmasi Pembayaran

Usia  Tahun

Pilih berdasarkan  Jenis Kulit  Masalah Kulit

Jenis Kulit

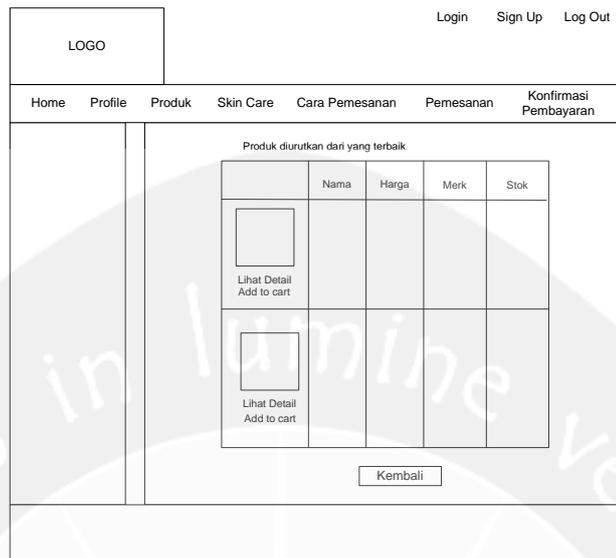
Masalah Kulit

Pilih Produk:  Toner  Cleanser  Moisturizer  Masker

Harga:  Kurang dari Rp80.000  
 Rp 80.000 - Rp 160.000  
 Lebih dari Rp 160.000

Gambar 4.13 Rancangan Antarmuka Skin Care 1

Rancangan antarmuka pada gambar 4.13 merupakan form untuk mencari tahu produk yang terbaik bagi user. Pertama – tama user memasukkan kriteria terlebih dahulu yaitu usia, kategori pemilihan produk (jenis kulit / masalah kulit), jenis produk yang dicari, dan rentang harga produk. Untuk melanjutkan ke tahap berikutnya maka user menekan tombol go.



**Gambar 4.14 Rancangan Antarmuka Skin Care 2**

Rancangan antarmuka pada gambar 4.14 menampilkan hasil dari perhitungan berdasarkan kriteria user. Produk akan ditampilkan urut berdasarkan yang terbaik bagi user.

## 4.8 Pengelolaan Kategori

LOGO

Login Sign Up Log Out

Home Produk User Pembayaran

Id Kategori	Nama Kategori	Edit	Delete
		Edit	Delete

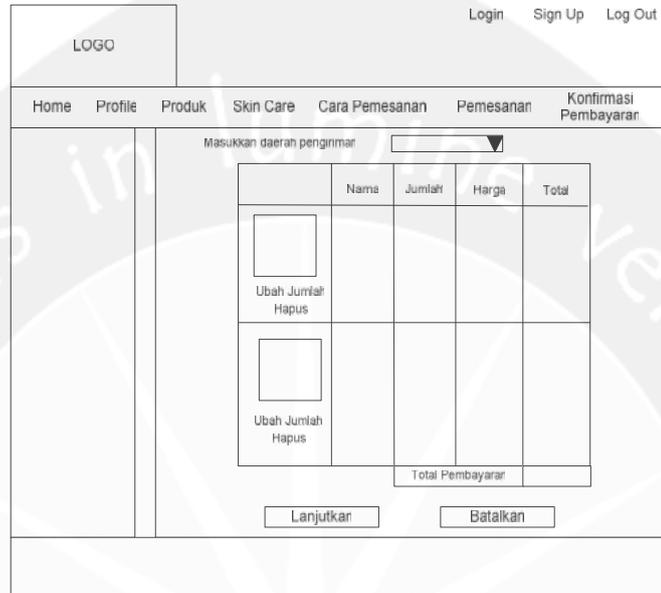
Tambahkan

**Gambar 4.15 Rancangan Antarmuka Pengelolaan Kategori**

Rancangan antarmuka pada gambar 4.15 merupakan tabel yang dapat menambahkan, mengubah sekaligus menghapus data kategori. Untuk menambahkan data maka user dapat langsung menuliskan nama kategori di kotak yang telah disediakan kemudian menekan tombol tambahkan untuk menyimpan data. Maka sistem secara otomatis akan memperbaharui data tabel. User dapat mengubah atau menghapus data kategori dengan menekan kata edit atau delete pada tabel.

