

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya dan pengujian aplikasi yang dibuat, maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem Informasi Klinik Hewan Calico telah berhasil dibuat dan dapat dipergunakan untuk mempermudah pencatatan data hewan di klinik hewan Calico.
2. *Website* Calico yang dibuat mampu untuk memberikan informasi tentang klinik dan pelayanan yang disediakan kepada masyarakat secara lengkap dan tepat serta informasi tambahan lainnya.
3. Dengan penggunaan HTML5 pada sistem, maka penanganan validasi *form* menjadi lebih baik serta adanya elemen-elemen baru seperti *sectioning elements* (<nav>, <section>, <aside>, <footer>), *embedded elements* (<audio>, <video>, <input>) dari HTML5 yang dapat menambah fungsionalitas web baik untuk *web developer* maupun *user* pengguna.

#### 6.2 Saran

Saran yang dapat diambil dari proses analisis sampai pada pembuatan tugas akhir ini adalah sistem diharapkan dapat dikembangkan lagi dengan menambahkan pengelolaan data keuangan, data obat, dan data inventaris barang yang ada di klinik yang dapat membuat

sistem semakin handal dan proses kerja di klinik hewan Calico menjadi lebih mudah.



## DAFTAR PUSTAKA

Amadatu, Yuvenjustus, 2010, ***Pengembangan Aplikasi Pelaporan Data Operasional Puskesmas Berbasis Web***, Universitas Atma Jaya, Yogyakarta

Angelov, Martin, 2010, <http://tutorialzine.com>, 27 November 2010

Haryatmoko, Kuntoro, 2007, ***Pembangunan Aplikasi Web Retail Komputer dengan AJAX Berbasis Teknologi .NET***, Universitas Atma Jaya, Yogyakarta

Keith, Jeremy, 2010, ***HTML5 for Web Designers***, A Book Apart, New York

Lawson, Bruce, 2011, ***Introducing HTML5***, New Riders, California

Malfi, M., 2009, ***Pembuatan Sistem Informasi Rekam Medis di Rumah Sakit Dr. Mohammad Hoesin Palembang***, Universitas Sriwijaya, Palembang

Nuryanto, Eka, 2008, ***Perancangan dan Implementasi Sistem Informasi Pelayanan Kesehatan pada Klinik***, Universitas Islam Sultan Agung, Semarang

Pilgrim, Mark, 2010, <http://diveintohtml5.org>, 26 November 2010

Shepherd, Richard, 2010, <http://richardshepherd.com>, 27 November 2010

Teurupun, Rosi, 2008, ***Pembangunan Sistem Informasi pelayanan Pasien Berbasis Web dengan AJAX.Net***, Universitas Atma Jaya, Yogyakarta

Way, Jeffrey, 2010, <http://nettuts.com>, 27 November 2010



**LAMP IRAN**

# SKPL

## SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

### Silico

(Sistem Informasi Klinik Hewan Calico)

Untuk :

Klinik Hewan Calico

Dipersiapkan oleh:

Duhita Anggayasti / 5017

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>SKPL-Silico</i>		1/49
		Revisi		

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	Revisi referensi, arsitektur perangkat lunak, Use Case Diagram, ERD
<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
8	Referensi		
10	Arsitektur perangkat lunak		
17	Use Case Diagram		
32	ERD		

## Daftar Isi

1	Pendahuluan.....	127
1.1	Tujuan.....	127
1.2	Lingkup Masalah.....	127
1.3	Definisi, Akronim dan Singkatan.....	127
1.4	Referensi.....	129
1.5	Deskripsi umum (Overview).....	130
2	Deskripsi Kebutuhan.....	9
2.1	Perspektif produk.....	9
2.2	Fungsi Produk.....	132
2.3	Karakteristik Pengguna.....	139
2.4	Batasan-batasan.....	140
2.5	Asumsi dan Ketergantungan.....	141
3	Kebutuhan khusus.....	141
3.1	Kebutuhan antarmuka eksternal.....	141
3.2	Kebutuhan fungsionalitas Perangkat Lunak.....	144
4	Spesifikasi Rinci Kebutuhan.....	145
4.1	Spesifikasi Kebutuhan Fungsionalitas.....	145
	5 ERD 170	
6	Kamus Data.....	170

## Daftar Gambar

1. Arsitektur Perangkat Lunak Silico ..... 10
2. Use Case Diagram ..... 20



## 1. Pendahuluan

### 1.1. Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak Silico (Sistem Informasi Klinik Hewan Calico) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna), performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-Silico ini juga mendefinisikan batasan perancangan perangkat lunak.

### 1.2. Lingkup Masalah

Perangkat lunak Silico dikembangkan dengan tujuan untuk :

6. Menangani pengelolaan data pendaftaran hewan.
7. Menangani pengelolaan data rekam medis hewan.
8. Menangani pengelolaan data penitipan hewan.
9. Menangani pengelolaan data grooming.
10. Menangani pengelolaan surat keterangan sehat.

Perangkat lunak Silico memiliki layanan Web-based Form yang dapat diakses melalui internet.

### 1.3. Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari

	perangkat lunak yang akan dikembangkan.
SKPL-Silico-XXX	Kode yang merepresentasikan kebutuhan pada Silico (Sistem Informasi Klinik Hewan Calico) dimana XXX merupakan nomor fungsi produk.
Silico	Perangkat lunak pengelolaan data klinik hewan Calico.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
Client	Komputer yang meminta sumber daya dari server yang terhubung melalui jaringan.
Aplikasi Web	Perangkat lunak pelayanan dan pengolahan informasi pengguna berbasis Web.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Role	Hak yang dimiliki oleh user untuk menggunakan system/hak akses. Role diatur oleh administrator.
Database	Kumpulan data yang terkait yang diorganisasikan dalam struktur tertentu dan dapat diakses dengan cepat.
Administrator	Orang yang bertugas mengelola sistem secara keseluruhan serta membuat account bagi pengguna sistem tersebut.
Dokter	Hak akses dari administrator yang dapat

	digunakan oleh dokter hewan yang ada di klinik dimana dokter dapat menggunakan fungsi-fungsi pada sistem sesuai dengan hak aksesnya.
Customer Service	Hak akses dari administrator yang dapat digunakan oleh customer service di klinik hewan dimana customer service dapat menggunakan fungsi-fungsi pada sistem sesuai dengan hak aksesnya.
Pengunjung	Hak akses dari administrator yang dapat digunakan oleh pengunjung situs dimana pengunjung dapat menggunakan fungsi-fungsi pada sistem sesuai dengan hak aksesnya.
Grooming	Perawatan yang disediakan untuk penampilan fisik anjing (terutama bulu dan kuku), atau pekerjaan spesifik dilakukan untuk meningkatkan tampilan untuk pertunjukan.
Rekam Medis	Keterangan baik yang tertulis maupun terekam tentang identitas, anamnesa, penentuan fisik, laboratorium, diagnosa segala pelayanan dan tindakan medik yang diberikan kepada pasien dan pengobatan baik yang dirawat inap, rawat jalan, maupun yang mendapatkan pelayanan gawat darurat.

#### 1.4. Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah :

1. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
2. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.
3. Deitel, *C# How to Program*, Prentice-Hall Inc, 2002.
4. Kartawan, I Made Surya, 2006, Pengembangan Perangkat Lunak *E-Ticketing* Melalui Perangkat *Mobile* Menggunakan *ASP.NET*, Skripsi, Jurusan Teknik Informatika, Universitas Atma Jaya, Yogyakarta.

### 1.5. Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian pertama adalah Pendahuluan, 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 adalah Deskripsi Kebutuhan, berisi penjelasan umum tentang perangkat lunak Silico 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 Silico tersebut.

Bagian ketiga adalah Kebutuhan Khusus, berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak Silico yang akan dikembangkan.

## **2. Deskripsi Kebutuhan**

### **2.1. Perspektif produk**

Silico merupakan perangkat lunak yang dikembangkan untuk membantu pengelolaan data di klinik hewan Calico. Sistem ini dapat menangani pengelolaan data pendaftaran pasien hewan, pengelolaan data rekam medis hewan, pengelolaan data penitipan hewan, pengelolaan data grooming hewan, dan pengelolaan surat keterangan sehat bagi hewan.

Perangkat lunak Silico ini berjalan pada platform Windows XP, dan dibuat menggunakan bahasa pemrograman HTML5. Sedangkan untuk lingkungan pemrogramannya menggunakan Notepad++ dan Microsoft Visual Studio 2008.

Sistem informasi ini memiliki aplikasi web-based.

#### **a. Web-based**

Sistem memiliki mekanisme login yang bisa membedakan role beberapa user seperti: administrator, dokter, dan pengunjung.

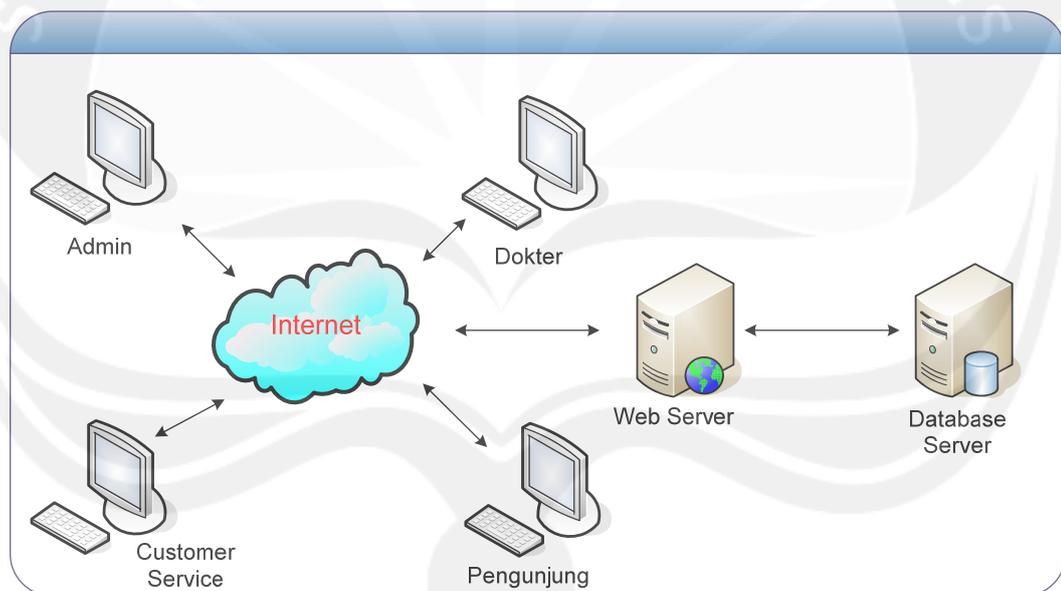
Bagian administrator mengelola account untuk dokter, customer service, dan pengunjung. Administrator dapat melakukan proses entry, edit, delete, dan display account dokter, customer service, dan pengunjung, serta menghapus thread post.

Bagian dokter dan customer service dapat mengelola data rekam medis, data hewan dan pemiliknya, data grooming, data penitipan hewan, dan surat keterangan sehat bagi hewan.

Bagian pengunjung dapat ikut serta berperan aktif dalam forum dengan membuat topik baru dan memberikan tanggapan atas topik yang ada.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (Graphical User Interface). Arsitektur perangkat lunak yang digunakan berupa client server, di mana semua data disimpan di server. User dapat mengakses data yang ada pada database server melalui akses online melalui web server yang terhubung ke internet.

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 client yang merequest melalui web server.



Gambar 2. Arsitektur Perangkat Lunak Silico

## 2.2. Fungsi Produk

Berikut dijelaskan fungsi aplikasi utama pada Sistem Informasi Klinik Hewan Calico:

- User Semua

1. Fungsi *Login* (**SKPL-Silico-001**)

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

□ User Administrator

3. Fungsi *Pengelolaan Account Dokter* (**SKPL-Silico-002**)

Pengelolaan account dokter ini memiliki beberapa fungsi, yaitu :

a. Fungsi *Entry Account Dokter* (**SKPL-Silico-002-01**), adalah fungsi untuk menambah data dokter beserta perannya kedalam database Silico.

b. Fungsi *Edit Account Dokter* (**SKPL-Silico-002-02**), adalah fungsi untuk melakukan perubahan data dokter di dalam database Silico.

c. Fungsi *Delete Account Dokter* (**SKPL-Silico-002-03**), adalah fungsi untuk menghapus data dokter di dalam database Silico.

d. Fungsi *Display Account Dokter* (**SKPL-Silico-002-04**), adalah fungsi untuk menampilkan data dokter dari database Silico.

2. Fungsi *Pengelolaan Account Customer Service* (**SKPL-Silico-003**)

Pengelolaan account customer service ini memiliki beberapa fungsi, yaitu :

a. Fungsi *Entry Account Customer Service* (**SKPL-Silico-003-01**), adalah fungsi untuk menambah data customer service beserta perannya kedalam database Silico.

b. Fungsi *Edit Account Customer Service* (**SKPL-Silico-003-02**), adalah fungsi untuk melakukan perubahan data customer service di dalam database Silico.

c. Fungsi *Delete Account Customer Service* (**SKPL-Silico-003-03**), adalah fungsi untuk menghapus data customer service di dalam database Silico.

d. Fungsi *Display Account Customer Service* (**SKPL-Silico-003-04**), adalah fungsi untuk menampilkan data customer service dari database Silico.

3. Fungsi *Pengelolaan Account Pengunjung* (**SKPL-Silico-004**)

Pengelolaan account pengunjung ini memiliki beberapa fungsi, yaitu :

a. Fungsi *Display Account Pengunjung* (**SKPL-Silico-004-01**), adalah fungsi untuk menampilkan data pengunjung yang mendaftar forum dari database Silico.

4. Fungsi *Pengelolaan Forum* (**SKPL-Silico-005**)

Pengelolaan forum ini memiliki fungsi :

a. Fungsi *Delete Thread Post* (**SKPL-Silico-005-001**), adalah fungsi untuk menghapus thread post yang bermasalah dalam forum.

□ User Customer Service

1. Fungsi *Pengelolaan Pendaftaran Pasien Hewan* (**SKPL-Silico-006**)

Merupakan fungsi yang digunakan untuk mengelola data pendaftaran pasien hewan.

Fungsi Pengelolaan *Pendaftaran Pasien Hewan* meliputi :

a. Fungsi *Entry Data Pasien Hewan Baru* (**SKPL-Silico-006-01**)

Merupakan fungsi yang digunakan untuk menambah data pendaftaran hewan yang datang ke klinik.

b. Fungsi *Entry Data Pasien Hewan Lama* (**SKPL-Silico-006-02**)

Merupakan fungsi yang digunakan untuk menambah data pendaftaran hewan dengan pemilik lama yang datang ke klinik.

c. Fungsi *Edit Data Pasien Hewan* (**SKPL-Silico-006-03**)

Merupakan fungsi yang digunakan untuk mengubah data pendaftaran hewan yang datang ke klinik.

d. Fungsi *Display Data Pasien Hewan* (**SKPL-Silico-006-04**)

Merupakan fungsi yang digunakan untuk menampilkan data pendaftaran hewan yang pernah datang ke klinik.

2. Fungsi *Pengelolaan Data Penitipan Hewan* (**SKPL-Silico-007**)

Merupakan fungsi yang digunakan untuk mengelola data penitipan hewan.

Fungsi Pengelolaan Penitipan Hewan mencakup :

a. Fungsi *Entry Data Penitipan Hewan Baru* (**SKPL-Silico-007-01**). Merupakan fungsi yang digunakan

untuk menambahkan data hewan yang dititipkan dimana hewan adalah pasien baru.

b. Fungsi *Entry Data Penitipan Hewan Lama* (**SKPL-Silico-007-02**). Merupakan fungsi yang digunakan untuk menambahkan data hewan yang dititipkan dimana hewan adalah pasien lama.

c. Fungsi *Display Data Penitipan Hewan* (**SKPL-Silico-007-03**).

Merupakan fungsi yang digunakan untuk menampilkan data hewan yang sedang/pernah dititipkan.

d. Fungsi *Cetak Surat Penitipan Hewan* (**SKPL-Silico-007-04**).

Merupakan fungsi yang digunakan untuk mencetak surat penitipan hewan.

3. Fungsi *Pengelolaan Data Grooming* (**SKPL-Silico-008**)

Merupakan fungsi yang digunakan untuk mengelola data hewan yang menggunakan jasa grooming.

Fungsi Pengelolaan Data Grooming meliputi :

a. Fungsi *Entry Data Grooming Baru* (**SKPL-Silico-008-01**)

Merupakan fungsi yang digunakan untuk menambahkan data hewan pasien baru yang menggunakan jasa grooming.

b. Fungsi *Entry Data Grooming Lama* (**SKPL-Silico-008-02**)

Merupakan fungsi yang digunakan untuk menambahkan data hewan pasien lama yang menggunakan jasa grooming.

c. Fungsi *Display Data Grooming* (**SKPL-Silico-008-03**)

Merupakan fungsi yang digunakan untuk menampilkan data hewan yang menggunakan jasa grooming.

□ User Dokter

1. Fungsi *Pengelolaan Data Rekam Medis Hewan* (**SKPL-Silico-009**)

Merupakan fungsi yang digunakan untuk mengelola data rekam medis hewan.

Fungsi *Pengelolaan Data Rekam Medis Hewan* meliputi :

a. Fungsi *Entry Data Rekam Medis Hewan* (**SKPL-Silico-009-01**)

Merupakan fungsi yang digunakan untuk memasukkan data rekam medis hewan.

b. Fungsi *Edit Data Rekam Medis Hewan* (**SKPL-Silico-009-02**)

Merupakan fungsi yang digunakan untuk mengubah data rekam medis hewan.

c. Fungsi *Display Data Rekam Medis Hewan* (**SKPL-Silico-009-03**)

Merupakan fungsi yang digunakan untuk menampilkan data rekam medis hewan.

d. Fungsi *Display History Rekam Medis Hewan* (**SKPL-Silico-009-04**)

Merupakan fungsi yang digunakan untuk menampilkan data history rekam medis hewan yaitu data rekam medis yang telah diubah.

2. Fungsi *Pengelolaan Surat Keterangan Sehat* (**SKPL-Silico-010**)

Merupakan fungsi yang digunakan untuk membuat surat keterangan sehat bagi hewan.

Fungsi Pengelolaan Surat Keterangan Sehat meliputi :

a. Fungsi *Cetak Data Surat Keterangan Sehat* (**SKPL-Silico-010-001**)

Merupakan fungsi yang digunakan untuk menampilkan data surat keterangan sehat.

□ User Pengunjung

1. Fungsi *Pendaftaran User* (**SKPL-Silico-011**)

Merupakan fungsi yang digunakan untuk mendaftarkan diri sebagai user dalam forum yang ada.

2. Fungsi *Pengelolaan Forum* (**SKPL-Silico-012**)

Merupakan fungsi yang digunakan dalam menggunakan forum.

a. Fungsi *Entry Thread Baru* (**SKPL-Silico-012-001**)

Merupakan fungsi yang digunakan untuk menambahkan thread baru.

b. Fungsi *Entry Thread Reply* (**SKPL-Silico-012-002**)

Merupakan fungsi yang digunakan untuk menambahkan thread reply terhadap thread yang ada.

### 2.3. Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak Silico adalah sebagai berikut :

1. Administrator

- ❑ Memahami pengoperasian komputer dengan system operasi Windows XP.
- ❑ Mengerti tentang internet dan memahami penggunaan internet browser.
- ❑ Mengerti proses pengelolaan data account dokter, customer service, dan pengunjung.

2. Dokter

- ❑ Memahami pengoperasian komputer dengan system operasi Windows XP.
- ❑ Mengerti dan memahami perangkat lunak yang digunakan.
- ❑ Mengerti proses pengelolaan data rekam medis hewan.
- ❑ Mengerti proses pengelolaan data pendaftaran pasien hewan.
- ❑ Mengerti proses pengelolaan data grooming.

- ❑ Mengerti proses pengelolaan data penitipan hewan.
- ❑ Mengerti proses pengelolaan surat keterangan sehat.

### 3. Customer Service

- ❑ Memahami pengoperasian komputer dengan system operasi Windows XP.
- ❑ Mengerti dan memahami perangkat lunak yang digunakan.
- ❑ Mengerti proses pengelolaan data rekam medis hewan.
- ❑ Mengerti proses pengelolaan data pendaftaran pasien hewan.
- ❑ Mengerti proses pengelolaan data grooming.
- ❑ Mengerti proses pengelolaan data penitipan hewan.
- ❑ Mengerti proses pengelolaan surat keterangan sehat.

### 4. Pengunjung

- ❑ Memahami pengoperasian komputer dengan system operasi Windows XP.
- ❑ Mengerti dan memahami perangkat lunak yang digunakan.
- ❑ Mengerti tentang internet dan memahami penggunaan internet browser.

## **2.4. Batasan-batasan**

Batasan-batasan dalam pengembangan perangkat lunak Silico tersebut adalah :

### **1. Kebijakan Umum**

Berpedoman pada tujuan dari pengembangan perangkat lunak Silico.

### **2. Keterbatasan perangkat keras**

Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

## **2.5. Asumsi dan Ketergantungan**

Sistem ini dapat dijalankan pada perangkat komputer dengan system operasi Windows XP yang terhubung ke internet. Mekanisme login pada system dapat membedakan role dari user. Layanan web-based dapat membedakan role beberapa user yaitu administrator, dokter, customer service, dan pengunjung. Perangkat lunak ini dibuat dengan asumsi bahwa user mampu menggunakan internet dan mengerti cara mengakses sistem sesuai dengan role yang dimilikinya.

## **3. Kebutuhan khusus**

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

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

### **3.1.1. Antarmuka pemakai**

Untuk layanan web-based, pengguna berinteraksi dengan antarmuka yang ditampilkan pada layar computer dengan format tampilan halaman web, dan antarmuka ini memungkinkan user untuk mengakses fungsionalitas system dari lokasi manapun selama terhubung dengan layanan internet.

### **3.1.2. Antarmuka perangkat keras**

Antarmuka perangkat keras yang digunakan dalam perangkat lunak Silico adalah:

5. Mouse.
6. Monitor.
7. Keyboard.
8. Modem atau LAN Card.

### **3.1.3. Antarmuka perangkat lunak**

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak Silico 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 XP  
Sumber : Microsoft.  
Sebagai sistem operasi untuk perangkat komputer.
3. Nama : IIS  
Sumber : Microsoft.  
Sebagai web server.

4. Nama : Opera 11

Sumber : Opera

Sebagai browser untuk mengakses aplikasi dari web.

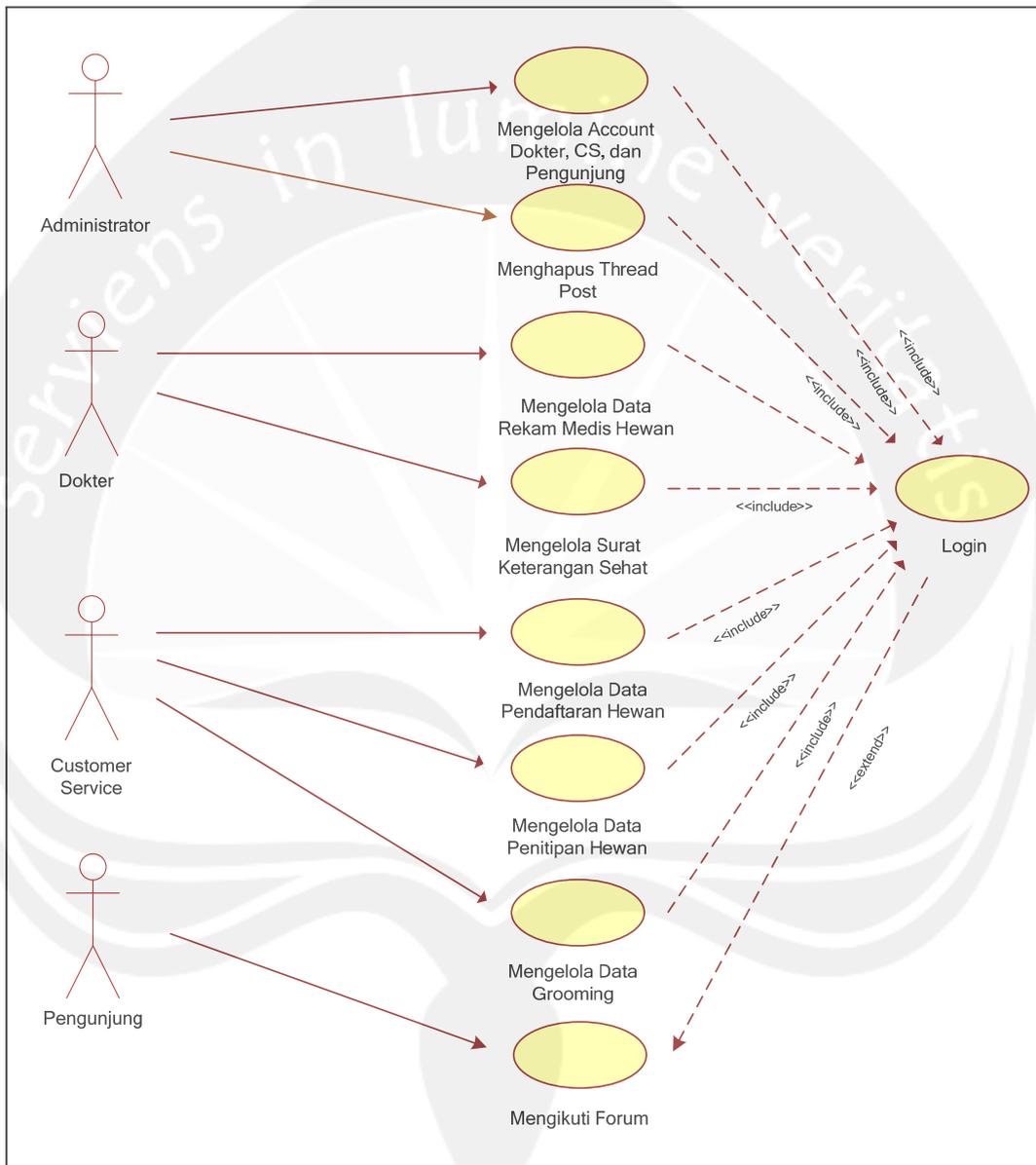
#### 3.1.4. Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak Silico menggunakan protocol TCP/IP.



## 3.2. Kebutuhan fungsionalitas Perangkat Lunak

### 3.2.1. Use Case Diagram



Gambar 3. 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 sebuah id unik yaitu username dan password yang berupa rangkaian karakter.

##### 2. Primary Actor

1. User

##### 3. Supporting Actor

none

##### 4. Basic Flow

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

E-1 Password atau id user tidak valid

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

##### 5. Alternative Flow

none

##### 6. Error Flow

E-1 Password atau nama user tidak ada dalam database

1. Sistem menampilkan peringatan bahwa id user dan password tidak valid

2. Kembali ke Basic Flow langkah ke 3

#### **7. PreConditions**

none

#### **8. PostConditions**

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

#### **4.1.2 Use Case Spesification : Pengelolaan Account Dokter**

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

Use Case ini digunakan oleh aktor untuk mengelola account dari dokter. Aktor dapat melakukan entry data account, edit data account, delete data account, atau display data account.

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

1. Administrator

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data account dokter dan pengunjung.
2. Sistem memberikan pilihan untuk melakukan entry data account, edit data account, delete data account, atau display data account.
3. Aktor memilih untuk melakukan entry data account  
A-1 Aktor memilih untuk melakukan edit data account

A-2 Aktor memilih untuk melakukan delete data account

A-3 Aktor memilih untuk melakukan display data account

4. Sistem menampilkan form pengisian data
5. Aktor menginputkan data account
6. Aktor meminta sistem untuk menyimpan data account yang telah diinputkan
7. Sistem mengecek data account yang telah diinputkan

E-1 Data account yang diinputkan aktor salah

8. Sistem menyimpan data account ke database
9. Use Case selesai

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

A-1 Aktor memilih untuk melakukan edit data account

1. Sistem meminta id account yang akan diedit
2. Aktor menginputkan id account yang akan diedit datanya
3. Sistem melakukan pengecekan id account ke dalam database

E-2 Data account yang akan diedit tidak valid

4. Aktor mengedit data account yang sudah ditampilkan
5. Aktor meminta sistem untuk menyimpan data account yang telah diedit
6. Sistem melakukan pengecekan terhadap data account yang telah diedit

E-3 Data account yang telah diedit aktor salah

7. Sistem menyimpan data account yang telah diedit ke database

8. Berlanjut ke Basic Flow langkah ke 9

A-2 Aktor memilih untuk melakukan delete data account

1. Sistem menampilkan seluruh data account

2. Aktor memilih data account yang akan dihapus

3. Sistem memberikan konfirmasi apakah aktor akan menghapus data atau tidak

4. Aktor meminta sistem untuk menghapus data account

5. Sistem menghapus data account dari database

6. Berlanjut ke Basic Flow langkah ke 9

A-3 Aktor memilih untuk melakukan display data account

1. Sistem menampilkan data atau profile seluruh account

2. Berlanjut ke Basic Flow langkah ke 8

## **6. Error Flow**

E-1 Data account yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah

2. Kembali ke Basic Flow Langkah ke 4

E-2 Data account yang akan diedit tidak valid

1. Sistem memberikan pesan peringatan bahwa data account yang akan diedit tidak valid

2. Kembali ke Alternative Flow A-1 Langkah ke  
2

E-3 Data account yang telah diedit aktor salah

1. Sistem memberikan pesan peringatan bahwa data account yang telah diedit salah
2. Kembali ke Alternative Flow A-2 Langkah ke  
4

#### **7. PreConditions**

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

#### **8. PostConditions**

1. Data account di database telah terupdate

#### **4.1.3 Use Case Spesification : Pengelolaan Account Customer Service**

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

Use Case ini digunakan oleh aktor untuk mengelola account dari customer service. Aktor dapat melakukan entry data account, edit data account, delete data account, atau display data account.

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

1. Administrator

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data account customer service

2. Sistem memberikan pilihan untuk melakukan entry data account, edit data account, delete data account, atau display data account.
3. Aktor memilih untuk melakukan entry data account
  - A-1 Aktor memilih untuk melakukan edit data account
  - A-2 Aktor memilih untuk melakukan delete data account
  - A-3 Aktor memilih untuk melakukan display data account
4. Sistem menampilkan form pengisian data
5. Aktor menginputkan data account
6. Aktor meminta sistem untuk menyimpan data account yang telah diinputkan
7. Sistem mengecek data account yang telah diinputkan
  - E-1 Data account yang diinputkan aktor salah
8. Sistem menyimpan data account ke database
9. Use Case selesai

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

- A-1 Aktor memilih untuk melakukan edit data account
1. Sistem meminta id account yang akan diedit
  2. Aktor menginputkan id account yang akan diedit datanya
  3. Sistem melakukan pengecekan id account ke dalam database

E-2 Data account yang akan diedit tidak valid

4. Aktor mengedit data account yang sudah ditampilkan
5. Aktor meminta sistem untuk menyimpan data account yang telah diedit
6. Sistem melakukan pengecekan terhadap data account yang telah diedit

E-3 Data account yang telah diedit aktor salah

7. Sistem menyimpan data account yang telah diedit ke database
8. Berlanjut ke Basic Flow langkah ke 9

A-2 Aktor memilih untuk melakukan delete data account

1. Sistem menampilkan seluruh data account
2. Aktor memilih data account yang akan dihapus
3. Sistem memberikan konfirmasi apakah aktor akan menghapus data atau tidak
4. Aktor meminta sistem untuk menghapus data account
5. Sistem menghapus data account dari database
6. Berlanjut ke Basic Flow langkah ke 9

A-3 Aktor memilih untuk melakukan display data account

1. Sistem menampilkan data atau profile seluruh account
2. Berlanjut ke Basic Flow langkah ke 8

## 6. Error Flow

E-1 Data account yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah
2. Kembali ke Basic Flow Langkah ke 4

E-2 Data account yang akan diedit tidak valid

3. Sistem memberikan pesan peringatan bahwa data account yang akan diedit tidak valid
4. Kembali ke Alternative Flow A-1 Langkah ke 2

E-3 Data account yang telah diedit aktor salah

1. Sistem memberikan pesan peringatan bahwa data account yang telah diedit salah
2. Kembali ke Alternative Flow A-2 Langkah ke 4

#### **7. PreConditions**

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

#### **8. PostConditions**

1. Data account di database telah terupdate

#### **4.1.4 Use Case Spesification : Pengelolaan Account Pengunjung**

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

Use Case ini digunakan oleh aktor untuk mengelola account pengunjung.

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

1. Administrator

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

None

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan account pengunjung
2. Sistem memberikan pilihan untuk melakukan display data account pengunjung
3. Aktor memilih untuk melakukan display data account pengunjung
  - A-1 Aktor memilih untuk melakukan display data account pengunjung
4. Sistem menampilkan data account pengunjung
5. Use Case selesai

**5. Alternative Flow**

none

**6. Error Flow**

none

**7. PreConditions**

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

**8. PostConditions**

1. Data account pengunjung ditampilkan

**4.1.5 Use Case Spesification : Pengelolaan Forum**

**1. Brief Description**

Use Case ini digunakan oleh aktor untuk mengelola forum yaitu menghapus thread post.

**2. Primary Actor**

1. Administrator

**3. Supporting Actor**

None

**4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan forum yaitu menghapus thread post
2. Sistem menampilkan data thread post
3. Aktor memilih data yang dihapus
4. Data terhapus dari sistem
5. Use Case selesai

**5. Alternative Flow**

none

**6. Error Flow**

none

**7. PreConditions**

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

**8. PostConditions**

1. Data thread post bermasalah terhapus dari sistem

**4.1.6 Use Case Spesification : Pengelolaan Pendaftaran Pasien Hewan**

**1. Brief Description**

Use Case ini memungkinkan user untuk melakukan pengelolaan atas data pendaftaran pasien hewan, seperti entry data, edit data, dan display data.

**2. Primary Actor**

1. Dokter

**3. Supporting Actor**

None

**4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data pendaftaran pasien hewan
2. Sistem memberikan pilihan untuk melakukan entry data pendaftaran pasien hewan, edit data pendaftaran pasien hewan, atau display data pendaftaran pasien hewan.
3. Aktor memilih untuk melakukan entry data pendaftaran pasien hewan
  - A-1 Aktor memilih untuk melakukan edit data pendaftaran pasien hewan
  - A-2 Aktor memilih untuk melakukan display data pendaftaran pasien hewan
4. Sistem menampilkan form pengisian data pendaftaran pasien hewan
5. Aktor menginputkan data pendaftaran pasien hewan
6. Aktor meminta sistem untuk menyimpan data pendaftaran pasien hewan yang telah diinputkan
7. Sistem mengecek data pendaftaran pasien hewan yang telah diinputkan
  - E-1 Data pendaftaran pasien hewan yang diinputkan aktor salah
8. Sistem menyimpan data pendaftaran pasien hewan ke database
9. Use Case selesai

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

- A-1 Aktor memilih untuk melakukan edit data pendaftaran pasien hewan

1. Sistem meminta Id pendaftaran pasien hewan yang akan diedit
  2. Aktor menginputkan Id pendaftaran pasien hewan yang akan diedit datanya
  3. Sistem melakukan pengecekan Id pendaftaran pasien hewan ke dalam database
    - E-2 Data pendaftaran pasien hewan yang akan diedit tidak valid
  4. Aktor mengedit data pendaftaran pasien hewan yang sudah ditampilkan
  5. Aktor meminta sistem untuk menyimpan data pendaftaran pasien hewan yang telah diedit
  6. Sistem melakukan pengecekan terhadap data pendaftaran pasien hewan yang telah diedit
    - E-3 Data pendaftaran pasien hewan yang telah diedit aktor salah
  7. Sistem menyimpan data pendaftaran pasien hewan yang telah diedit ke database
  8. Berlanjut ke Basic Flow langkah ke 9
- A-2 Aktor memilih untuk melakukan display data pendaftaran pasien hewan
1. Sistem meminta Id pendaftaran pasien hewan yang akan didisplay
  2. Aktor menginputkan Id pendaftaran pasien hewan
  3. Sistem melakukan pencarian Id pendaftaran pasien hewan ke dalam database
  4. Sistem menampilkan data atau profile pendaftaran pasien hewan sesuai inputan dari user
  5. Berlanjut ke Basic Flow langkah ke 8

## 6. Error Flow

E-1 Data pendaftaran pasien hewan yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah
2. Kembali ke Basic Flow Langkah ke 4

E-2 Data pendaftaran pasien hewan yang akan diedit tidak valid

1. Sistem memberikan pesan peringatan bahwa data pendaftaran pasien hewan yang akan diedit tidak valid
2. Kembali ke Alternative Flow A-1 Langkah ke 2

E-3 Data pendaftaran pasien hewan yang telah diedit aktor salah

1. Sistem memberikan pesan peringatan bahwa data pendaftaran pasien hewan yang telah diedit salah
2. Kembali ke Alternative Flow A-1 Langkah ke 4

## 7. PreConditions

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

## 8. PostConditions

1. Data pendaftaran pasien hewan di database telah terupdate

**4.1.7 Use Case Spesification :** Pengelolaan Data Rekam Medis Hewan

## 1. Brief Description

Use Case ini memungkinkan user untuk melakukan pengelolaan atas data rekam medis hewan, seperti entry data, edit data, dan display data.

## 2. Primary Actor

1. Dokter

## 3. Supporting Actor

none

## 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data rekam medis hewan
2. Sistem memberikan pilihan untuk melakukan entry data rekam medis, edit data rekam medis, atau display data rekam medis.
3. Aktor memilih untuk melakukan entry data rekam medis
  - A-1 Aktor memilih untuk melakukan edit data rekam medis
  - A-2 Aktor memilih untuk melakukan display data rekam medis
4. Sistem menampilkan form pengisian data
5. Aktor menginputkan data rekam medis
6. Aktor meminta sistem untuk menyimpan data rekam medis yang telah diinputkan
7. Sistem mengecek data rekam medis yang telah diinputkan
  - E-1 Data rekam medis yang diinputkan aktor salah
8. Sistem menyimpan data rekam medis ke database
9. Use Case selesai

## 5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data rekam medis

1. Sistem meminta id rekam medis yang akan diedit
2. Aktor menginputkan id rekam medis yang akan diedit datanya
3. Sistem melakukan pengecekan id rekam medis ke dalam database

E-2 Data rekam medis yang akan diedit tidak valid

4. Aktor mengedit data rekam medis yang sudah ditampilkan
5. Aktor meminta sistem untuk menyimpan data rekam medis yang telah diedit
6. Sistem melakukan pengecekan terhadap data rekam medis yang telah diedit

E-3 Data rekam medis yang telah diedit aktor salah

7. Sistem menyimpan data rekam medis yang telah diedit ke database
8. Berlanjut ke Basic Flow langkah ke 9

A-2 Aktor memilih untuk melakukan display data rekam medis

1. Sistem meminta id rekam medis yang akan didisplay
2. Aktor menginputkan id rekam medis
3. Sistem melakukan pencarian id rekam medis ke dalam database
4. Sistem menampilkan data rekam medis sesuai inputan dari user

5. Berlanjut ke Basic Flow langkah ke 8

## 6. Error Flow

E-1 Data rekam medis yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah
2. Kembali ke Basic Flow Langkah ke 4

E-2 Data rekam medis yang akan diedit tidak valid

1. Sistem memberikan pesan peringatan bahwa data rekam medis yang akan diedit tidak valid
2. Kembali ke Alternative Flow A-1 Langkah ke 2

E-3 Data rekam medis yang telah diedit aktor salah

1. Sistem memberikan pesan peringatan bahwa data rekam medis yang telah diedit salah
2. Kembali ke Alternative Flow A-1 Langkah ke 4

## 7. PreConditions

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

## 8. PostConditions

1. Data rekam medis di database telah terupdate

### 4.1.8 Use Case Spesification : Pengelolaan Data Penitipan Hewan

## 1. Brief Description

Use Case ini memungkinkan aktor untuk mengelola data penitipan hewan seperti entry data dan display data penitipan hewan.

## **2. Primary Actor**

1. Dokter

## **3. Supporting Actor**

none

## **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data penitipan hewan
2. Sistem memberikan pilihan untuk melakukan entry data penitipan hewan baru, entry data penitipan hewan lama, atau display data penitipan hewan
3. Aktor memilih untuk melakukan entry data penitipan hewan baru
  - A-1 Aktor memilih untuk melakukan entry data penitipan hewan lama
  - A-2 Aktor memilih untuk melakukan edit data penitipan hewan
4. Sistem menampilkan form pengisian data
5. Aktor menginputkan data penitipan hewan
6. Aktor meminta sistem untuk menyimpan data penitipan yang telah diinputkan
7. Sistem mengecek data penitipan hewan yang telah diinputkan
  - E-1 Data penitipan hewan yang diinputkan aktor salah
8. Sistem menyimpan data penitipan hewan ke database

9. Use Case selesai

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

A-1 Aktor memilih untuk melakukan entry data data penitipan hewan lama

1. Berlanjut ke Basic Flow langkah ke 4

A-2 Aktor memilih untuk melakukan display data penitipan hewan

1. Sistem menampilkan data penitipan hewan
2. Berlanjut ke Basic Flow langkah ke 9

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

E-1 Data penitipan hewan yang diinputkan aktor salah

1. Sistem memberikan peringatan bahwa data penitipan hewan yang diinputkan salah
2. Kembali ke Basic Flow langkah ke 4

#### **7. PreConditions**

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

#### **8. PostConditions**

1. Data penitipan hewan telah tersimpan atau tertampil

### **4.1.9 Use Case Spesification : Pengelolaan Data Grooming**

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

Use Case ini digunakan oleh aktor untuk mengelola data grooming hewan seperti entry data dan display data.

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

1. Dokter

### 3. Supporting Actor

none

### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data grooming
2. Sistem memberikan pilihan untuk entry data grooming hewan baru, entry data grooming hewan lama, atau menampilkan data grooming
3. Aktor memilih untuk entry data grooming hewan baru
  - A-1 Aktor memilih untuk entry data grooming hewan lama
  - A-2 Aktor memilih untuk menampilkan data grooming
4. Sistem menampilkan form untuk pengisian data grooming
5. Aktor menambahkan data grooming
6. Aktor meminta sistem untuk menyimpan data ke dalam database
  - E-1 Data grooming yang diinputkan salah
7. Sistem menyimpan data grooming ke dalam database
8. Use Case selesai

### 5. Alternative Flow

- A-1 Aktor memilih untuk melakukan entry data grooming hewan lama
1. Sistem meminta Id hewan yang akan ditambahkan data groomingnya
  2. Aktor menginputkan Id hewan yang akan ditambahkan data groomingnya

3. Sistem melakukan pengecekan Id ke dalam database

E-2 Id tidak valid

4. Aktor menambahkan data grooming hewan lama

5. Aktor meminta sistem untuk menyimpan data grooming yang telah ditambahkan

6. Sistem melakukan pengecekan terhadap data grooming yang telah diedit

E-3 Data grooming yang telah ditambahkan salah

7. Sistem menyimpan data grooming yang telah ditambahkan ke database

8. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan display data grooming

1. Sistem menampilkan seluruh data grooming hewan

2. Berlanjut ke Basic Flow langkah ke 8

## 9. Error Flow

E-1 Data grooming yang diinputkan salah

1. Sistem memberi peringatan bahwa data yang diinputkan salah

2. Kembali ke Basic Flow langkah 5

E-2 Id tidak valid

1. Sistem memberikan pesan peringatan bahwa Id yang diinputkan tidak ada dalam database

2. Kembali ke Alternative Flow A-1 langkah 2

E-3 Data grooming yang ditambahkan salah

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah

2. Kembali ke Alternative Flow A-1 Langkah ke 4

#### **6. PreConditions**

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

#### **7. PostConditions**

1. Data grooming hewan telah masuk ke database

#### **4.1.10 Use Case Spesification : Pengelolaan Surat Keterangan Sehat**

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

Use Case ini digunakan oleh aktor untuk mengelola surat keterangan sehat bagi hewan.

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

1. Dokter

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

None

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan surat keterangan sehat bagi hewan
2. Sistem memberikan pilihan untuk melakukan cetak surat keterangan sehat
3. Aktor memilih untuk melakukan cetak data surat keterangan sehat
4. Sistem menampilkan form pengisian data
5. Aktor menginputkan data surat keterangan sehat
6. Aktor meminta sistem untuk mencetak data surat keterangan sehat yang telah diinputkan

7. Sistem mencetak data surat keterangan sehat yang telah diinputkan

8. Use Case selesai

**5. Alternative Flow**

none

**6. Error Flow**

none

**7. PreConditions**

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

**8. PostConditions**

1. Surat keterangan sehat dicetak

**4.1.11 Use Case Spesification : Pendaftaran Forum**

**1. Brief Description**

Use Case ini digunakan oleh aktor untuk mendaftar account di forum.

**2. Primary Actor**

1. Pengunjung

**3. Supporting Actor**

none

**4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pendaftaran account
2. Aktor memilih untuk mendaftar account
3. Sistem menampilkan form pendaftaran account
4. Aktor menginputkan detail account
5. Aktor meminta sistem untuk menyimpan detail account yang telah diinputkan ke dalam database

- E-1 Data account yang diinputkan salah
6. Sistem menyimpan account yang diinputkan aktor ke database
  7. Use case selesai

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

none

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

E-1 Data account yang diinputkan salah

1. Sistem memberi peringatan bahwa data yang diinputkan salah
2. Kembali ke Basic Flow langkah 4

#### **7. PreConditions**

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

#### **8. PostConditions**

2. Data account tersimpan di database

### **4.1.12 Use Case Spesification : Pengelolaan Forum**

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

Use Case ini digunakan oleh aktor untuk menambahkan thread atau memberikan tanggapan di forum.