## 4.9 Pemesanan

### 4.9.1 Edit Pemesanan



	Nama	Jumlah	Harga	Total
<input type="text"/> Ubah Jumlah Hapus				
<input type="text"/> Ubah Jumlah Hapus				
Total Pembayaran				

Lanjutkan      Batalan

**Gambar 4.16 Rancangan Antarmuka Edit Pemesanan**

User dapat mengubah dan menghapus data pemesanan pada gambar 4.16 di atas. Data yang dapat diubah adalah jumlah produk yang dipesan, sedangkan untuk menghapus produk yang dipesan user dapat langsung memilih produk yang akan dihapus dengan menekan kata hapus. Bila user ingin membatalkan semua pemesanan, user dapat menekan tombol batalan, sedangkan bila user ingin melanjutkan proses pemesanan maka user dapat menekan tombol lanjutkan yang akan membawa user ke halaman konfirmasi pemesanan.

#### 4.9.2 Konfirmasi Pemesanan

LOGO		Login Sign Up Log Out					
Home	Profile	Produk	Skin Care	Cara Pemesanan	Pemesanan	Konfirmasi Pembayaran	
No Nota	Nama	Alamat		Email	No Hp	Total	Tgl Bayar
			Ubah				
Kirim Email				Kembali			

Gambar 4.17 Rancangan Antarmuka Konfirmasi Pemesanan

Rancangan antarmuka pada gambar 4.17 merupakan halaman untuk mengkonfirmasi user akan pemesanan yang telah dilakukan. Bila user ingin membatalkan semua pemesanan maka user dapat menekan tombol kembali. Tetapi bila user ingin melanjutkan data pemesanan telah tersimpan dan user dapat mengirimkan detail pemesanan ke email user dengan menekan tombol kirim email.

#### 4.10 Pengelolaan Pemesanan

LOGO	Login	Sign Up	Log Out		
Home	Produk	Anggota	Pemesanan	Pembayaran	
Delete Data Pemesanan	Cari berdasarkan <input type="text"/> <input type="text"/> <input type="button" value="Cari"/>				
	No Nota	Username	Tgl Pemesanan	Nama Produk	Jumlah

**Gambar 4.18 Rancangan Antarmuka Pengelolaan Pemesanan**

Rancangan antarmuka pada gambar 4.18 menampilkan data pemesanan yang telah terjadi. Untuk mencari data tertentu maka user dapat mencari menurut kategori dan memasukkan kata kuncinya kemudian menekan tombol cari. Bila ditemukan maka tabel akan menampilkan data pemesanan yang sesuai.

Menu di samping yaitu delete data pemesanan digunakan untuk menghapus data pemesanan kadaluwarsa. Pemesanan yang kadaluwarsa adalah pemesanan yang dilakukan tanpa menekan tombol lanjutkan atau batalkan sehingga data tetap tersimpan.

#### 4.11 Pengelolaan Pembayaran

LOGO	Login Sign Up Log Out							
Home	Produk	Anggota	Pemesanan	Pembayaran				
Delete Data Pembayaran	Cari berdasarkan <input type="text"/> <input type="button" value="Cari"/>							
	No Nota	Total	Tgl Bayar	Status	Tgl Jatuh Bayar	Nama rekening		
							Ubah Status	Hapus
							Ubah Status	Hapus
							Ubah Status	Hapus
							Ubah Status	Hapus
							Ubah Status	Hapus

Gambar 4.19 Rancangan Antarmuka Kelola Pembayaran

Rancangan antarmuka pada gambar 4.19 menampilkan data pembayaran yang telah terjadi. Untuk mencari data tertentu maka user dapat mencari menurut kategori dan memasukkan kata kuncinya kemudian menekan tombol cari. Bila ditemukan maka tabel akan menampilkan data pembayaran yang sesuai. Untuk mengubah status pembayaran, maka user dapat langsung menekan kata ubah status dan menyimpan perubahan. Tetapi bila user ingin menghapus data pembayaran maka user dapat menekan kata hapus pada baris data yang ingin dihapus.

Menu di samping yaitu delete data pembayaran digunakan untuk menghapus data pembayaran yang telah kadaluwarsa. Pembayaran yang kadaluwarsa adalah pembayaran yang tidak dilakukan hingga tanggal jatuh pembayaran. Maka pemesanan tersebut dianggap batal.