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

1. Pengunjung

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk ikut berpartisipasi dalam forum
2. Aktor memilih untuk membuat thread baru

- A-1 Aktor memilih untuk memberikan reply atas thread tertentu
3. Sistem menampilkan form pembuatan thread
  4. Aktor menginputkan detail thread
  5. Aktor meminta sistem untuk menyimpan detail thread yang telah diinputkan ke dalam database
  6. Sistem menyimpan account detail thread yang diinputkan aktor ke database
  7. Use case selesai

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

A-1 Aktor memilih untuk memberikan reply atas thread tertentu

1. Aktor memilih salah satu judul thread
2. Aktor memilih untuk menambahkan reply atas thread tersebut
3. Sistem menampilkan form pembuatan thread reply
4. Aktor menginputkan detail thread reply
5. Aktor meminta sistem untuk menyimpan detail thread reply yang telah diinputkan ke dalam database
6. Sistem menyimpan account detail thread reply yang diinputkan aktor ke database
5. Kembali ke Basic Flow langkah 7

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

none

#### **7. PreConditions**

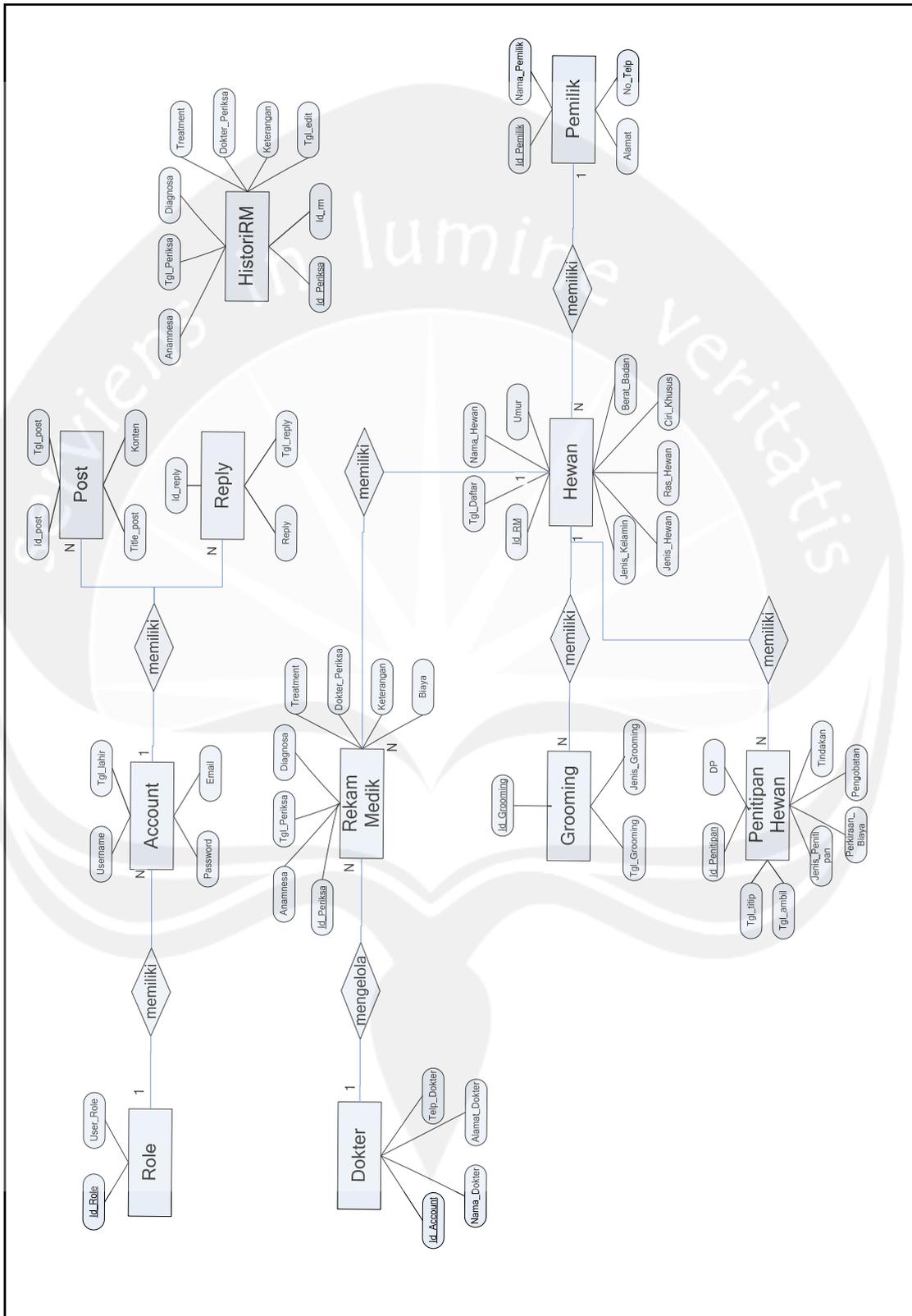
1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem

## 8. PostConditions

1. Data thread atau reply tersimpan di database



## Entity Relationship Diagram (ERD)



# DPPL

## DESKRIPSI PERANCANGAN PERANGKAT LUNAK

### Silico

(Sistem Informasi Klinik Hewan Calico)

Untuk :

Praktek Dokter Hewan dan Pet Shop Calico

Dipersiapkan oleh:

Duhita Anggayasti / 5017

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<b>DPPL-Silico</b>		1/86
		Revisi		

Program Studi Teknik Informatika	DPPL – Silico	171/ 86
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		

## 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 .....	9
1.1	Tujuan .....	9
1.2	Ruang Lingkup .....	9
1.3	Definisi, Akronim, dan Singkatan .....	9
1.4	Referensi .....	12
2	Perancangan Sistem .....	13
2.1	Perancangan Arsitektur .....	13
2.2	Perancangan Rinci .....	14
2.2.1	Sequence Diagram .....	14
2.2.2	Class Diagram .....	31
2.2.3	Deskripsi Kelas .....	32
3	Perancangan Data .....	49
3.1	Dekomposisi Data .....	49
3.1.1	Deskripsi Entitas Data Account .....	49
3.1.2	Deskripsi Entitas Data Role .....	49
3.1.3	Deskripsi Entitas Data Dokter .....	49
3.1.4	Deskripsi Entitas Data Rekam Medis .....	49
3.1.5	Deskripsi Entitas Data Hewan .....	50
3.1.6	Deskripsi Entitas Data Pemilik Hewan .....	50
3.1.7	Deskripsi Entitas Data Penitipan Hewan .....	50
3.1.8	Deskripsi Entitas Data Grooming .....	51
3.1.9	Deskripsi Entitas Histori Rekam Medis .....	51
3.1.10	Deskripsi Entitas Data Post .....	52
3.1.11	Deskripsi Entitas Data Reply .....	52
3.2	Physical Data Model .....	53
4	Perancangan Antarmuka .....	54
4.1	Login.....	54
4.2	Main Page Administrator.....	55
4.2.1	Pengelolaan Account Dokter.....	56
4.2.1.1	Entry Data Account Dokter.....	57
4.2.1.2	Edit Data Account Dokter.....	58
4.2.1.3	Delete Data Account Dokter.....	59
4.2.1.4	Display Data Account Dokter.....	60
4.2.2	Pengelolaan Account Customer Service.....	60
4.2.2.1	Entry Data Account Customer Service.....	61
4.2.2.2	Edit Data Account Customer Service.....	62
4.2.2.3	Delete Data Account Customer Service.....	63
4.2.2.4	Display Account Customer Service.....	64
4.2.3	Display Account Pengunjung.....	64
4.2.4	Delete Thread Post.....	65
4.3	Main Page Dokter.....	66
4.3.1	Pengelolaan data Rekam Medis Hewan.....	67
4.3.1.1	Entry Data Rekam Medis Hewan.....	68
4.3.1.2	Edit Data Rekam Medis Hewan.....	69
4.3.1.3	Display Data Rekam Medis Hewan.....	70
4.3.2	Display Surat Keterangan Sehat.....	71
4.4	Main Page Customer Service.....	72
4.4.1	Pengelolaan Pendaftaran Pasien Hewan.....	73
4.4.1.1	Entry Data Pasien Hewan.....	74
4.4.1.2	Edit Data Pasien Hewan.....	75
4.4.1.3	Display Data Pasien Hewan.....	76
4.4.2	Pengelolaan Data Penitipan Hewan.....	77
4.4.2.1	Entry Data Penitipan Hewan.....	78
4.4.2.2	Entry Data Penitipan Hewan Lama.....	79

4.4.2.3 Display Data Penitipan Hewan.....	80
4.4.3 Pengelolaan Data Grooming.....	81
4.4.3.1 Entry Data Grooming Baru.....	82
4.4.3.2 Entry Data Grooming Lama.....	83
4.4.3.3 Display Data Grooming.....	84
4.5 Pendaftaran user.....	85
4.6 Pengelolaan Form.....	85
4.6.1 Entry Thread Baru.....	85
4.6.2 Entry Thread Reply.....	86



## Daftar Gambar

Gambar 2.1 Rancangan Arsitektur Silico.....	13
Gambar 2.2 Sequence Diagram : Login.....	14
Gambar 2.3 Sequence Diagram : Pengelolaan Account Dokter - Entry Account Dokter.....	15
Gambar 2.4 Sequence Diagram : Pengelolaan Account Dokter - Edit Account Dokter.....	15
Gambar 2.5 Sequence Diagram : Pengelolaan Account Dokter - Delete Account Dokter.....	16
Gambar 2.6 Sequence Diagram : Pengelolaan Account Dokter - Display Account Dokter.....	16
Gambar 2.7 Sequence Diagram : Pengelolaan Account Customer Service - Entry Account Customer Service.....	17
Gambar 2.8 Sequence Diagram : Pengelolaan Account Customer Service - Edit Account Customer Service.....	18
Gambar 2.9 Sequence Diagram : Pengelolaan Account Customer Service - Delete Account Customer Service.....	19
Gambar 2.10 Sequence Diagram : Pengelolaan Account Customer Service - Display Account Customer Service.....	19
Gambar 2.11 Sequence Diagram : Pengelolaan Account Pengunjung..	20
Gambar 2.12 Sequence Diagram : Pengelolaan Forum - Delete Thread Post.....	20
Gambar 2.13 Sequence Diagram : Pengelolaan Pendaftaran Pasien Hewan - Entry Data Pasien Hewan.....	21
Gambar 2.14 Sequence Diagram : Pengelolaan Pendaftaran Pasien Hewan - Edit Data Pasien Hewan.....	22
Gambar 2.15 Sequence Diagram : Pengelolaan Pendaftaran Pasien Hewan - Display Data Pasien Hewan.....	23
Gambar 2.16 Sequence Diagram : Pengelolaan Data Rekam Medis Hewan - Entry Data Rekam Medis Hewan.....	24
Gambar 2.17 Sequence Diagram : Pengelolaan Data Rekam Medis Hewan - Edit Data Rekam Medis Hewan.....	25
Gambar 2.18 Sequence Diagram : Pengelolaan Data Rekam Medis Hewan - Display Data Rekam Medis Hewan.....	25
Gambar 2.19 Sequence Diagram : Pengelolaan Data Rekam Medis Hewan - Display Histori Rekam Medis Hewan .....	26
Gambar 2.20 Sequence Diagram : Pengelolaan Data Penitipan Hewan - Entry Data Penitipan Hewan Baru.....	26
Gambar 2.21 Sequence Diagram : Pengelolaan Data Penitipan Hewan - Entry Data Penitipan Hewan Lama.....	27
Gambar 2.22 Sequence Diagram : Pengelolaan Data Penitipan Hewan - Display Data Penitipan Hewan.....	27
Gambar 2.23 Sequence Diagram : Pengelolaan Data Grooming - Entry Data Grooming Baru.....	28

Gambar 2.24 Sequence Diagram : Pengelolaan Data Grooming - Entry Data Grooming Lama.....	28
Gambar 2.25 Sequence Diagram : Pengelolaan Data Grooming - Display Data Grooming.....	29
Gambar 2.26 Sequence Diagram : Cetak Surat Keterangan Sehat....	29
Gambar 2.27 Sequence Diagram : Pendaftaran User.....	30
Gambar 2.28 Sequence Diagram : Pengelolaan Forum - Entry Thread Baru.....	30
Gambar 2.29 Sequence Diagram : Pengelolaan Forum - Entry Thread Baru.....	31
Gambar 2.28 Class Diagram.....	32
Gambar 3.2 Physical Data Model.....	52
Gambar 4.1 Rancangan Antarmuka Login.....	54
Gambar 4.2 Rancangan Antarmuka Main Page Administrator - Pengelolaan Account.....	55
Gambar 4.3 Rancangan Antarmuka Main Page Administrator - Pengelolaan Account Dokter.....	56
Gambar 4.4 Rancangan Antarmuka Pengelolaan Account Dokter - Entry Data Account Dokter.....	57
Gambar 4.5 Rancangan Antarmuka Pengelolaan Account Dokter - Edit Data Account Dokter .....	58
Gambar 4.6 Rancangan Antarmuka Pengelolaan Account Dokter - Delete Data Account Dokter.....	59
Gambar 4.7 Rancangan Antarmuka Pengelolaan Account Dokter - Display Data Account Dokter.....	60
Gambar 4.8 Rancangan Antarmuka Main Page Administrator - Pengelolaan Account Customer Service.....	60
Gambar 4.9 Rancangan Antarmuka Pengelolaan Account Customer Service - Entry Data Account Customer Service.....	61
Gambar 4.10 Rancangan Antarmuka Pengelolaan Account Customer Service - Edit Data Account Customer Service.....	62
Gambar 4.11 Rancangan Antarmuka Pengelolaan Account Customer Service - Delete Data Account Customer Service.....	63
Gambar 4.12 Rancangan Antarmuka Pengelolaan Account Customer Service - Display Data Account Customer Service.....	64
Gambar 4.13 Rancangan Antarmuka Pengelolaan Account Pengunjung - Display Data Account Pengunjung.....	64
Gambar 4.14 Rancangan Antarmuka Pengelolaan Forum - Delete Thread Post.....	65
Gambar 4.15 Rancangan Antarmuka Main Page Dokter.....	66
Gambar 4.16 Rancangan Antarmuka Main Page Dokter - Pengelolaan rekam Medis Hewan.....	67
Gambar 4.17 Rancangan Antarmuka Pengelolaan Data Rekam Medis Hewan - Entry Data Rekam Medis Hewan.....	68
Gambar 4.18 Rancangan Antarmuka Pengelolaan Data Rekam Medis Hewan - Edit Data Rekam Medis Hewan.....	69
Gambar 4.19 Rancangan Antarmuka Pengelolaan Data Rekam Medis Hewan - Display Data Rekam Medis Hewan.....	70
Gambar 4.20 Rancangan Antarmuka Display Surat Keterangan Sehat.	71
Gambar 4.21 Rancangan Antarmuka Main Page Customer Service.....	72
Gambar 4.22 Rancangan Antarmuka Main Page Customer Service - Pengelolaan Pendaftaran Pasien Hewan.....	73
Gambar 4.23 Rancangan Antarmuka Pengelolaan Pendaftaran Pasien Hewan - Entry Data Pasien Hewan.....	74
Gambar 4.24 Rancangan Antarmuka Pengelolaan Pendaftaran Pasien Hewan - Edit Data Pasien Hewan.....	75

Gambar 4.25 Rancangan Antarmuka Pengelolaan Pendaftaran Pasien Hewan - Display Data Pasien Hewan.....	76
Gambar 4.26 Rancangan Antarmuka Main Page Customer Service - Pengelolaan Data Penitipan Hewan.....	77
Gambar 4.27 Rancangan Antarmuka Pengelolaan Data Penitipan Hewan - Entry Data Penitipan Hewan Baru.....	78
Gambar 4.28 Rancangan Antarmuka Pengelolaan Data Penitipan Hewan - Entry Data Penitipan Hewan lama.....	79
Gambar 4.29 Rancangan Antarmuka Pengelolaan Data Penitipan Hewan - Display Data Penitipan Hewan.....	80
Gambar 4.30 Rancangan Antarmuka Main Page Customer Service - Pengelolaan Data Grooming.....	81
Gambar 4.31 Rancangan Antarmuka Pengelolaan Data Grooming - Entry Data Grooming Baru.....	82
Gambar 4.32 Rancangan Antarmuka Pengelolaan Data Grooming - Entry Data Grooming Lama.....	83
Gambar 4.33 Rancangan Antarmuka Pengelolaan Data Grooming - Entry Data Grooming Lama.....	84
Gambar 4.34 Rancangan Antarmuka Pendaftaran User.....	84
Gambar 4.35 Rancangan Antarmuka Pengelolaan Forum - Entry Thread Baru.....	85
Gambar 4.36 Rancangan Antarmuka Pengelolaan Forum - Entry Thread Reply.....	86

## 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 Silico dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan data pendaftaran hewan.
2. Menangani pengelolaan data rekam medis hewan.
3. Menangani pengelolaan data penitipan hewan.
4. Menangani pengelolaan data grooming.
5. Menangani pengelolaan surat keterangan sehat.

Perangkat lunak Silico memiliki layanan Web-based Form yang dapat diakses melalui internet.

### 1.3 Definisi, Akronim dan Singkatan

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.

Silico	Perangkat lunak pengelolaan data Tugas Akhir mahasiswa.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
Client	Komputer yang meminta sumber daya dari server yang terhubung melalui jaringan.
Aplikasi Web	Perangkat lunak pelayanan dan pengolahan informasi pengguna berbasis Web.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Role	Hak yang dimiliki oleh user untuk menggunakan system/hak akses. Role diatur oleh administrator.
Database	Kumpulan data yang terkait yang diorganisasikan dalam struktur tertentu dan dapat diakses dengan cepat.
Administrator	Orang yang bertugas mengelola sistem secara keseluruhan serta membuat account bagi pengguna sistem tersebut.
Dokter	Hak akses dari administrator yang dapat digunakan oleh dokter hewan yang ada di klinik dimana dokter dapat menggunakan fungsi-fungsi pada sistem sesuai dengan hak aksesnya.

Customer Service	Hak akses dari administrator yang dapat digunakan oleh customer service di klinik hewan dimana customer service dapat menggunakan fungsi-fungsi pada sistem sesuai dengan hak aksesnya.
Pengunjung	Hak akses dari administrator yang dapat digunakan oleh pengunjung situs dimana pengunjung dapat menggunakan fungsi-fungsi pada sistem sesuai dengan hak aksesnya.
Grooming	Perawatan yang disediakan untuk penampilan fisik anjing (terutama bulu dan kuku), atau pekerjaan spesifik dilakukan untuk meningkatkan tampilan untuk pertunjukan.
Rekam Medis	Keterangan baik yang tertulis maupun terekam tentang identitas, anamnesa, penentuan fisik, laboratorium, diagnosa segala pelayanan dan tindakan medik yang diberikan kepada pasien dan pengobatan baik yang dirawat inap, rawat jalan, maupun yang mendapatkan pelayanan gawat darurat.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
Client	Komputer yang meminta sumber daya dari server yang terhubung melalui jaringan.
Aplikasi Web	Perangkat lunak pelayanan dan pengolahan informasi pengguna berbasis Web.

Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Forum	Fasilitas di internet dimana penggunaanya bisa berdiskusi.

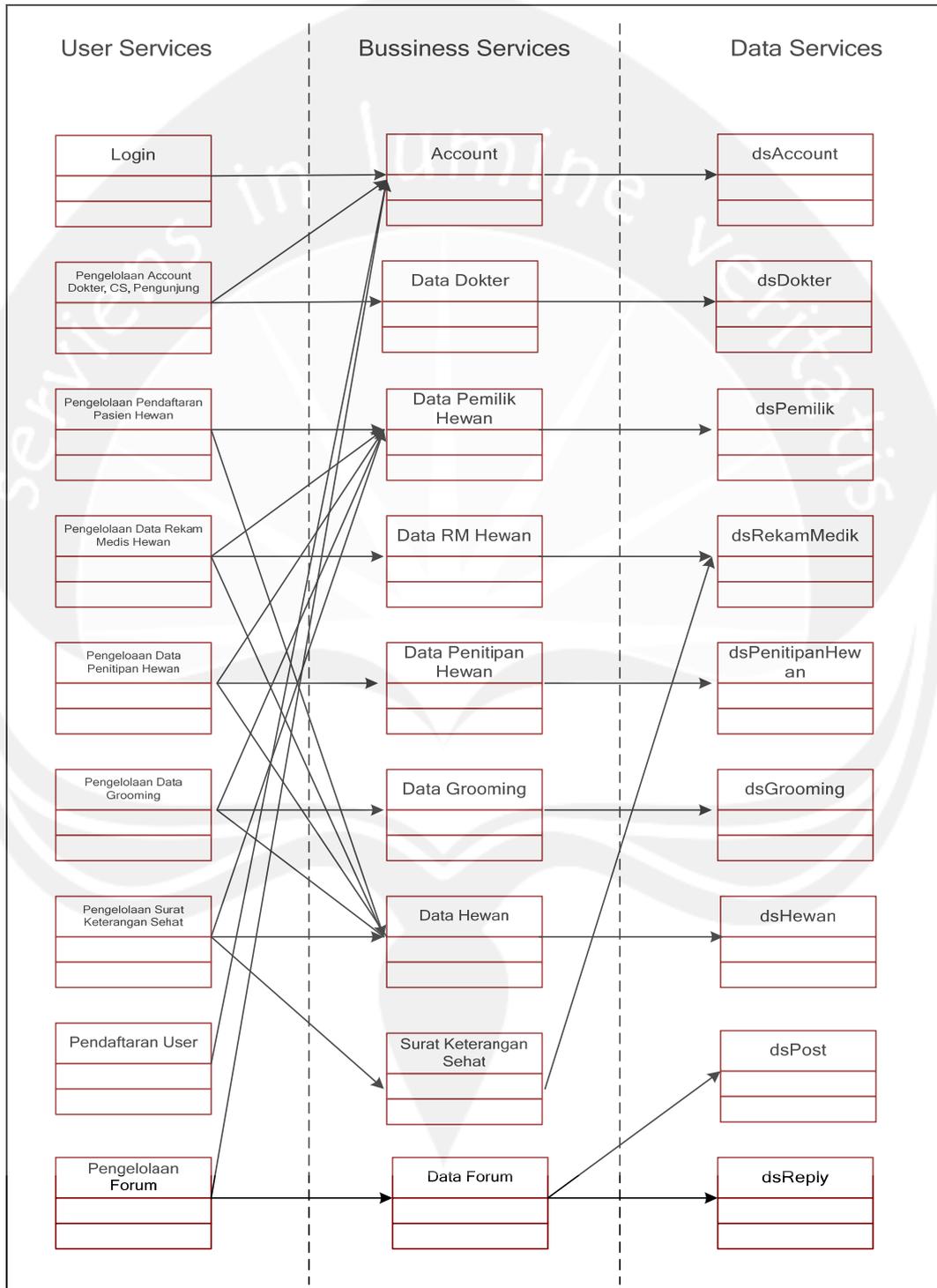
## Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
2. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.
3. MSDN Library-October 2005, Microsoft, 2005.
4. Anggayasti Duhita, *Spesifikasi Kebutuhan Perangkat Lunak Silico*, Universitas Atma Jaya Yogyakarta, 2009.

## 2. Perancangan Sistem

### 2.1. Perancangan Arsitektur

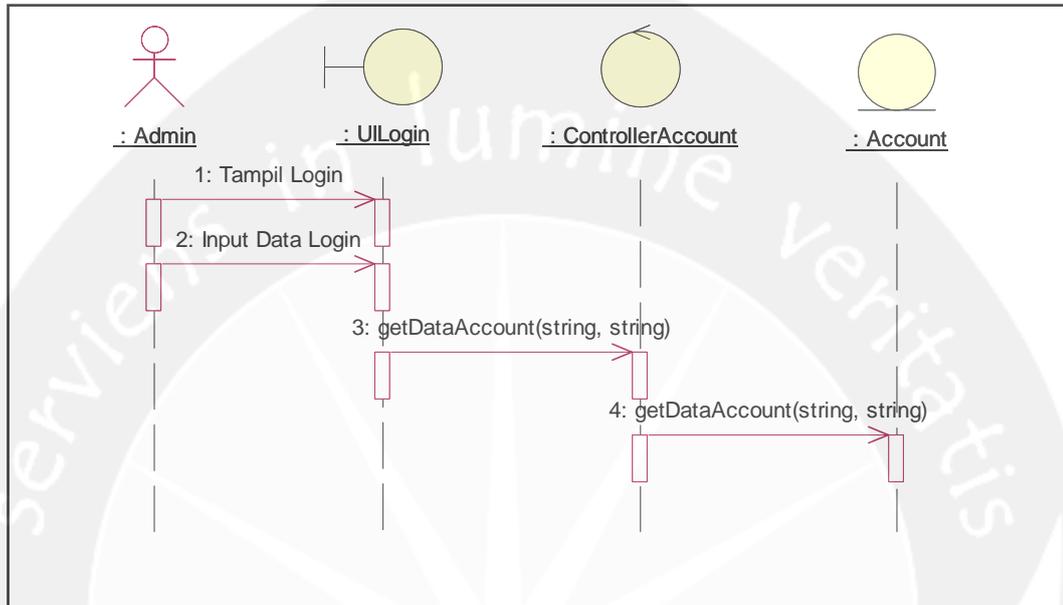


Gambar 2.1 Arsitektur Aplikasi Silico

## 2.2. Perancangan Rinci

### 2.2.1. Sequence Diagram

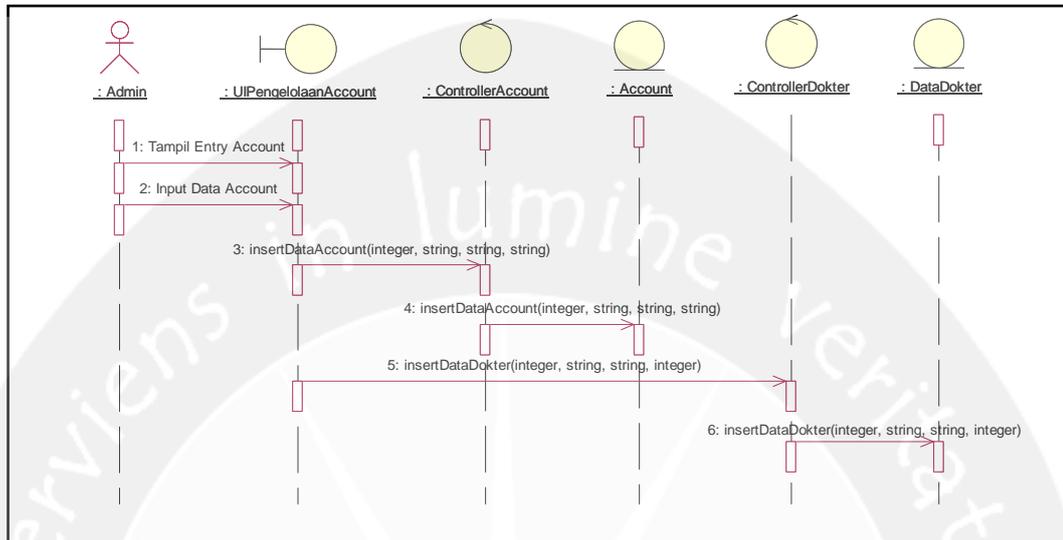
#### 2.2.1.1. Login



Gambar 2.2 Sequence Diagram - Login

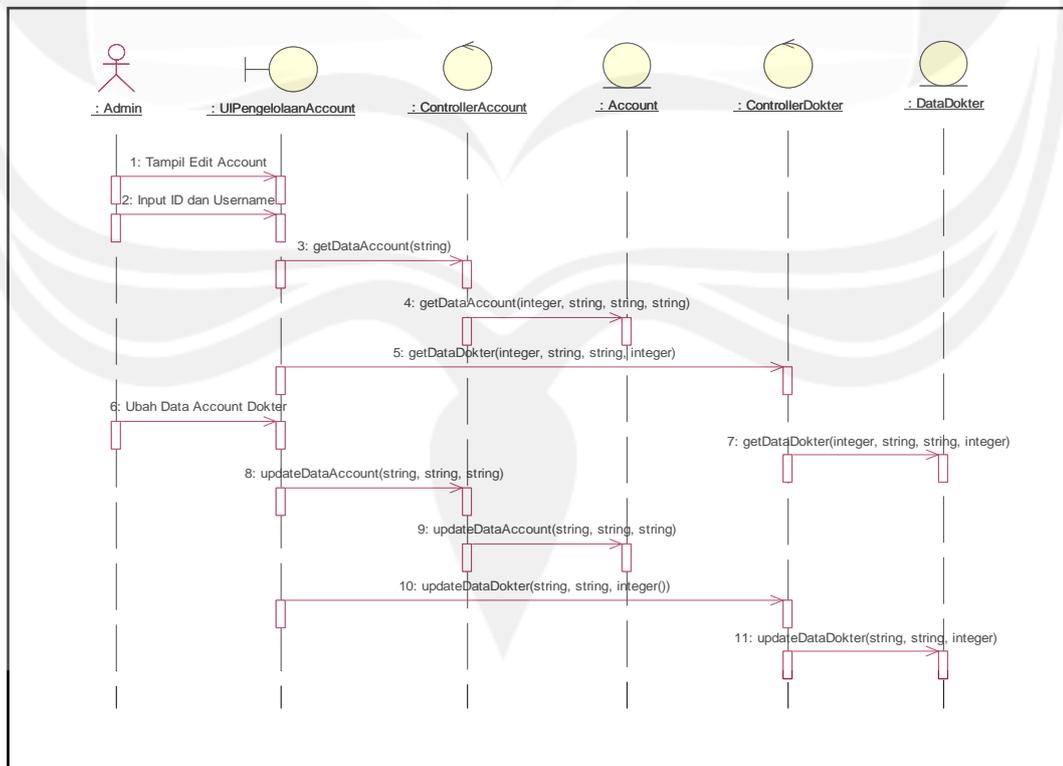
## 2.2.1.2. Pengelolaan Account Dokter

### 2.2.1.2.1. Entry Account Dokter



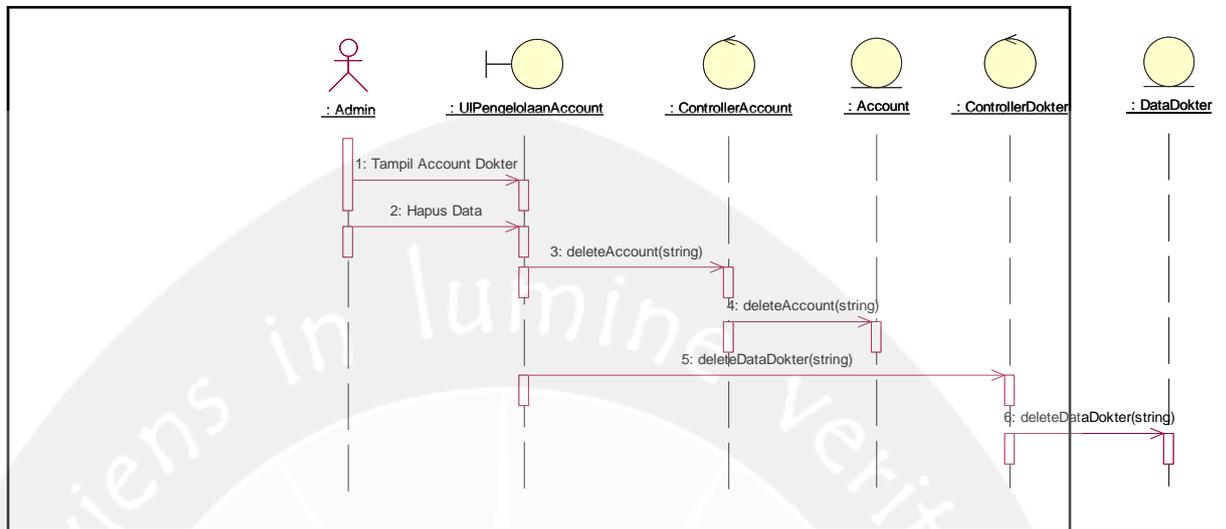
Gambar 2.3 Sequence Diagram : Pengelolaan Dokter - Entry Account Dokter

### 2.2.1.2.2. Edit Account Dokter



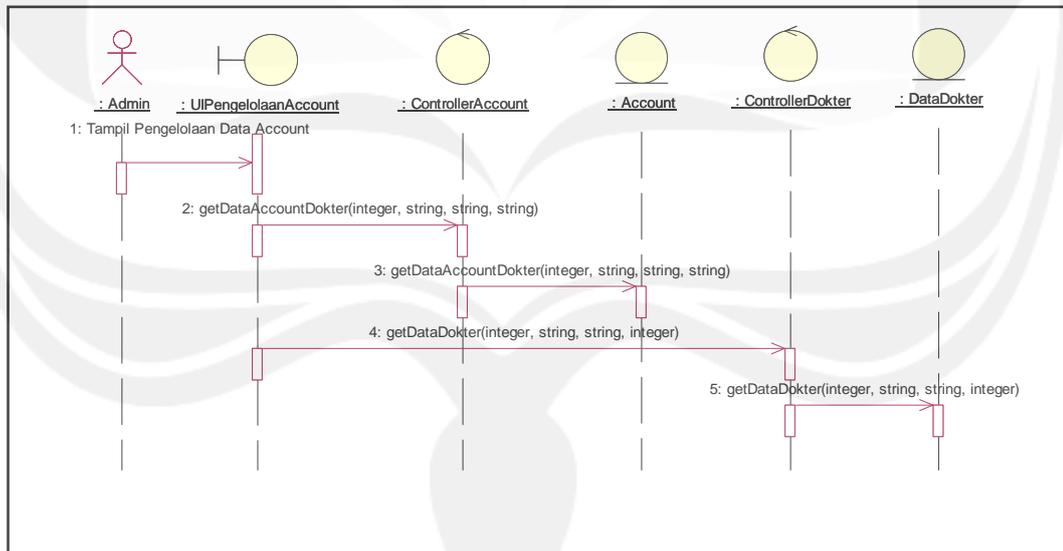
Gambar 2.4 Sequence Diagram : Pengelolaan Account Dokter - Edit Account Dokter

### 2.2.1.2.3. Delete Account Dokter



Gambar 2.5 Sequence Diagram : Pengelolaan Account Dokter - Delete Account Dokter

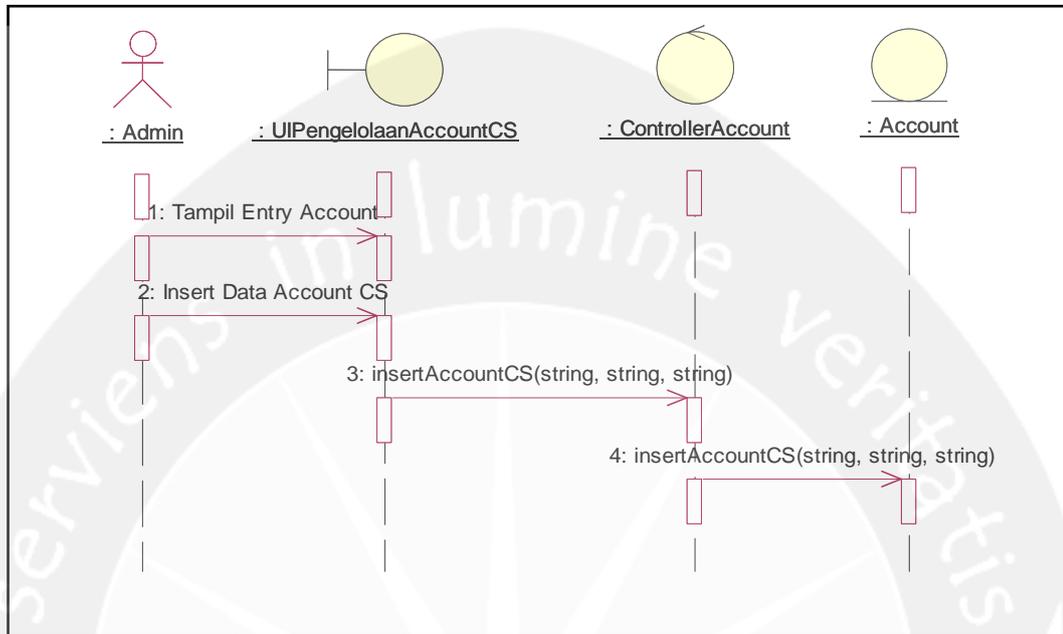
### 2.2.1.2.4. Display Account Dokter



Gambar 2.6 Sequence Diagram : Pengelolaan Account Dokter - Display Account Dokter

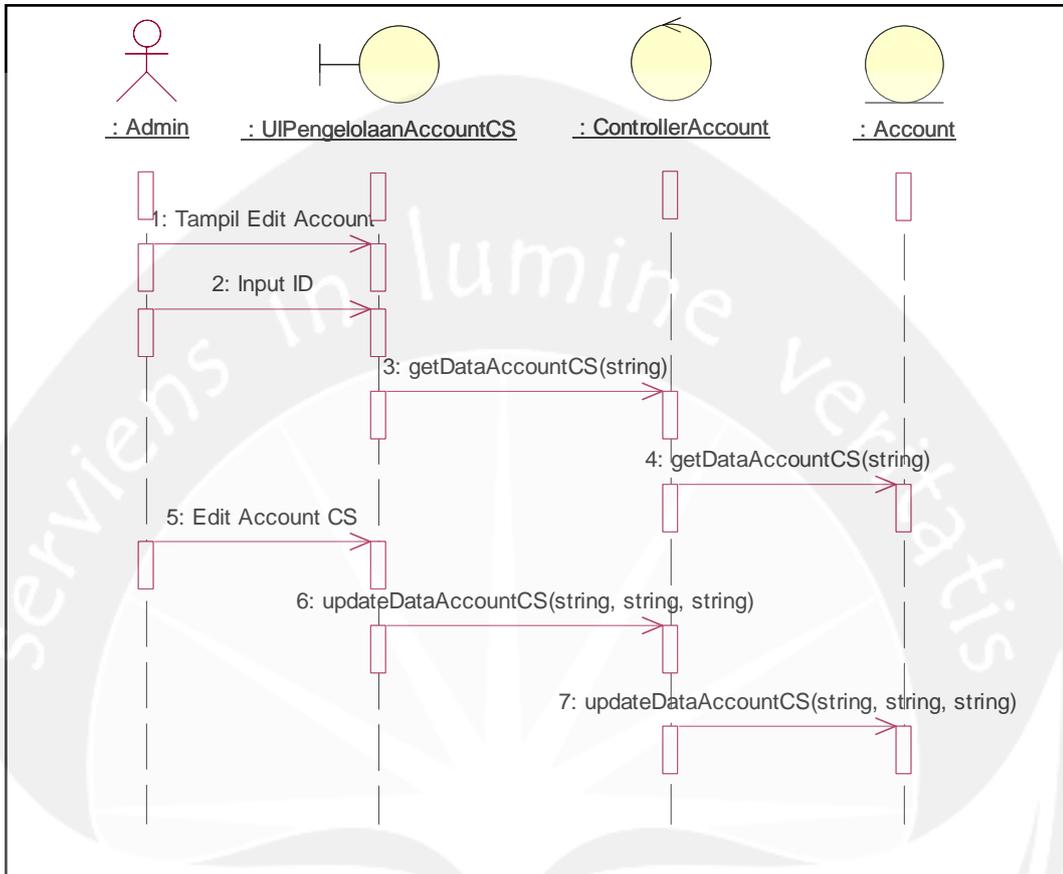
### 2.2.1.3. Pengelolaan Account Customer Service

#### 2.2.1.3.1. Entry Account Customer Service



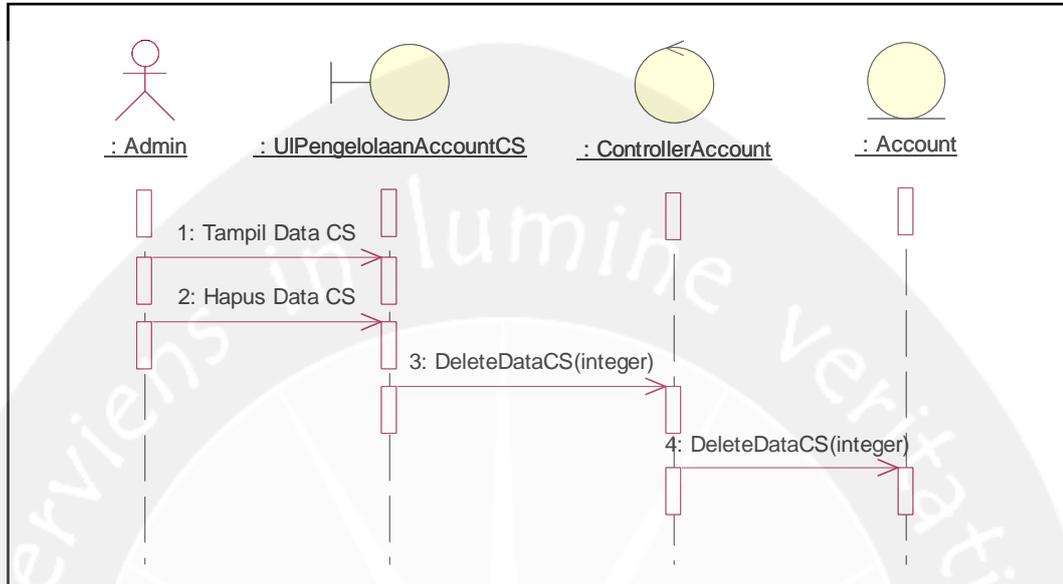
Gambar 2.7 Sequence Diagram : Pengelolaan Account Customer Service  
- Entry Account Customer Service

### 2.2.1.3.2. Edit Account Customer Service



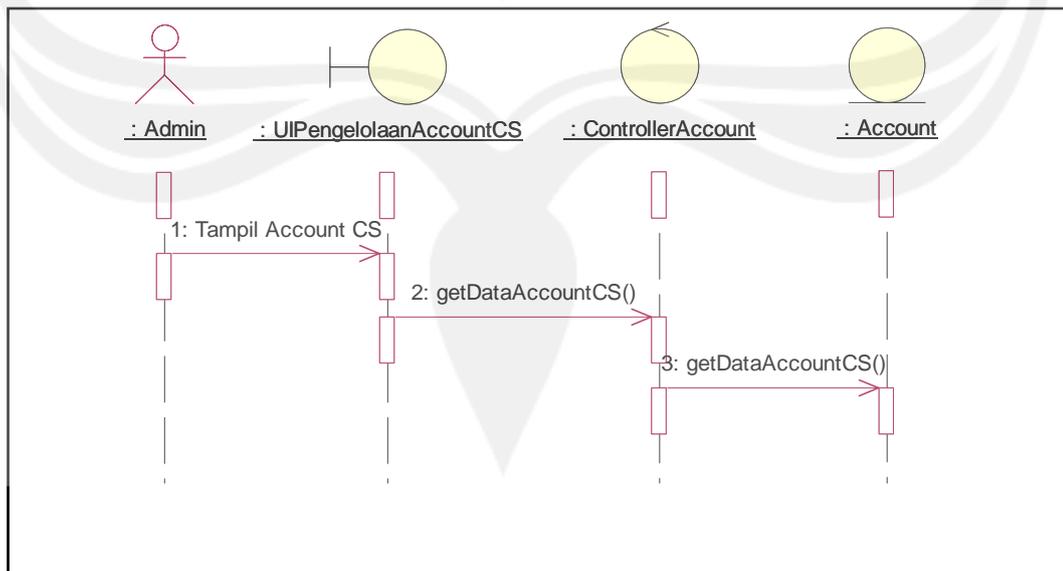
Gambar 2.8 Sequence Diagram : Pengelolaan Account Customer Service - Edit Account Customer Service

### 2.2.1.3.3. Delete Account Customer Service



Gambar 2.9 Sequence Diagram : Pengelolaan Account Customer Service - Delete Account Customer Service

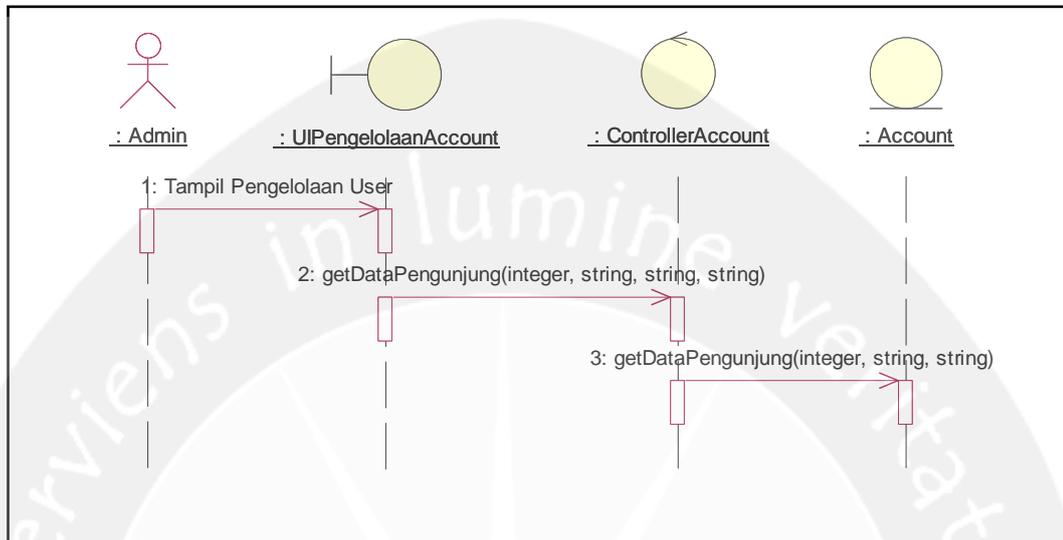
### 2.2.1.3.4. Display Account Customer Service



Gambar 2.10 Sequence Diagram : Pengelolaan Account Customer Service - Display Account Customer Service

## 2.2.1.4 Pengelolaan Account Pengunjung

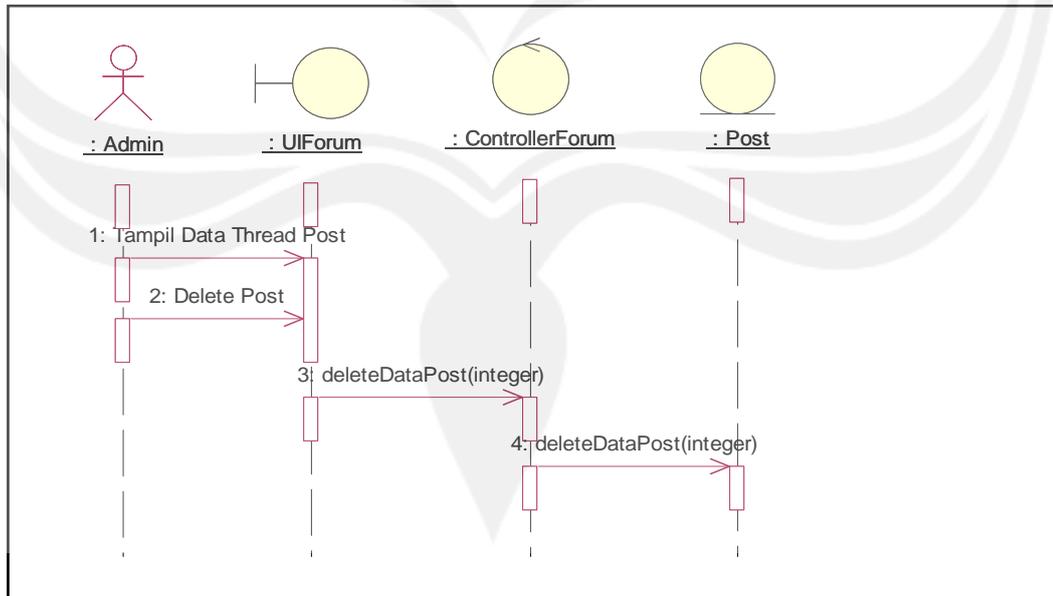
### 2.2.1.4.1. Display Account Pengunjung



Gambar 2.11 Sequence Diagram : Pengelolaan Account Pengunjung - Display Account Pengunjung