Gambar 4.21 di bawah merupakan antarmuka untuk melihat detail nomor nota tertentu. Untuk mengakses antarmuka di bawah maka user dapat menekan nomor nota yang ingin dilihat datanya pada antarmuka pengelolaan pembayaran. Maka akan tampil segala yang berhubungan nomor nota yang ditekan.

Gambar 4.20 Rancangan Antarmuka Detail Nota

#### 4.12 Konfirmasi Pembayaran

Gambar 4.21 Rancangan Antarmuka Konfirmasi Pembayaran

Rancangan antarmuka pada gambar 4.21 merupakan form bagi user yang telah melakukan pembayaran. User dapat melakukan konfirmasi pembayaran dengan memasukkan nomor nota, total yang telah ditransfer, nama rekening yang digunakan untuk mentransfer, dan tanggal melakukan pembayaran atau transfer. Kemudian apabila user menekan tombol submit maka status pembayaran user akan berubah dari belum lunas menjadi konfirmasi secara otomatis.



Program Studi Teknik Informatika	DPPL – SCA	71/ 71
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### Kuesioner Aplikasi Skin Care

Nama :

Pekerjaan :

Petunjuk : Berikan tanda centang ( ✓ ) pada kotak di kolom jawaban

No.	Pernyataan	Jawaban				
1	Penggunaan warna dalam aplikasi	<input type="checkbox"/> Sangat kurang	<input type="checkbox"/> Kurang	<input type="checkbox"/> Biasa	<input type="checkbox"/> Bagus	<input type="checkbox"/> Sangat bagus
2	Penggunaan gambar dalam aplikasi	<input type="checkbox"/> Sangat kurang	<input type="checkbox"/> Kurang	<input type="checkbox"/> Biasa	<input type="checkbox"/> Bagus	<input type="checkbox"/> Sangat bagus
3	Tampilan aplikasi secara keseluruhan	<input type="checkbox"/> Sangat kurang	<input type="checkbox"/> Kurang	<input type="checkbox"/> Biasa	<input type="checkbox"/> Bagus	<input type="checkbox"/> Sangat bagus
4	Kemudahan berbelanja	<input type="checkbox"/> Sangat mudah	<input type="checkbox"/> Mudah	<input type="checkbox"/> Sulit	<input type="checkbox"/> Sangat sulit	
5	Kelengkapan informasi	<input type="checkbox"/> Sangat lengkap	<input type="checkbox"/> Lengkap	<input type="checkbox"/> Tidak lengkap	<input type="checkbox"/> Sangat tidak lengkap	
6	Kemudahan penggunaan simulasi	<input type="checkbox"/> Sangat mudah	<input type="checkbox"/> Mudah	<input type="checkbox"/> Sulit	<input type="checkbox"/> Sangat sulit	
7	Ketepatan simulasi dalam memilihkan produk	<input type="checkbox"/> Sangat tepat	<input type="checkbox"/> Tepat	<input type="checkbox"/> Kurang tepat	<input type="checkbox"/> Tidak tepat	

### Kuesioner Kosmetik Perawatan Kulit

Nama :  
Pekerjaan :

Petunjuk : Berikan tanda centang ( ✓ ) pada kotak di kolom jawaban

No.	Pernyataan	Jawaban				
1	Usia	<input type="checkbox"/> 20 tahun	<input type="checkbox"/> 21-25 tahun	<input type="checkbox"/> 25-30 tahun	<input type="checkbox"/> 30-40 tahun	<input type="checkbox"/> >40 tahun
2	Penghasilan per bulan	<input type="checkbox"/> <Rp 1juta	<input type="checkbox"/> Rp 1.1juta-3 juta	<input type="checkbox"/> Rp 3.1juta-5juta	<input type="checkbox"/> >Rp 5juta	
3	Pertimbangan dalam memilih produk kosmetik perawatan kulit	<input type="checkbox"/> Harga	<input type="checkbox"/> Merk	<input type="checkbox"/> Kemasan	<input type="checkbox"/> Rekomendasi	
4	Merk yang sering digunakan	<input type="checkbox"/> Body Shop	<input type="checkbox"/> Face Shop	<input type="checkbox"/> Oriflame	<input type="checkbox"/> Etude	<input type="checkbox"/> .....
5	Jenis produk yang sering digunakan	<input type="checkbox"/> Cleanser	<input type="checkbox"/> Toner	<input type="checkbox"/> Mousturizer	<input type="checkbox"/> Masker	<input type="checkbox"/> .....