## 2.2.1.5 Pengelolaan Forum

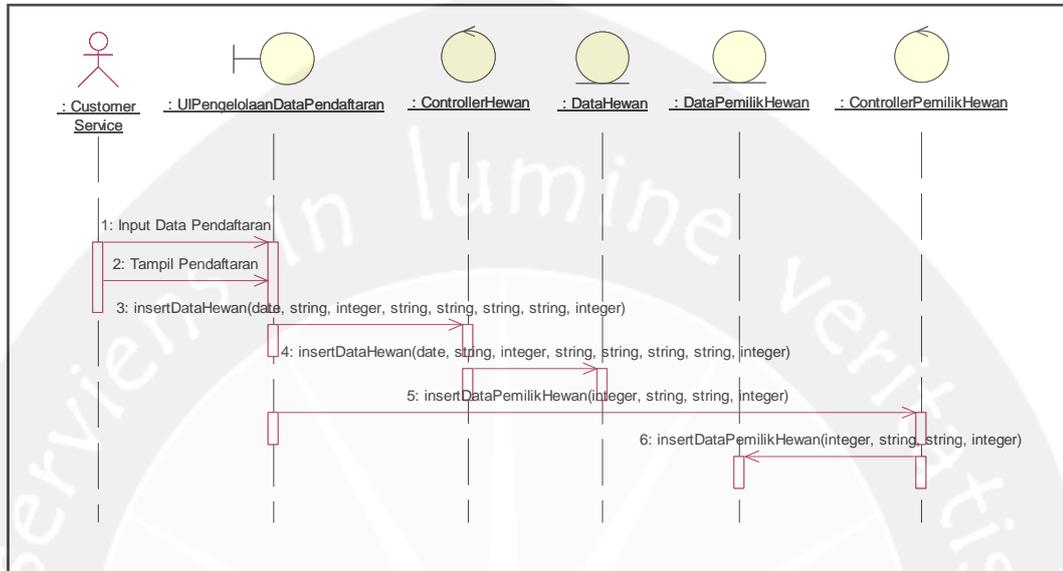
### 2.2.1.5.1. Delete Thread Post



Gambar 2.12 Sequence Diagram : Pengelolaan Forum - Delete Thread Post

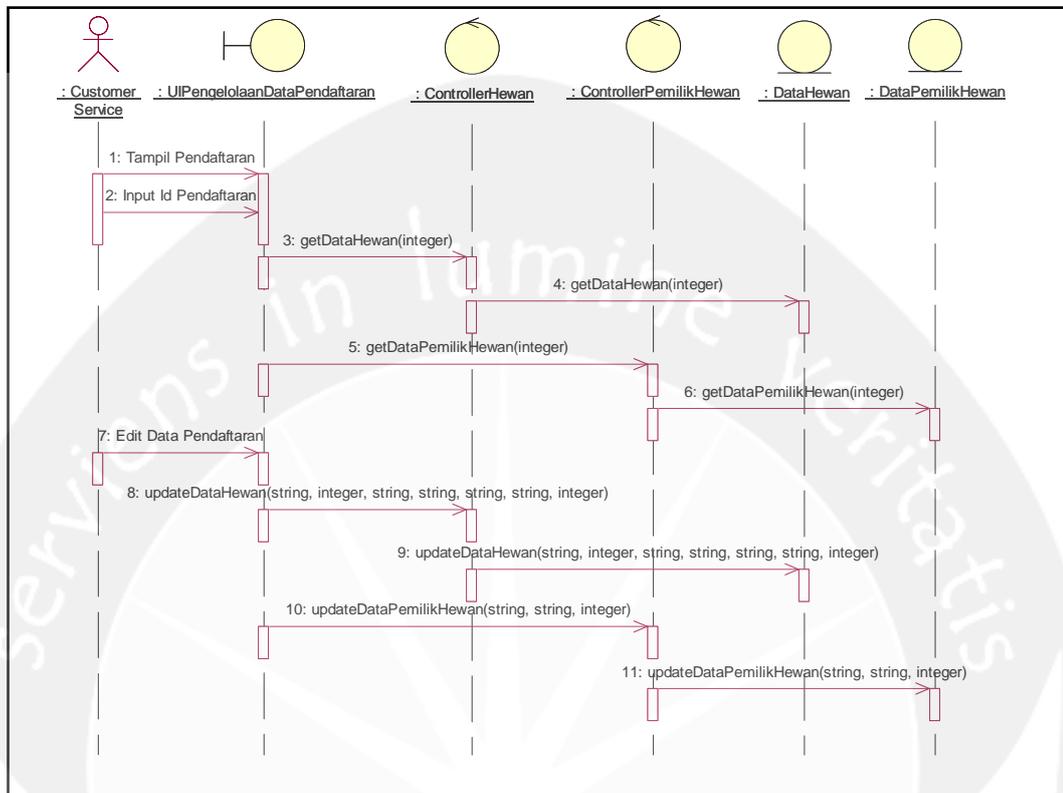
## 2.2.1.6. Pengelolaan Pendaftaran Pasien Hewan

### 2.2.1.6.1. Entry Data Pasien Hewan



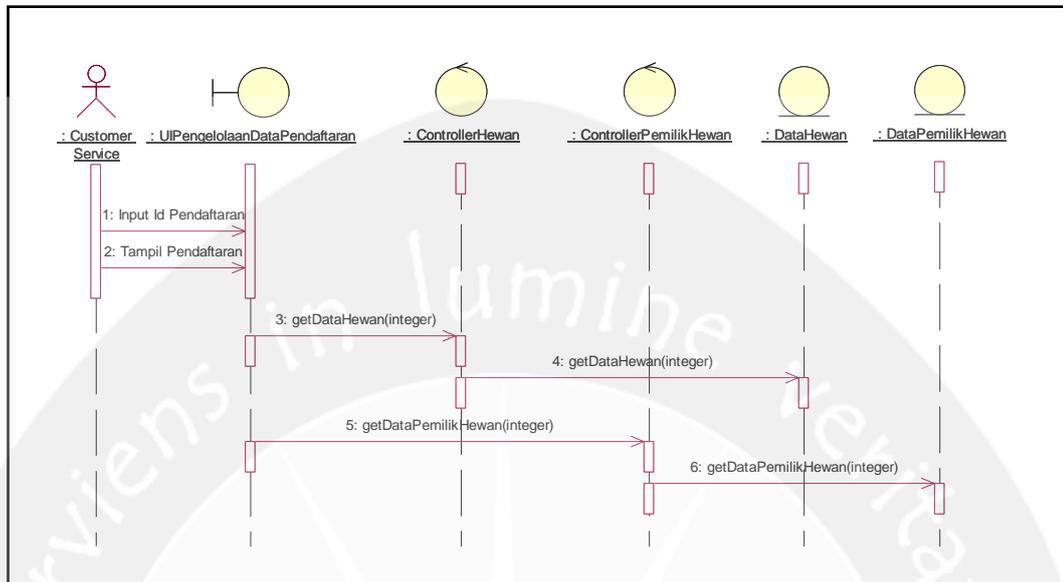
Gambar 2.13 Sequence Diagram : Pengelolaan Pendaftaran Pasien Hewan - Entry Data Pasien Hewan

### 2.2.1.6.2. Edit Data Pasien Hewan



Gambar 2.14 Sequence Diagram : Pengelolaan Pendaftaran Pasien Hewan - Edit Data Pasien Hewan

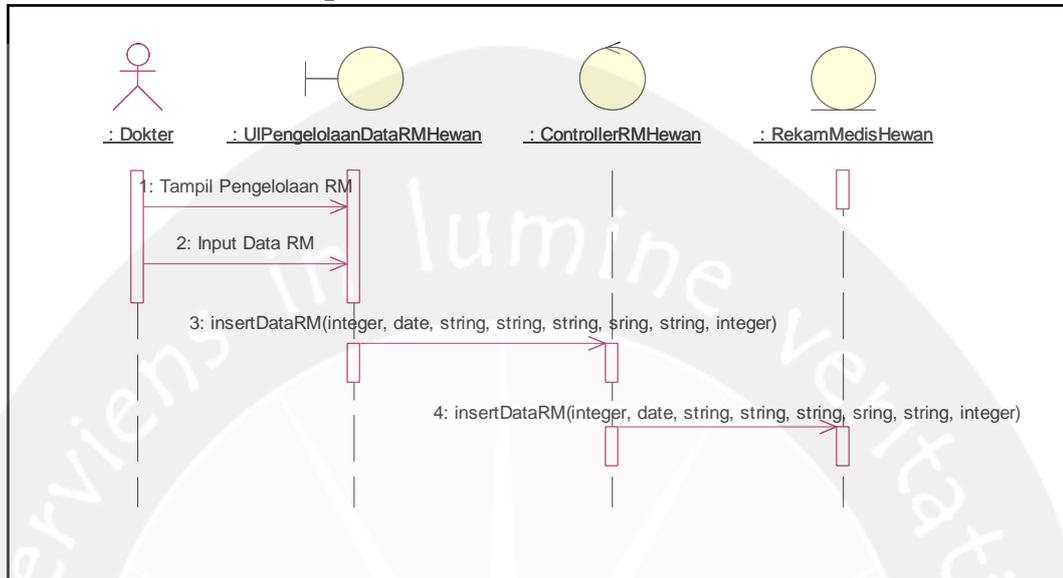
### 2.2.1.6.3. Display Data Pasien Hewan



Gambar 2.15 Sequence Diagram : Pengelolaan Pendaftaran Pasien Hewan - Display Data Pasien Hewan

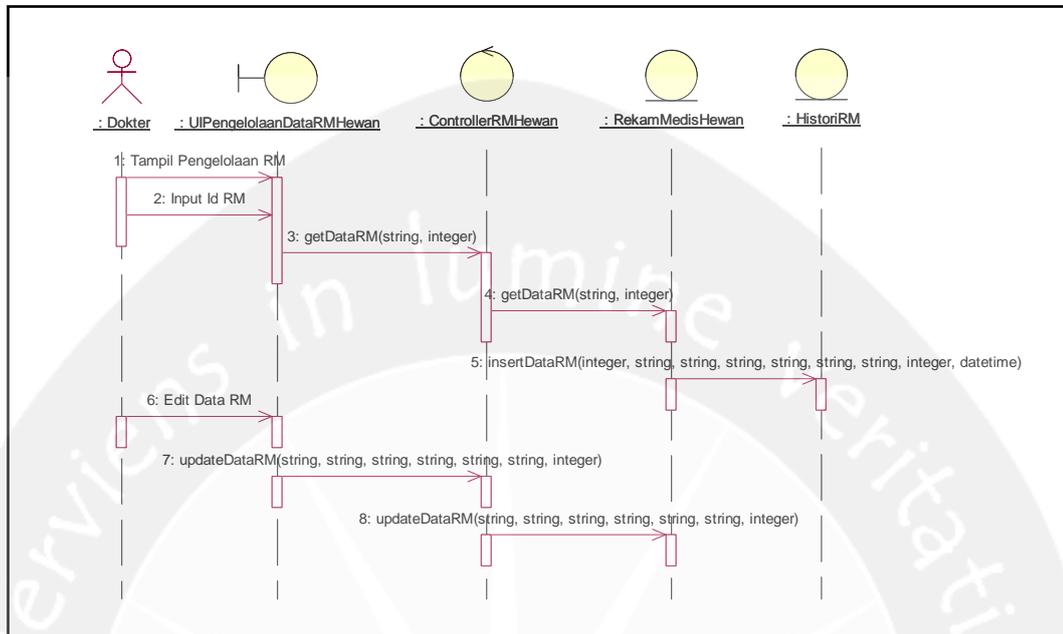
## 2.2.1.7. Pengelolaan Data Rekam Medis Hewan

### 2.2.1.7.1. Entry Data Rekam Medis Hewan



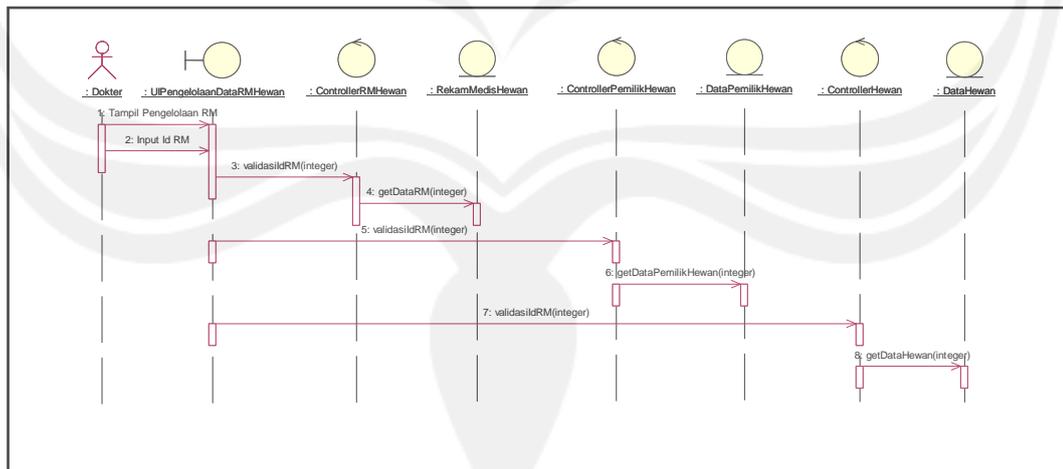
Gambar 2.16 Sequence Diagram : Pengelolaan Data Rekam Medis Hewan - Entry Data Rekam Medis Hewan

### 2.2.1.7.2. Edit Data Rekam Medis Hewan



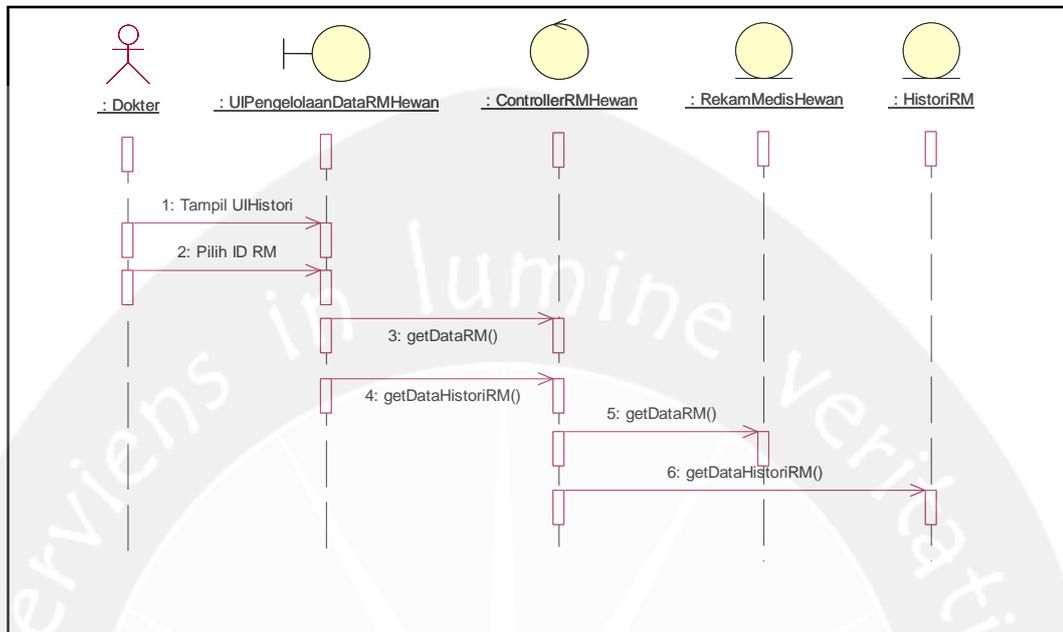
Gambar 2.17 Sequence Diagram : Pengelolaan Data Rekam Medis Hewan - Edit Data Rekam Medis Hewan

### 2.2.1.7.3. Display Data Rekam Medis Hewan



Gambar 2.18 Sequence Diagram : Pengelolaan Data Rekam Medis Hewan - Display Data Rekam Medis Hewan

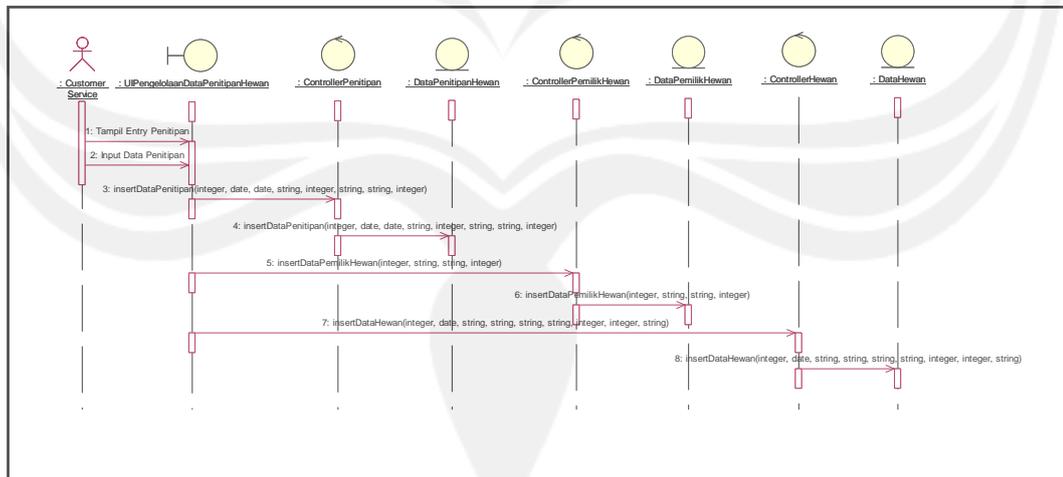
#### 2.2.1.7.4. Display Histori Rekam Medis Hewan



Gambar 2.19 Sequence Diagram : Pengelolaan Data Rekam Medis Hewan - Display Histori Rekam Medis Hewan

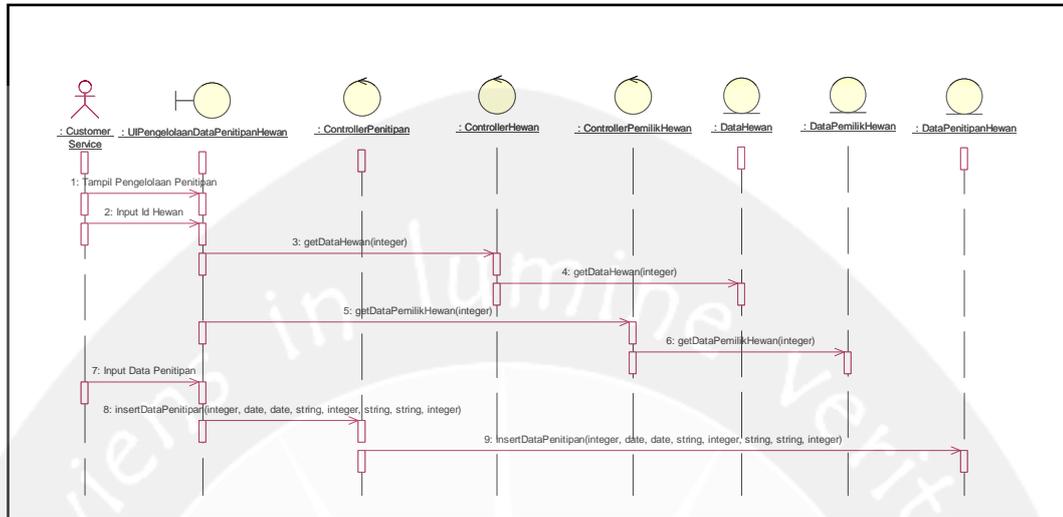
#### 2.2.1.8. Pengelolaan Data Penitipan Hewan

##### 2.2.1.8.1 Entry Data Penitipan Hewan Baru



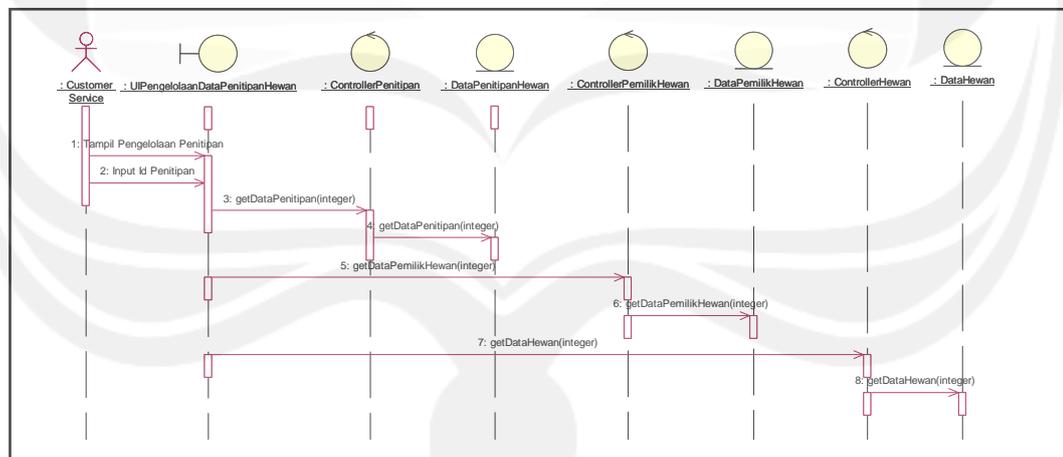
Gambar 2.20 Sequence Diagram : Pengelolaan Data Penitipan Hewan - Entry Data Penitipan Hewan Baru

### 2.2.1.8.2 Entry Data Penitipan Hewan Lama



Gambar 2.21 Sequence Diagram : Pengelolaan Data Penitipan Hewan - Entry Data Penitipan Hewan Lama

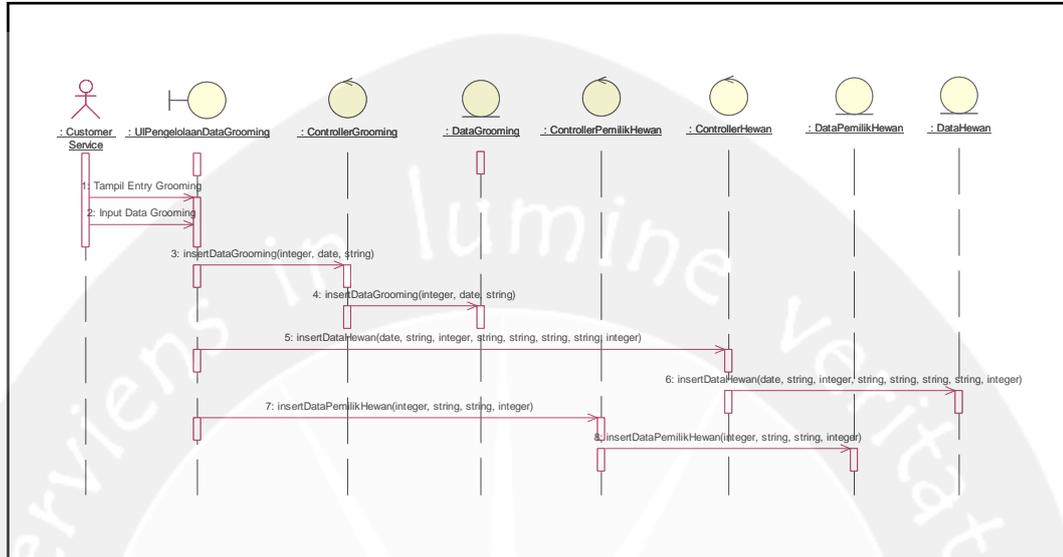
### 2.2.1.8.3 Display Data Penitipan Hewan



Gambar 2.22 Sequence Diagram : Pengelolaan Data Penitipan Hewan - Display Data Penitipan Hewan

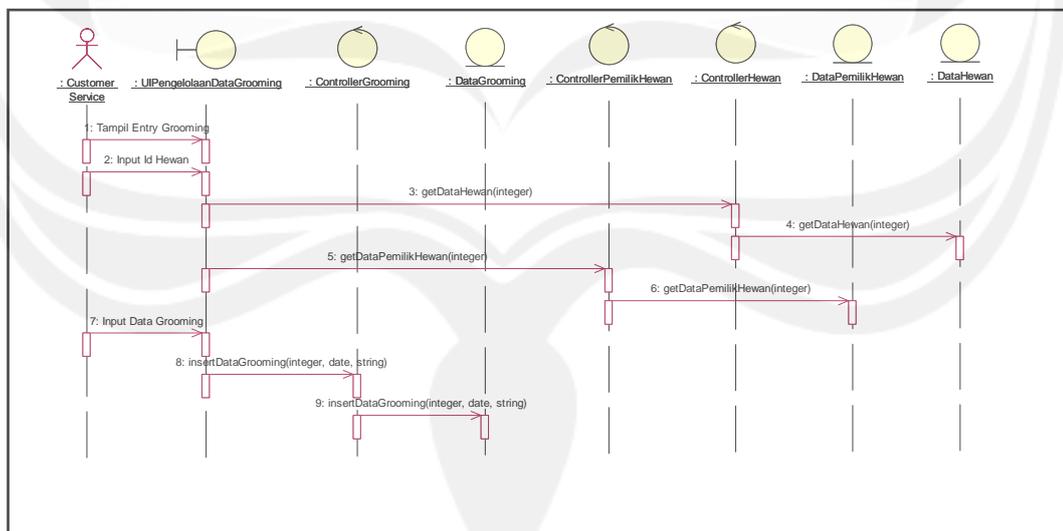
## 2.2.1.9 Pengelolaan Data Grooming

### 2.2.1.9.1 Entry Data Grooming Baru



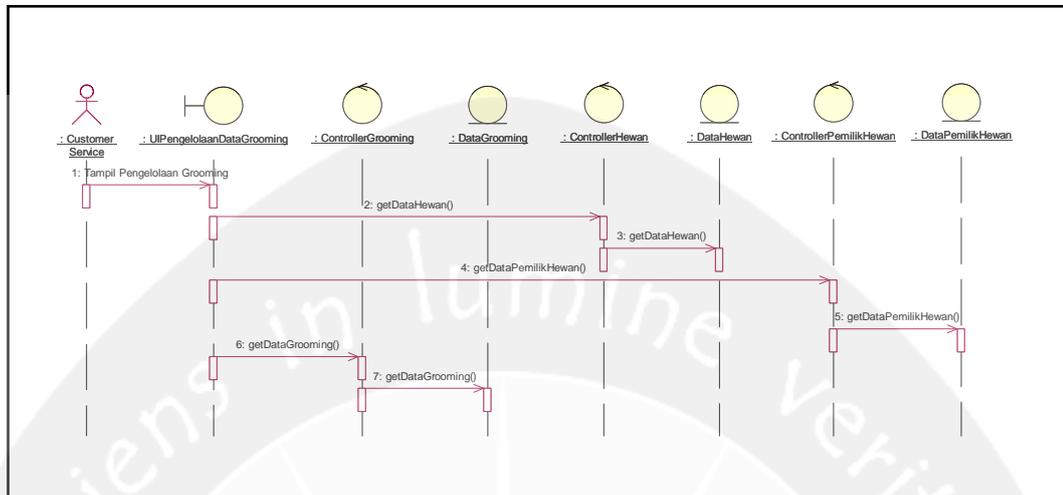
Gambar 2.23 Sequence Diagram : Pengelolaan Data Grooming - Entry Data Grooming Baru

### 2.2.1.9.2 Entry Data Grooming Lama



Gambar 2.24 Sequence Diagram : Pengelolaan Data Grooming - Entry Data Grooming Lama

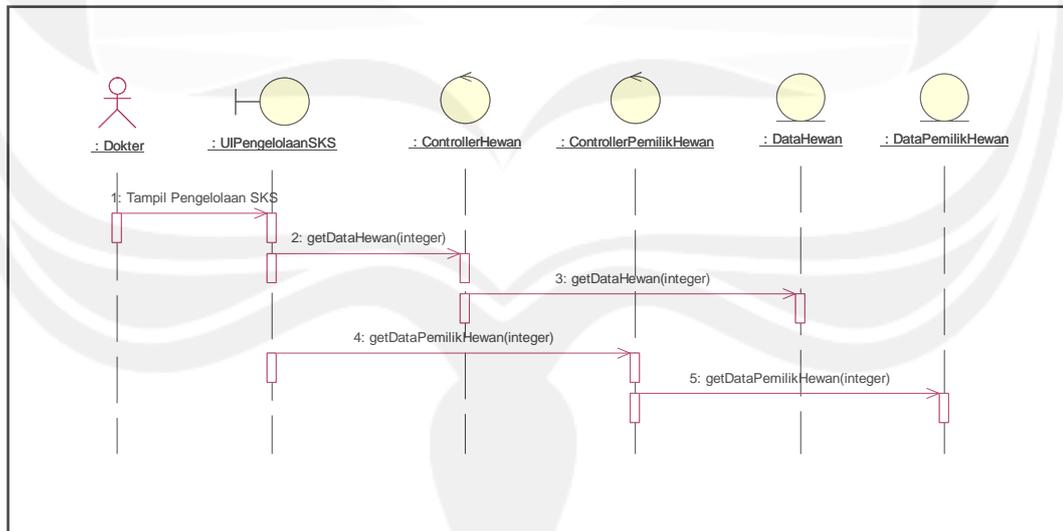
### 2.2.1.9.3 Display Data Grooming



Gambar 2.25 Sequence Diagram : Pengelolaan Data Grooming - Display Data Grooming

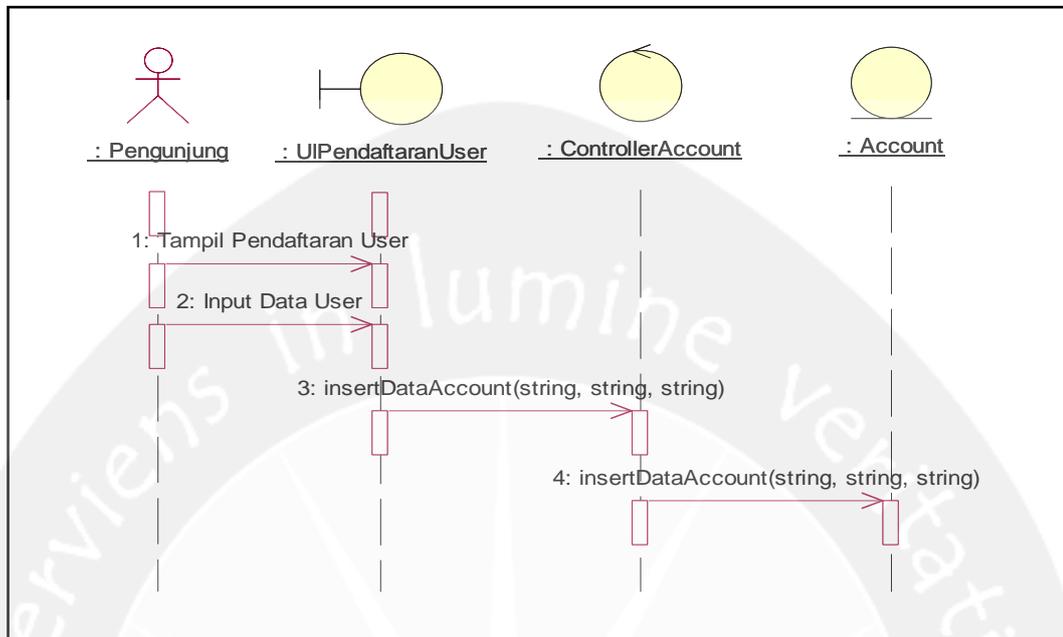
### 2.2.1.10 Pengelolaan Surat Keterangan Sehat

#### 2.2.1.10.1 Cetak Surat Keterangan Sehat



Gambar 2.26 Sequence Diagram : Pengelolaan Surat Keterangan Sehat - Cetak Surat Keterangan Sehat

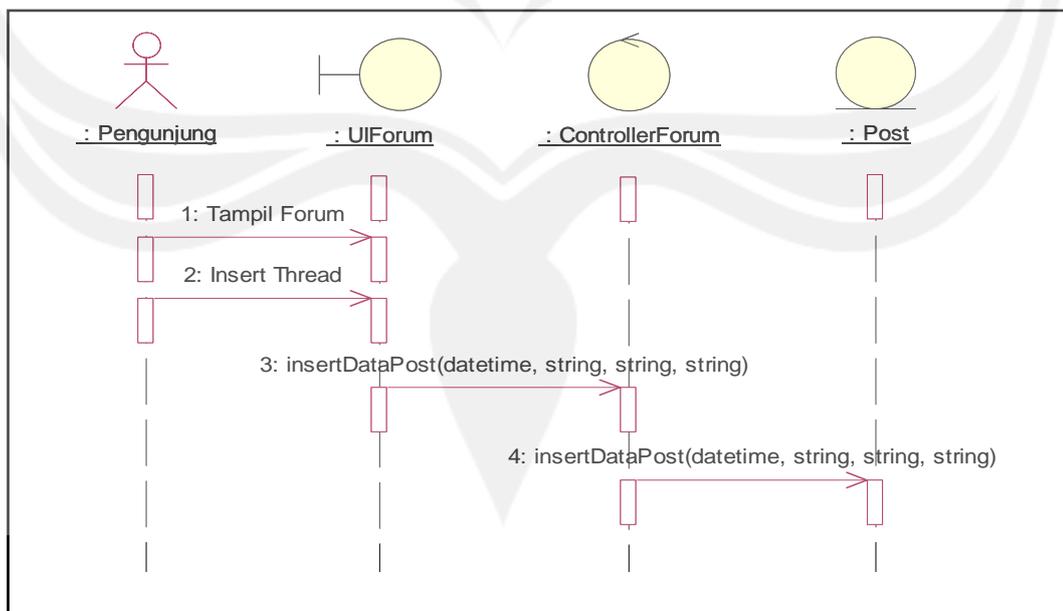
### 2.2.1.11 Pendaftaran User



Gambar 2.27 Sequence Diagram : Pendaftaran User

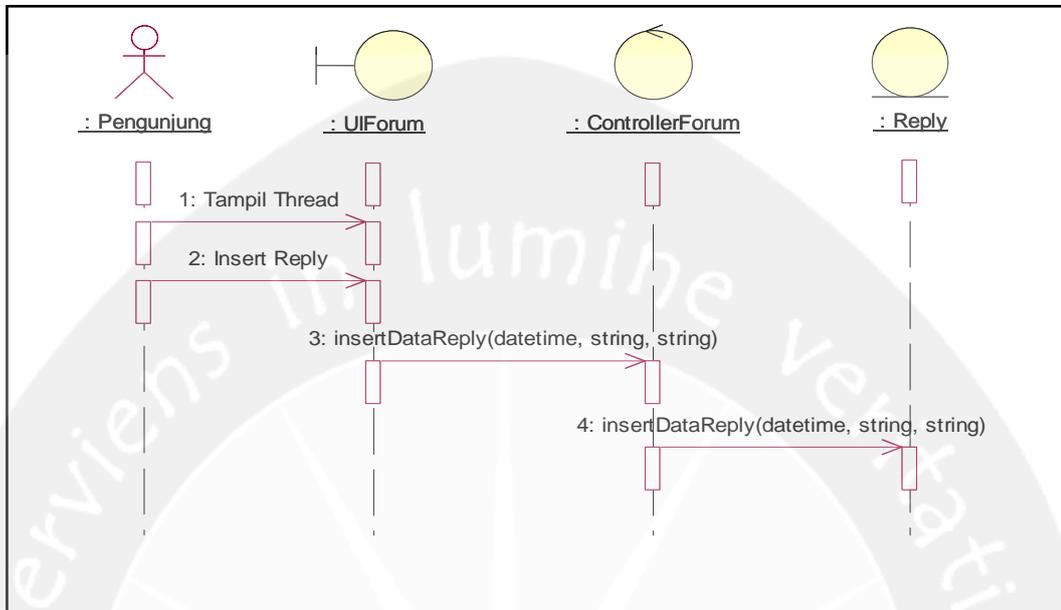
### 2.2.1.12 Pengelolaan Forum

#### 2.2.1.12.1 Entry Thread Baru



Gambar 2.28 Sequence Diagram : Pengelolaan Forum - Entry Thread Baru

### 2.2.1.12.2 Entry Thread Reply



Gambar 2.29 Sequence Diagram : Pengelolaan Forum - Entry Thread Reply



## 2.2.3 Class Diagram Specific Descriptions

### 2.2.3.1 Specific Design Class UILogin

<b>UILogin</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<p>- ControllerAccount () Merepresentasikan objek control class dari Pengelolaan Account.</p>	
<p>+ UILogin () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

### 2.2.3.2 Specific Design Class UIPengelolaanAccount

<b>UIPengelolaanAccount</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<p>- ControllerAccount() Merepresentasikan objek control class dari Pengelolaan Account.</p>	
<p>- ControllerDokter () Merepresentasikan objek control class dari Data Dokter.</p>	
<p>+UIPengelolaanAccount() Default konstruktor, digunakan untuk inisialisasi semua attribute.</p>	

### 2.2.3.3 Specific Design Class UIPengelolaanDataPendaftaran

<b>UIPengelolaanDataPendaftaran</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<p>- ControllerHewan () Merepresentasikan objek control class dari Data Hewan.</p>	
<p>- ControllerPemilikHewan () Merepresentasikan objek control class dari Data Pemilik Hewan.</p>	

<pre>+UIPengelolaanDataPendaftaran()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>
---

#### 2.2.3.4 Specific Design Class UIPengelolaanDataRMHewan

<b>UIPengelolaanDataRMHewan</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>- ControllerRMHewan ()</pre> <p>Merepresentasikan objek control class dari Data RM Hewan.</p> <pre>- ControllerHewan ()</pre> <p>Merepresentasikan objek control class dari Data Hewan.</p> <pre>- ControllerPemilikHewan ()</pre> <p>Merepresentasikan objek control class dari Data Pemilik Hewan.</p>	
<pre>+UIPengelolaanDataRMHewan()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

#### 2.2.3.5 Specific Design Class UIPengelolaanDataPenitipanHewan

<b>UIPengelolaanDataPenitipanHewan</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>- ControllerHewan ()</pre> <p>Merepresentasikan objek control class dari Data Hewan.</p> <pre>- ControllerPemilikHewan ()</pre> <p>Merepresentasikan objek control class dari Data Pemilik Hewan.</p> <pre>- ControllerPenitipan ()</pre> <p>Merepresentasikan objek control class dari Data Penitipan Hewan.</p>	
<pre>+UIPengelolaanDataPenitipanHewan()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

### 2.2.3.6 Specific Design Class UIPengelolaanDataGrooming

UIPengelolaanDataGrooming	<<boundary>>
<p>- ControllerHewan () Merepresentasikan objek control class dari Data Hewan.</p> <p>- ControllerPemilikHewan () Merepresentasikan objek control class dari Data Pemilik Hewan.</p> <p>- ControllerGrooming () Merepresentasikan objek control class dari Data Grooming.</p>	
<p>+UIPengelolaanDataGrooming() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

### 2.2.3.7 Specific Design Class UIPengelolaanSuratKeteranganSehat

UIPengelolaanSuratKeteranganSehat	<<boundary>>
<p>- ControllerHewan () Merepresentasikan objek control class dari Data Hewan.</p> <p>- ControllerPemilikHewan () Merepresentasikan objek control class dari Data Pemilik Hewan.</p>	
<p>+UIPengelolaanSuratKeteranganSehat() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

### 2.2.3.8 Specific Design Class UIPendaftaranUser

UIPendaftaranUser	<<boundary>>
<p>- ControllerAccount () Merepresentasikan objek control class dari Pendaftaran User.</p>	

+UIPendaftaranUser()

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

### 2.2.3.9 Specific Design Class UIPengelolaanAccountCS

UIPendaftaranUser	<<boundary>>
- ControllerAccount () Merepresentasikan objek control class dari Pengelolaan Account CS.	
+UIPengelolaanAccountCS() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	

### 2.2.3.10 Specific Design Class UIPengelolaanForum

UIPendaftaranUser	<<boundary>>
- ControllerForum () Merepresentasikan objek control class dari Pendaftaran User.	
+UIPengelolaanForum() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	

### 2.2.3.11 Specific Design Class ControllerAccount

ControllerAccount	<<control>>
+ insertDataAccount(id_account : integer, username : string, password : string, email : string) Operasi ini digunakan untuk memasukkan data account ke dalam database.	
+ getDataAccount(username : string, password : string) Operasi ini digunakan untuk mengambil data account berdasarkan username dan password dari database.	
+ updateDataAccount(username : string, password : string,	

```
email : string)
```

Operasi ini digunakan untuk mengubah (update) data account di database.

```
+ deleteAccount(id_account : integer, username : string, password : string, email : string)
```

Operasi ini digunakan untuk menghapus data account di database.

### 2.2.3.12 Specific Design Class ControllerRMHewan

<b>ControllerRMHewan</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ insertDataRM(id_periksa : integer, tgl_periksa : date, anamnesa : string, diagnosa : string, treatment : string, dokter_periksa : string, keterangan : string, biaya : integer)</pre>	
Operasi ini digunakan untuk memasukkan data rekam medik hewan ke dalam database.	
<pre>+ updateDataRM(tgl_periksa : date, anamnesa : string, diagnosa : string, treatment : string, dokter_periksa : string, keterangan : string, biaya : string)</pre>	
Operasi ini digunakan untuk mengubah (update) data rekam medis hewan di database.	
<pre>+ getDataRM()</pre>	
Operasi ini digunakan untuk mengambil data rekam medis hewan berdasarkan id rekam medis dari database,	

### 2.2.3.13 Specific Design Class ControllerHewan

<b>ControllerHewan</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ getDataHewan(id_hewan : string)</pre>	
Operasi ini digunakan untuk mengambil data hewan dari database.	

```
+ insertDataHewan(id_hewan : string, tgl_daftar : date,
jenis_hewan : string, ras_hewan : string, nama_hewan :
string, jenis_kelamin : string, umur : integer, berat_badan
: integer, ciri_khusus : string)
```

Operasi ini digunakan untuk memasukkan data hewan ke dalam database.

```
+ updateDataHewan(jenis_hewan : string, umur : integer,
ras_hewan : string, nama_hewan : string, jenis_kelamin :
string, ciri_khusus : string, berat_badan : integer)
```

Operasi ini digunakan untuk mengubah (update) data hewan di database.

#### 2.2.3.14 Specific Design Class ControllerPemilikHewan

<b>ControllerPemilikHewan</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ getDataPemilikHewan(id_pemilik : integer)</pre> <p>Operasi ini digunakan untuk mengambil data pemilik hewan dari database.</p> <pre>+ insertDataPemilikHewan(id_pemilik : integer, nama_pemilik : string, alamat : string, no_telp : integer)</pre> <p>Operasi ini digunakan untuk memasukkan data pemilik hewan ke dalam database.</p> <pre>+ updateDataPemilikHewan(nama_pemilik : string, alamat : string, no_telp : integer)</pre> <p>Operasi ini digunakan untuk mengubah (update) data pemilik hewan di database.</p>	

#### 2.2.3.15 Specific Design Class ControllerDokter

<b>ControllerDokter</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ insertDataDokter(id_dokter : integer, nama_dokter :</pre>	

```
string, alamat_dokter : string, telp_dokter : integer)
Operasi ini digunakan untuk memasukkan data dokter ke
database.
+ getDataDokter(id_dokter : integer, nama_dokter : string,
alamat_dokter : string, telp_dokter : integer)
Operasi ini digunakan untuk mengambil data dokter dari
database.
+ updateDataDokter(id_dokter : integer, nama_dokter :
string, alamat_dokter : string, telp_dokter : integer)
Operasi ini digunakan untuk mengubah (update) data dokter di
database.
```

### 2.2.3.16 Specific Design Class ControllerPenitipan

<b>ControllerPenitipan</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ insertDataPenitipan(id_penitipan : integer, tgl_titip : datetime, tgl_ambil : datetime, sinyalemen : string, jenis_penitipan : string, perkiraan_biaya : integer, pengobatan : string, tindakan : string, dp : integer) Operasi ini digunakan untuk memasukkan data penitipan hewan ke database. + getDataPenitipan() Operasi ini digunakan untuk mengambil data penitipan hewan dari database.</pre>	

### 2.2.3.17 Specific Design Class ControllerGrooming

<b>ControllerGrooming</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ insertDataGrooming(id_grooming : integer, tgl_grooming : date, jenis_grooming : string) Operasi ini digunakan untuk memasukkan data grooming ke</pre>	

database.

+ getDataGrooming()

Operasi ini digunakan untuk mengambil data grooming dari database.

### 2.2.3.18 Specific Design Class ControllerForum

<b>ControllerGrooming</b>	<b>&lt;&lt;control&gt;&gt;</b>
+ getDataPost() Operasi ini digunakan untuk mengambil data thread dari database.	
+ getDataReply() Operasi ini digunakan untuk mengambil data thread reply dari database.	
+ insertDataPost() Operasi ini digunakan untuk memasukkan data thread ke dalam database.	
+ insertDataReply() Operasi ini digunakan untuk memasukkan data thread reply ke dalam database.	

### 2.2.3.19 Specific Design Class Account

<b>Account</b>	<b>&lt;&lt;entity&gt;&gt;</b>
- Id_Account : integer Merepresentasikan id dari account.	
- Username : String Merepresentasikan nama account.	
- Password : String Merepresentasikan password account.	
- Email : String Merepresentasikan email account.	
- Tgl_lahir : Date Merepresentasikan tanggal lahir pembuat account	

<p>- Id_Role : Integer</p> <p>Merepresentasikan id_role dari account.</p>
<p>+ insertDataAccount(id_account : integer, username : string, password : string, email : string, tgl_lahir : date, id_role : integer)</p> <p>Operasi ini digunakan untuk memasukkan data account ke database.</p> <p>+ getDataLogin(username : string, password : string)</p> <p>Operasi ini digunakan untuk mengambil data account dari database.</p> <p>+ updateDataAccount(username : string, password : string, email : string)</p> <p>Operasi ini digunakan untuk mengubah (update) data account di database.</p> <p>+ deleteAccount(id_account : integer, username : string, password : string, email : string, id_role : integer)</p> <p>Operasi ini digunakan untuk menghapus data account dari database.</p> <p>+ getDataAccountPengunjung(id_role : integer)</p> <p>Operasi ini digunakan untuk mengambil data pengunjung dari database.</p>

### 2.2.3.20 Specific Design Class RekamMedisHewan

<b>RekamMedisHewan</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>- Id_periksa : integer</p> <p>Merepresentasikan id dari rekam medis hewan.</p> <p>- tgl_periksa : Date</p> <p>Merepresentasikan tanggal pemeriksaan hewan.</p> <p>- anamnesa : String</p> <p>Merepresentasikan anamnesa hewan.</p> <p>- diagnosa: String</p> <p>Merepresentasikan diagnosa terhadap hewan.</p> <p>- treatment : String</p> <p>Merepresentasikan treatment yang diperoleh hewan.</p>	

<p>- dokter_periksa : String Merepresentasikan nama dokter yang memeriksa hewan.</p> <p>- keterangan : String Merepresentasikan data keterangan periksa lainnya.</p> <p>- biaya : integer Merepresentasikan biaya pemeriksaan hewan.</p>
<p>+ insertDataRM(id_periksa : integer, tgl_periksa : date, anamnesa : string, diagnosa : string, treatment : string, dokter_periksa : string, keterangan : string, biaya : integer)</p> <p>Operasi ini digunakan untuk memasukkan data rekam medis hewan ke database.</p> <p>+ updateDataRM(tgl_periksa : date, anamnesa : string, diagnosa : string, treatment : string, dokter_periksa : string, keterangan : string, biaya : string)</p> <p>Operasi ini digunakan untuk mengubah (update) data rekam medis hewan di database.</p> <p>+ getDataRM()</p> <p>Operasi ini digunakan untuk mengambil data rekam medis hewan dari database.</p>

### 2.2.3.21 Specific Design Class DataHewan

<b>DataHewan</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>- id_rm : string Merepresentasikan id dari hewan.</p> <p>- tgl_daftar : Date Merepresentasikan tanggal pendaftaran hewan di klinik.</p> <p>- jenis_hewan : String Merepresentasikan jenis hewan.</p> <p>- ras_hewan : String Merepresentasikan ras dari hewan.</p>	

<p>- nama_hewan : String</p> <p>Merepresentasikan nama hewan.</p> <p>- jenis_kelamin : String</p> <p>Merepresentasikan jenis kelamin hewan.</p> <p>- umur : Integer</p> <p>Merepresentasikan umur hewan.</p> <p>- berat_badan : Integer</p> <p>Merepresentasikan berat badan hewan.</p> <p>- ciri_khusus : String</p> <p>Merepresentasikan ciri khusus yang dimiliki hewan.</p>
<p>+ getDataHewan(id_rm : string)</p> <p>Operasi ini digunakan untuk mengambil data hewan dari database.</p> <p>+ insertDataHewan(id_rm : string, tgl_daftar : date, jenis_hewan : string, ras_hewan : string, nama_hewan : string, jenis_kelamin : string, umur : integer, berat_badan : integer, ciri_khusus : string)</p> <p>Operasi ini digunakan untuk memasukkan data hewan ke database.</p> <p>+ updateDataHewan(jenis_hewan : string, umur : integer, ras_hewan : string, nama_hewan : string, jenis_kelamin : string, ciri_khusus : string, berat_badan : integer)</p> <p>Operasi ini digunakan untuk mengubah (update) data hewan di database.</p>

### 2.2.3.22 Specific Design Class DataPemilikHewan

<b>DataPemilikHewan</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>- id_pemilik : string</p> <p>Merepresentasikan id dari pemilik hewan.</p> <p>- nama_pemilik : String</p> <p>Merepresentasikan nama dari pemilik hewan.</p> <p>- alamat : String</p>	

Merepresentasikan alamat dari pemilik hewan.

- no\_telp : Integer

Merepresentasikan nomor telepon dari pemilik hewan.

+ getDataPemilikHewan(id\_pemilik : string)

Operasi ini digunakan untuk mengambil data pemilik hewan dari database.

+ insertDataPemilikHewan(id\_pemilik : string, nama\_pemilik : string, alamat : string, no\_telp : integer)

Operasi ini digunakan untuk memasukkan data pemilik hewan ke dalam database.

+ updateDataPemilikHewan(nama\_pemilik : string, alamat : string, no\_telp : integer)

Operasi ini digunakan untuk mengubah (update) data pemilik hewan di database.

### 2.2.3.23 Specific Design Class DataDokter

<b>DataDokter</b>	<b>&lt;&lt;entity&gt;&gt;</b>
- id_dokter : Integer Merepresentasikan id dari dokter.	
- nama_dokter : String Merepresentasikan nama dokter.	
- alamat_dokter : String Merepresentasikan alamat dokter.	
- telp_dokter : Integer Merepresentasikan nomor telepon dokter.	
+ insertDataDokter(id_dokter : string, nama_dokter : string, alamat_dokter : string, telp_dokter : integer) Operasi ini digunakan untuk memasukkan data dokter ke database.	
+ getDataDokter(id_dokter : integer, nama_dokter : string, alamat_dokter : string, telp_dokter : integer) Operasi ini digunakan untuk mengambil data dokter dari database.	

```
+ updateDataDokter(id_dokter : integer, nama_dokter :
string, alamat_dokter : string, telp_dokter : integer)
```

Operasi ini digunakan untuk mengubah (update) data dokter di database.

#### 2.2.3.24 Specific Design Class DataPenitipanHewan

DataPenitipanHewan	<<entity>>
<pre>- id_penitipan : Integer Merepresentasikan id dari penitipan hewan. - tgl_titip : Datetime Merepresentasikan tanggal dititipkannya hewan. - tgl_ambil : Datetime Merepresentasikan tanggal diambilnya hewan. - sinyalemen : String Merepresentasikan sinyalemen hewan. - jenis_penitipan : String Merepresentasikan jenis penitipan hewan. - perkiraan_biaya : Integer Merepresentasikan perkiraan biaya dari penitipan hewan. - pengobatan : String Merepresentasikan pengobatan yang dilakukan ketika hewan dititipkan. - tindakan : String Merepresentasikan tindakan yang dilakukan terhadap hewan selama dititipkan. - dp : Integer Merepresentasikan down payment (uang muka) dari penitipan hewan.</pre>	
<pre>+ insertDataPenitipan(id_penitipan : integer, tgl_titip : datetime, tgl_ambil : datetime, sinyalemen : string, jenis_penitipan : string, perkiraan_biaya : integer, pengobatan : string, tindakan : string, dp : integer)</pre>	

Operasi ini digunakan untuk memasukkan data penitipan hewan ke database.

```
+ getDataPenitipan(id_penitipan : integer)
```

Operasi ini digunakan untuk mengambil data penitipan hewan dari database.

### 2.2.3.25 Specific Design Class DataGrooming

DataGrooming	<<entity>>
- id_grooming: Integer Merepresentasikan id dari grooming.	
- tgl_grooming : Date Merepresentasikan tanggal dilakukannya grooming.	
- jenis_grooming : String Merepresentasikan jenis grooming yang dilakukan.	
+ insertDataGrooming(id_grooming : integer, tgl_grooming : date, jenis_grooming : string) Operasi ini digunakan untuk memasukkan data grooming ke database.	
+ getDataGrooming() Operasi ini digunakan untuk mengambil data grooming dari database.	

### 2.2.3.26 Specific Design Class Post

Post	<<entity>>
- id_post: Integer Merepresentasikan id dari thread post.	
- tgl_post : Date Merepresentasikan tanggal dibuatnya thread post.	
- Title_post : String Merepresentasikan judul thread post.	
- Konten : String	

Merepresentasikan isi thread post.

```
+ insertDataPost(id_post : integer, tgl_post : date,  
title_post : string, konten : string)
```

Operasi ini digunakan untuk memasukkan data thread post ke database.

```
+ getDataPost()
```

Operasi ini digunakan untuk mengambil data thread post dari database.

### 2.2.3.27 Specific Design Class Reply

<b>Reply</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>- id_reply : Integer Merepresentasikan id dari thread reply. - tgl_reply : Date Merepresentasikan tanggal dibuatnya thread reply. - reply : String Merepresentasikan isi thread reply.</pre>	
<pre>+ insertDataReply(id_reply : integer, tgl_reply : date, reply : string) Operasi ini digunakan untuk memasukkan data thread reply ke database. + getDataReply() Operasi ini digunakan untuk mengambil data thread reply dari database.</pre>	

### 2.2.3.28 Specific Design Class HistoriRM

<b>HistoriRM</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>- Id_periksa : integer Merepresentasikan id dari rekam medis hewan. - tgl_periksa : Date Merepresentasikan tanggal pemeriksaan hewan. - anamnesa : String Merepresentasikan anamnesa hewan.</pre>	

- diagnosa: String  
Merepresentasikan diagnosa terhadap hewan.

- treatment : String  
Merepresentasikan treatment yang diperoleh hewan.

- dokter\_periksa : String  
Merepresentasikan nama dokter yang memeriksa hewan.

- keterangan : String  
Merepresentasikan data keterangan periksa lainnya.

- tgl\_edit : Date  
Merepresentasikan tanggal perubahan terhadap rekam medis hewan.

---

+ insertDataHistoriRM(id\_periksa : integer, tgl\_periksa : date, anamnesa : string, diagnosa : string, treatment : string, dokter\_periksa : string, keterangan : string, tgl\_edit : date)  
Operasi ini digunakan untuk memasukkan data histori rekam medis ke database.

+getDataHistoriRM()  
Operasi ini digunakan untuk mengambil data histori rekam medis dari database.

### 3 Perancangan Data

#### 3.1 Dekomposisi Data

##### 3.1.1 Deskripsi Entitas Data Account

Nama	Tipe	Panjang	Keterangan
Id_Account	Integer	-	Id Account, Primary Key
Username	Varchar	80	Username account
Password	Varchar	8	Password account
Email	Varchar	60	Email account
Tgl_lahir	Datetime	-	Tanggal lahir pembuat account
Id_role	Integer	-	Role dari account, Foreign key ke tabel Role

##### 3.1.2 Deskripsi Entitas Data Role

Nama	Tipe	Panjang	Keterangan
Id_role	Integer	-	Id role, Primary Key
User_role	Varchar	30	Role dari user

##### 3.1.3 Deskripsi Entitas Data Dokter

Nama	Tipe	Panjang	Keterangan
Id_account	Integer	-	Id account dokter
Nama_dokter	Varchar	80	Nama dokter, Primary Key
Alamat_dokter	Varchar	100	Alamat dokter
Telp_dokter	Integer	20	Nomor telepon dokter

##### 3.1.4 Deskripsi Entitas Data Rekam Medis

Nama	Tipe	Panjang	Keterangan
Id_periksa	Integer	-	Id pemeriksaan, Primary Key
Id_rm	Varchar	10	Id hewan, Foreign Key ke tabel Hewan
Tgl_Periksa	Datetime	-	Tanggal periksa hewan
Anamnesa	Varchar	400	Anamnesa hewan
Diagnosa	Varchar	400	Diagnosa hewan
Treatment	Varchar	500	Treatment terhadap hewan

Dokter_periksa	Varchar	80	Nama dokter yang memeriksa hewan, Foreign Key ke tabel Dokter
Keterangan	Varchar	400	Keterangan tambahan hewan
Biaya	Integer	-	Biaya pemeriksaan hewan

### 3.1.5 Deskripsi Entitas Data Hewan

Nama	Tipe	Panjang	Keterangan
Id_rm	Varchar	10	Id hewan, Primary key
Id_pemilik	Varchar	10	Id pemilik hewan, foreign key ke tabel Pemilik Hewan
Tgl_Daftar	Datetime	-	Tanggal pendaftaran hewan
Jenis_hewan	Varchar	15	Jenis hewan
Ras_hewan	Varchar	100	Ras hewan
Nama_hewan	Varchar	100	Nama hewan
Jenis_kelamin	Varchar	10	Jenis kelamin hewan
Umur	Decimal	-	Umur hewan
Berat_badan	Integer	-	Berat badan hewan
Ciri_khusus	Varchar	200	Ciri khusus hewan

### 3.1.6 Deskripsi Entitas Data Pemilik Hewan

Nama	Tipe	Panjang	Keterangan
Id_pemilik	Varchar	10	Id pemilik hewan, Primary Key
Nama_pemilik	Varchar	100	Nama pemilik
Alamat	Varchar	100	Alamat pemilik hewan
No_telp	Integer	-	Nomor telepon pemilik hewan

### 3.1.7 Deskripsi Entitas Data Penitipan Hewan

Nama	Tipe	Panjang	Keterangan
Id_penitipan	Integer	-	Id penitipan hewan, Primary Key

Id_rm	Varchar	10	Id hewan, Foreign Key ke tabel Hewan
Tgl_titip	Datetime	-	Tanggal dititipkannya hewan
Tgl_ambil	Datetime	-	Tanggal diambilnya hewan
Jenis_penitipan	Varchar	15	Jenis penitipan hewan
Perkiraan_biaya	Integer	-	Perkiraan biaya penitipan hewan
Pengobatan	Varchar	400	Pengobatan yang dilakukan selama penitipan
Tindakan	Varchar	400	Tindakan yang dilakukan selama penitipan
DP	Integer	-	Uang muka pembayaran penitipan hewan

### 3.1.8 Deskripsi Entitas Data Grooming

Nama	Tipe	Panjang	Keterangan
Id_grooming	Integer	-	Id grooming, Primary Key
Id_rm	Varchar	10	Id hewan, Foreign Key ke tabel Hewan
Tgl_grooming	Datetime	-	Tanggal grooming
Jenis_grooming	Varchar	50	Jenis grooming

### 3.1.9 Deskripsi Entitas Histori Rekam Medis

Nama	Tipe	Panjang	Keterangan
Id_periksa	Integer	-	Id pemeriksaan, Primary Key
Id_rm	Varchar	10	Id hewan
Tgl_Periksa	Datetime	-	Tanggal periksa hewan
Anamnesa	Varchar	400	Anamnesa hewan

Diagnosa	Varchar	400	Diagnosa hewan
Treatment	Varchar	500	Treatment terhadap hewan
Dokter_periksa	Varchar	80	Nama dokter yang memeriksa hewan
Keterangan	Varchar	400	Keterangan tambahan hewan
Tgl_edit	Datetitime	-	Tanggal perubahan terhadap data rekam medis

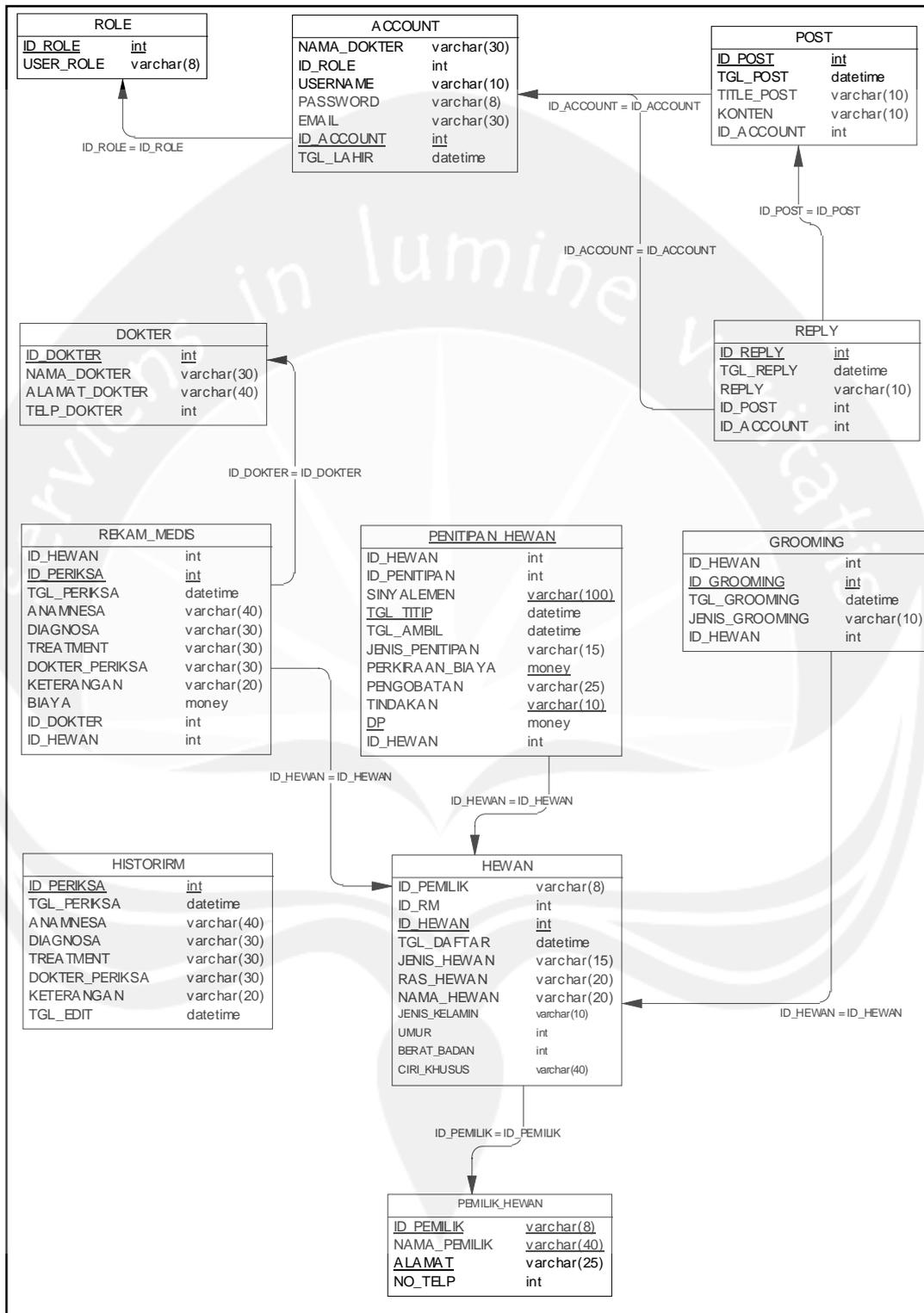
### 3.1.10 Deskripsi Entitas Data Post

Nama	Tipe	Panjang	Keterangan
Id_post	Integer	-	Id grooming, Primary Key
Id_account	Integer	-	Id account, Foreign Key ke tabel Account
Title	Varchar	100	Judul post
Tgl_post	Datetime	-	Tanggal post
Konten	Varchar	400	Isi dari post

### 3.1.11 Deskripsi Entitas Data Reply

Nama	Tipe	Panjang	Keterangan
Id_reply	Integer	-	Id reply, Primary Key
Id_post	Integer	-	Id post, Foreign Key ke tabel Post
Id_account	Integer	-	Id account, Foreign Key ke tabel Account
Tgl_reply	Datetime	-	Tanggal reply
Reply	Varchar	400	Isi dari post reply

### 3.2 Physical Data Model



Gambar 3.2 Physical Data Model

## 4 Perancangan Antarmuka

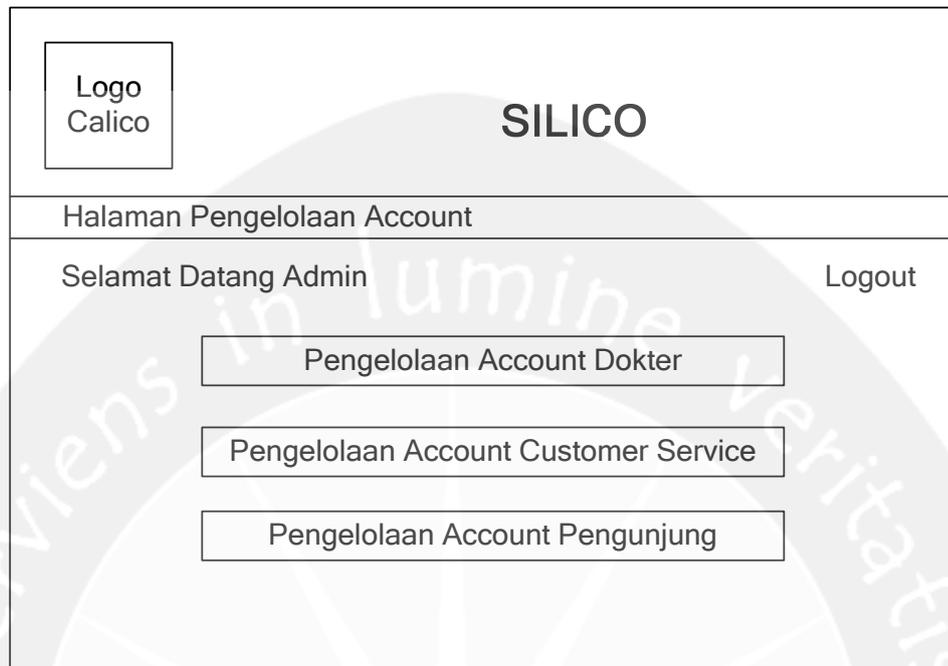
### 4.1 Antarmuka Login

Logo Calico	<b>SILICO</b>
Halaman Login	
Username	<input type="text"/>
Password	<input type="text"/>
<input type="button" value="Login"/>	<input type="button" value="Cancel"/>

Gambar 4.1 Rancangan Antarmuka Login

Antarmuka ini digunakan untuk melakukan proses login ke dalam sistem. Untuk mendapat akses masuk ke dalam sistem, user harus menginputkan username dan password dengan benar pada textbox yang telah disediakan. Pada saat tombol Login ditekan, sistem akan mengecek user id dan password yang diinputkan dengan data id dan password yang telah tersimpan di database. Jika data id dan password benar atau cocok maka user akan masuk ke dalam system sesuai dengan perannya, sebaliknya jika id dan password salah atau tidak cocok maka akan diberikan pesan peringatan. Jika user akan keluar dari sistem maka dapat dilakukan dengan menekan Cancel.

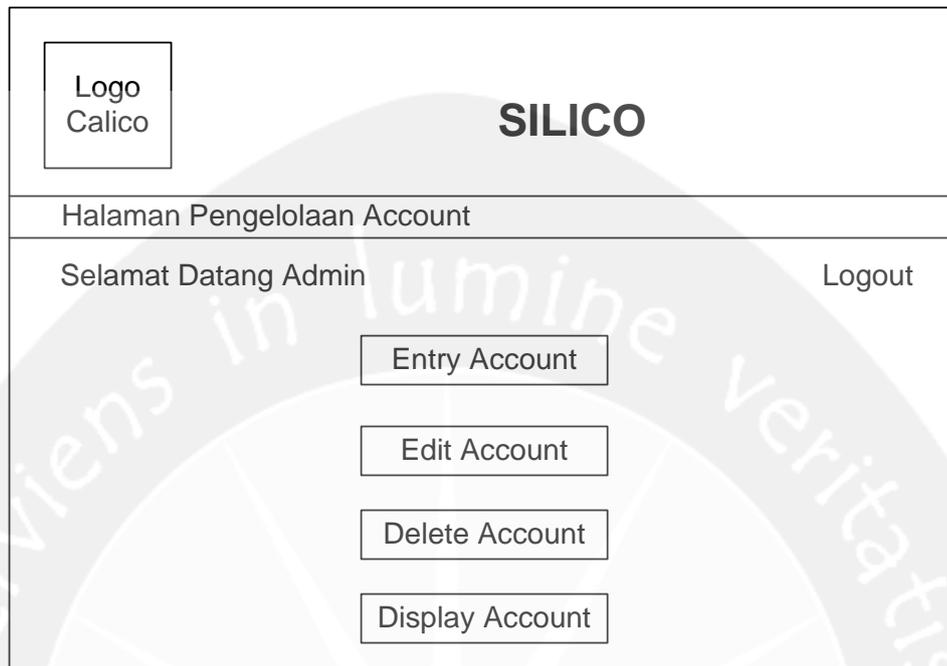
#### 4.2 Antarmuka Main Page Administrator



Gambar 4.2 Rancangan Antarmuka Main Page Administrator -  
Pengelolaan Account

Antarmuka ini digunakan untuk menampilkan fungsi-fungsi pengelolaan account yang ada pada sistem. Main page ini hanya dapat diakses setelah user dengan role admin melakukan login. Administrator dapat melakukan pengelolaan account dokter, customer service, dan pengunjung melalui antarmuka ini. Apabila tombol Pengelolaan Account Dokter ditekan maka menu pengelolaan account dokter akan ditampilkan, apabila tombol Pengelolaan Account Customer Service ditekan maka menu pengelolaan account customer service akan ditampilkan, apabila tombol Pengelolaan Account Pengunjung ditekan maka menu pengelolaan account pengunjung akan dan ditampilkan. Jika user akan keluar dari sistem maka cukup dengan menekan Logout pada sisi kanan atas halaman.

#### 4.2.1 Antarmuka Pengelolaan Account Dokter



**Gambar 4.3 Rancangan Antarmuka Main Page Administrator -  
Pengelolaan Account Dokter**

Antarmuka ini digunakan untuk menampilkan fungsi-fungsi pengelolaan account dokter yang ada pada menu Pengelolaan Account Dokter. Main page ini hanya dapat diakses setelah user dengan role admin melakukan login. Administrator dapat melakukan pengelolaan account dokter melalui antarmuka ini. Apabila tombol Entry Account ditekan maka form Entry Account akan ditampilkan, apabila tombol Edit Account ditekan maka form Edit Account akan ditampilkan, apabila tombol Delete Account ditekan maka antarmuka Delete Account akan dan ditampilkan, dan apabila tombol Display Account ditekan maka data seluruh account akan ditampilkan. Jika user akan keluar dari sistem maka cukup dengan menekan Logout pada sisi kanan atas halaman.

#### 4.2.1.1 Antarmuka Entry Data Account Dokter

The screenshot shows a web interface for the SILICO system. At the top left is a box labeled 'Logo Calico'. To its right is the title 'SILICO'. Below the title is a subtitle 'Halaman Pengelolaan Account / Entry Account'. The main area contains a form with the following fields: 'Username' (text input), 'Password' (text input), 'Role' (dropdown menu), 'Id Dokter' (text input), 'Nama' (text input), 'Alamat' (text input), and 'Nomor Telepon' (text input). At the bottom of the form are two buttons: 'Submit' and 'Cancel'.

Gambar 4.4 Rancangan Antarmuka Pengelolaan Account Dokter - Entry Data Account Dokter

Antarmuka ini digunakan administrator untuk menginputkan data account dokter yang baru. Disini admin dapat mengisikan nama, password, dan role dari account yang akan dibuat. Jika form telah selesai diisi kemudian tekan tombol Submit untuk memasukkan data ke dalam database. Sebelum data dimasukkan ke dalam database, data akan divalidasi terlebih dahulu. Jika ada kesalahan dalam inputan data maupun validasi yang benar, maka akan muncul pesan peringatan. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.2.1.2 Antarmuka Edit Data Account Dokter

The screenshot shows a web application interface for editing doctor account data. At the top left, there is a box labeled 'Logo Calico'. To its right, the word 'SILICO' is displayed in a large, bold font. Below this, a horizontal line separates the header from the sub-header, which reads 'Halaman Pengelolaan Account / Edit Account'. The main content area is a form with several input fields and buttons. At the top of the form, there is a label 'Id Dokter' followed by a text input field and a 'Search' button. Below this, there is a larger container with a light gray background containing several more input fields: 'Username', 'Password', 'Role' (with a dropdown arrow), 'Nama', 'Alamat', and 'Nomor Telepon'. At the bottom of this container, there are two buttons: 'Submit' and 'Cancel'.

**Gambar 4.5 Rancangan Antarmuka Pengelolaan Account Dokter - Edit Data Account Dokter**

Antarmuka ini digunakan administrator untuk mengedit data account dokter. Pertama user diminta untuk menginputkan id dokter lalu tombol Search ditekan untuk mencari data dalam database. Jika data ditemukan maka data account akan tertampil di field box dan muncul pesan peringatan, lalu data dapat diedit. Setelah itu tekan tombol Submit untuk menyimpan data editan. Jika validasi benar maka data akan disimpan dalam database dan muncul pesan peringatan. Jika ada kesalahan dalam inputan data, maka akan muncul pesan

peringatan juga. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.2.1.3 Antarmuka Delete Data Account Dokter

Logo Calico		<b>SILICO</b>				
Halaman Pengelolaan Account / Delete Account Dokter						
Username	Email	Id Dokter	Nama	Alamat	Telepon	
_____	_____	_____	_____	_____	_____	Delete
_____	_____	_____	_____	_____	_____	Delete
_____	_____	_____	_____	_____	_____	Delete
						<input type="button" value="Cancel"/>

**Gambar 4.6 Rancangan Antarmuka Pengelolaan Account Dokter - Delete Data Account Dokter**

Antarmuka ini digunakan administrator untuk menghapus data account dokter. User cukup menekan link delete pada data yang akan dihapus. Untuk membatalkan/kembali ke menu utama tekan Cancel.

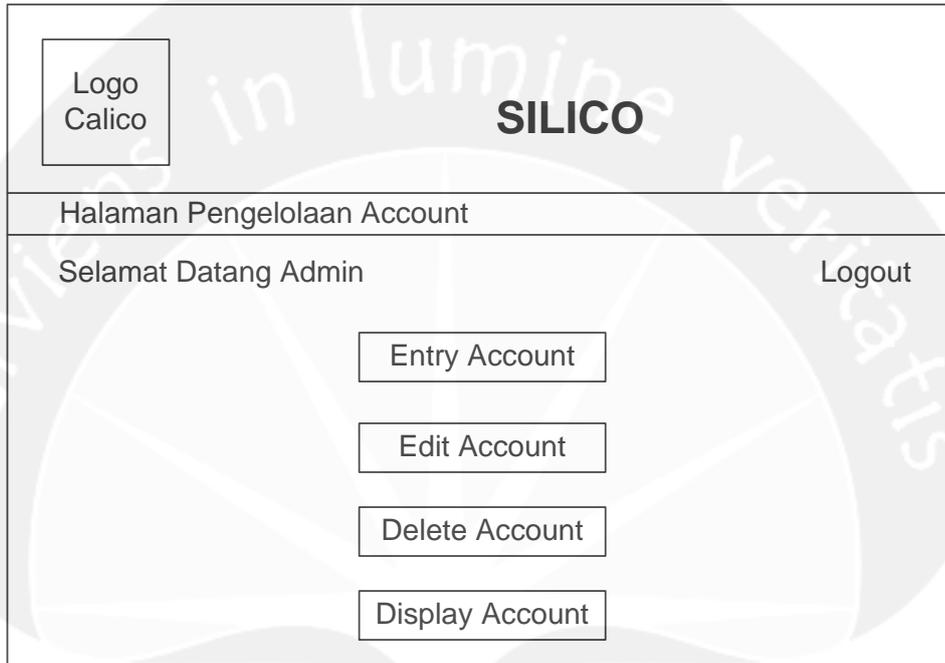
#### 4.2.1.4 Antarmuka Display Account Dokter

Logo Calico		<b>SILICO</b>				
Halaman Pengelolaan Account / Display Account Dokter						
Username	Email	Id Dokter	Nama	Alamat	Telepon	
_____	_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	_____	
						<input type="button" value="Cancel"/>

**Gambar 4.7 Rancangan Antarmuka Pengelolaan Account Dokter - Display Data Account Dokter**

Antarmuka ini digunakan administrator untuk menampilkan data account dokter. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.2.2 Antarmuka Pengelolaan Account Customer Service



Gambar 4.8 Rancangan Antarmuka Main Page Administrator - Pengelolaan Account Customer Service

Antarmuka ini digunakan untuk menampilkan fungsi-fungsi pengelolaan account customer service yang ada pada menu Pengelolaan Account Customer Service. Main page ini hanya dapat diakses setelah user dengan role customer service melakukan login. Administrator dapat melakukan pengelolaan account dokter melalui antarmuka ini. Apabila tombol Entry Account ditekan maka form Entry Account akan ditampilkan, apabila tombol Edit Account ditekan maka form Edit Account akan ditampilkan, apabila tombol Delete Account ditekan maka antarmuka Delete Account akan dan ditampilkan, dan

apabila tombol Display Account ditekan maka data seluruh account akan ditampilkan. Jika user akan keluar dari sistem maka cukup dengan menekan Logout pada sisi kanan atas halaman.

#### 4.2.2.1 Antarmuka Entry Data Account Customer Service

The image shows a web interface for account management. At the top left is a box labeled 'Logo Calico'. To its right is the text 'SILICO'. Below this is a header 'Halaman Pengelolaan Account / Entry Account'. The main area contains a form with two input fields: 'Username' and 'Password'. Below the form are two buttons: 'Submit' and 'Cancel'.

Gambar 4.9 Rancangan Antarmuka Pengelolaan Account Customer Service - Entry Data Account Customer Service

Antarmuka ini digunakan administrator untuk menginputkan data account customer service yang baru. Disini admin dapat mengisikan username dan password dari account yang akan dibuat. Jika form telah selesai diisi kemudian tekan tombol Submit untuk memasukkan data ke dalam database. Sebelum data dimasukkan ke dalam database, data akan divalidasi terlebih dahulu. Jika ada kesalahan dalam inputan data maupun validasi yang benar, maka akan muncul pesan peringatan. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.2.2.2 Antarmuka Edit Data Account Customer Service

Logo Calico	<b>SILICO</b>
Halaman Pengelolaan Account / Edit Account	
Id Account CS	<input type="text"/>
	<input type="button" value="Search"/>
Username	<input type="text"/>
Password	<input type="text"/>
	<input type="button" value="Update"/> <input type="button" value="Cancel"/>

Gambar 4.10 Rancangan Antarmuka Pengelolaan Account Customer Service - Edit Data Account Customer Service

Antarmuka ini digunakan administrator untuk mengedit data account customer service. Pertama user diminta untuk menginputkan id customer service lalu tombol Search ditekan untuk mencari data dalam database. Jika data ditemukan maka data account akan tertampil di field box dan muncul pesan peringatan, lalu data dapat diedit. Setelah itu tekan tombol Update untuk menyimpan data editan. Jika validasi benar maka data akan disimpan dalam database dan muncul pesan peringatan. Jika ada kesalahan dalam inputan data, maka akan muncul pesan peringatan juga. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.2.2.3 Antarmuka Delete Data Account Customer Service

Logo Calico	<b>SILICO</b>	
Halaman Pengelolaan Account / Delete Account		
ID	Username	
_____	_____	Delete
_____	_____	Delete
_____	_____	Delete
		Cancel

Gambar 4.11 Rancangan Antarmuka Pengelolaan Account Customer Service - Delete Data Account Customer Service

Antarmuka ini digunakan administrator untuk menghapus data account customer service. User cukup menekan link delete pada data yang akan dihapus. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.2.2.4 Antarmuka Display Account Customer Service

Logo Calico	<b>SILICO</b>	
Halaman Pengelolaan Account / Display Account		
ID	Username	
_____	_____	
_____	_____	
_____	_____	
		Cancel

Gambar 4.12 Rancangan Antarmuka Pengelolaan Account Customer Service - Display Data Account Customer Service

Antarmuka ini digunakan administrator untuk menampilkan data account customer service. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.2.3 Antarmuka Display Account Pengunjung



Logo Calico	SILICO	
Halaman Pengelolaan Account / Display Account Pengunjung		
Username	Email	
_____	_____	
_____	_____	
_____	_____	
		Cancel

Gambar 4.13 Rancangan Antarmuka Pengelolaan Account Pengunjung - Display Data Account Pengunjung

Antarmuka ini digunakan administrator untuk menampilkan data account pengunjung. Untuk membatalkan/kembali ke menu utama tekan Cancel.

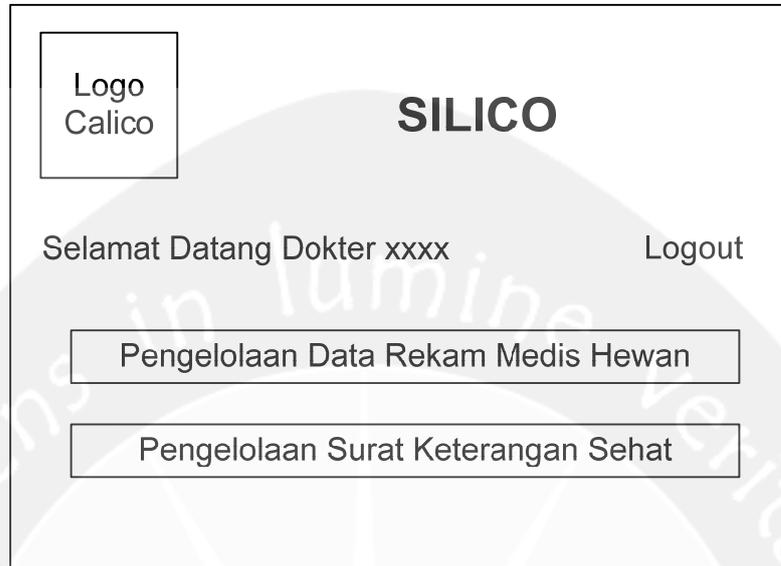
#### 4.2.4 Antarmuka Delete Thread Post

Logo Calico	<b>SILICO</b>		
Halaman Pengelolaan Forum			
ID	Title	Konten	
_____	_____	_____	Delete
_____	_____	_____	Delete
_____	_____	_____	Delete
			<input type="button" value="Cancel"/>

Gambar 4.14 Rancangan Antarmuka Pengelolaan Forum - Delete Thread Post

Antarmuka ini digunakan administrator untuk menghapus thread post yang bermasalah. Klik tombol Delet pada baris data yang akan dihapus, maka data akan terhapus dari database. Untuk membatalkan/kembali ke menu utama tekan Cancel.

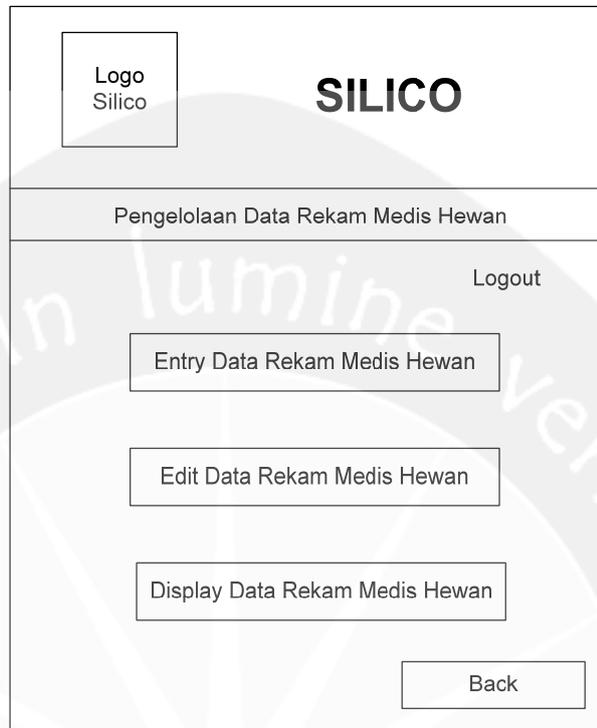
### 4.3 Antarmuka Main Page Dokter



Gambar 4.15 Rancangan Antarmuka Main Page Dokter

Antarmuka ini hanya dapat diakses oleh user dengan role dokter. Apabila tombol Pengelolaan Data Rekam Medis Hewan ditekan maka menu Pengelolaan Data Rekam Medis Hewan akan ditampilkan, apabila tombol Pengelolaan Surat Keterangan Sehat ditekan maka menu Surat Keterangan Sehat akan ditampilkan. Untuk keluar dari sistem tekan Logout.

#### 4.3.1 Antarmuka Pengelolaan Data Rekam Medis Hewan



**Gambar 4.16 Rancangan Antarmuka Main Page Dokter -  
Pengelolaan Rekam Medis Hewan**

Antarmuka ini hanya dapat diakses oleh user dengan role dokter yang berisi menu pengelolaan rekam medis hewan. Apabila tombol Entry Data Rekam Medis Hewan ditekan maka form Entry Data Rekam Medis Hewan akan ditampilkan, apabila tombol Edit Data Rekam Medis Hewan ditekan maka form Edit Data Rekam Medis Hewan akan ditampilkan, apabila tombol Display Data Rekam Medis Hewan ditekan maka form Display Data Rekam Medis Hewan akan ditampilkan. Untuk keluar dari sistem tekan Logout.

#### 4.3.1.1 Antarmuka Entry Data Rekam Medis Hewan

The screenshot shows the 'SILICO' web application interface for 'Entry Data Rekam Medis Hewan'. The interface is divided into several sections:

- Header:** Contains the 'Logo Calico' and the title 'SILICO'.
- Section Header:** 'Entry Data Rekam Medis Hewan'.
- Form Fields:**
  - Left Column:** Id RM, Tanggal (with a dropdown arrow), Nama Pemilik, Alamat Pemilik, Telepon, Jenis Hewan, Nama Hewan, Ras, Kelamin (with a dropdown arrow), and Umur.
  - Right Column:** Ciri Khusus, Berat Badan, Anamnesa, Diagnosa, Treatment, and Dokter Periksa (with a dropdown arrow).
- Buttons:** 'Submit' and 'Cancel' buttons are located at the bottom right of the form.

**Gambar 4.17 Rancangan Antarmuka Pengelolaan Data Rekam Medis Hewan - Entry Data Rekam Medis Hewan**

Antarmuka ini digunakan dokter untuk mengisi data rekam medis hewan. Jika form telah selesai diisi, tekan tombol Submit untuk menyimpan data ke dalam database. Jika validasi benar maka data akan tersimpan ke database. Jika ada kesalahan dalam inputan data maupun validasinya benar, maka akan muncul pesan peringatan. Tekan Cancel untuk membatalkan pengisian form dan kembali ke menu utama dokter.

#### 4.3.1.2 Antarmuka Edit Data Rekam Medis Hewan

The screenshot shows the 'SILICO' application interface for editing animal medical records. At the top left is the 'Logo Calico' and the title 'SILICO'. Below this is the sub-header 'Edit Data Rekam Medis Hewan'. The main area contains a search bar for 'Id RM' with a 'Search' button. Below the search bar is a large form with two columns of input fields. The left column includes: 'Id RM', 'Tanggal' (with a dropdown arrow), 'Nama Pemilik', 'Alamat Pemilik', 'Telepon', 'Jenis Hewan', 'Nama Hewan', 'Ras', 'Kelamin' (with a dropdown arrow), and 'Umur'. The right column includes: 'Ciri Khusus', 'Berat Badan', 'Anamnesa', 'Diagnosa', 'Treatment', and 'Dokter Periksa' (with a dropdown arrow). At the bottom of the form are 'Save' and 'Cancel' buttons.

**Gambar 4.18 Rancangan Antarmuka Pengelolaan Data Rekam Medis Hewan  
- Edit Data Rekam Medis Hewan**

Antarmuka ini digunakan dokter untuk mengubah data rekam medis hewan. Pertama user mengisi Id rekam medis yang akan diubah, lalu tombol Search ditekan maka sistem akan mencari data dalam database. Jika data ditemukan akan ditampilkan di field box untuk kemudian dapat diubah. Untuk menyimpan data yang telah diupdate tekan tombol Save. Jika validasi benar maka data akan tersimpan ke database. Jika ada kesalahan dalam inputan data maupun validasinya benar, maka akan muncul pesan peringatan. Untuk membatalkan/kembali ke menu utama dokter tekan Cancel.

#### 4.3.1.3 Antarmuka Display Data Rekam Medis Hewan

The screenshot shows a web application interface titled "SILICO" with a "Logo Calico" in the top left. The main heading is "Display Data Rekam Medis Hewan". Below this, there is a search section with an input field for "Id RM" and a "Search" button. A large form is centered on the page, containing various input fields for patient details:

- Id RM:
- Ciri Khusus:
- Tanggal:
- Berat Badan:
- Nama Pemilik:
- Anamnesa:
- Alamat Pemilik:
- Diagnosa:
- Telepon:
- Treatment:
- Jenis Hewan:
- Nama Hewan:
- Ras:
- Kelamin:
- Dokter Periksa:
- Umur:

A "Cancel" button is located at the bottom right of the form.

Gambar 4.19 Rancangan Antarmuka Pengelolaan Data Rekam Medis Hewan - Display Data Rekam Medis Hewan

Antarmuka ini digunakan dokter untuk menampilkan data rekam medis hewan. Pertama user mengisi Id rekam medis yang akan ditampilkan datanya, lalu tombol Search ditekan maka sistem akan mencari data dalam database. Kemudian data akan dimunculkan di tabel. Jika user memilih salah satu data pemeriksaan maka detailnya akan ditampilkan di field box. Untuk membatalkan/kembali ke menu utama dokter tekan Cancel.

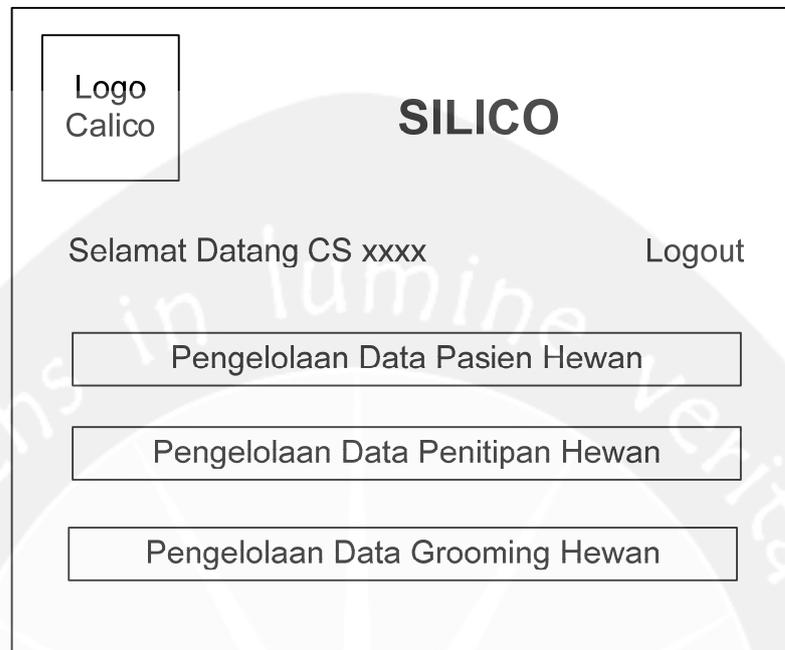
#### 4.3.2 Antarmuka Display Surat Keterangan Sehat

Logo Calico	<b>SILICO</b>
Surat Keterangan Sehat	
Dengan ini saya Dr. .... dari Calico menyatakan bahwa memberikan Surat Keterangan Sehat kepada :	
Nama Pemilik	<input type="text"/>
Alamat	<input type="text"/>
Nama Hewan	<input type="text"/>
Jenis Hewan	<input type="text"/>
Jenis Kelamin	<input type="text"/>
Umur	<input type="text"/>
Ciri Khusus	<input type="text"/>
<input type="button" value="Cetak"/> <input type="button" value="Back"/>	

Gambar 4.20 Rancangan Antarmuka Display Surat Keterangan Sehat

Antarmuka ini digunakan untuk mencetak surat keterangan sehat bagi hewan. Data akan diambil dari tabel hewan, pemilik hewan, dan rekam medis hewan untuk kemudian ditampilkan dalam form. Jika belum ada data hewan di klinik, maka data akan dimasukkan satu per satu ke form. Untuk mencetak form tekan tombol Cetak. Untuk membatalkan/kembali ke menu utama dokter tekan Back.

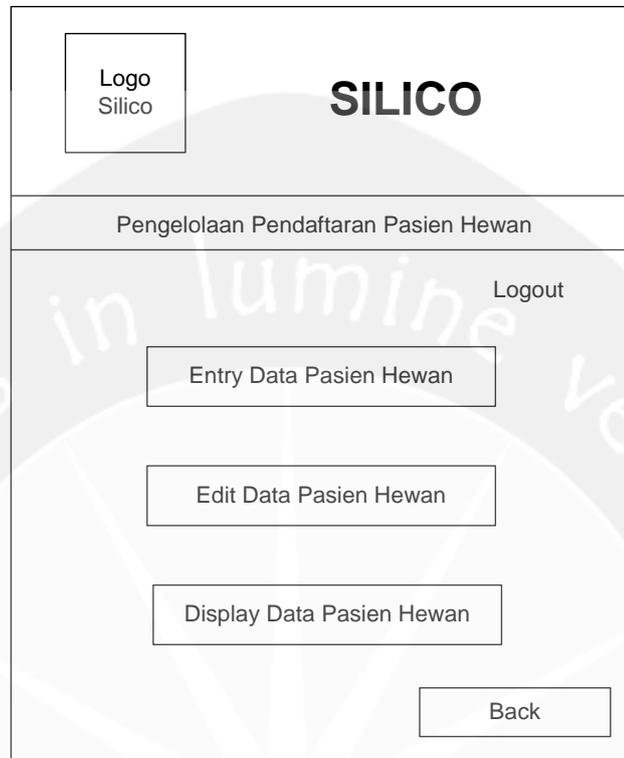
#### 4.4 Antarmuka Main Page Customer Service



Gambar 4.21 Rancangan Antarmuka Main Page Customer Service

Antarmuka ini hanya dapat diakses oleh user dengan role customer service. Apabila tombol Pengelolaan Data Pasien Hewan ditekan maka menu pengelolaan data pasien hewan akan ditampilkan, apabila tombol Pengelolaan Data Penitipan Hewan ditekan maka menu pengelolaan data penitipan hewan akan ditampilkan, apabila tombol Pengelolaan Data Grooming Hewan ditekan maka menu pengelolaan data grooming hewan akan ditampilkan. Untuk keluar dari sistem tekan Logout.

#### 4.4.1 Antarmuka Pengelolaan Pendaftaran Pasien Hewan



Gambar 4.22 Rancangan Antarmuka Main Page Customer Service - Pengelolaan Pendaftaran Pasien Hewan

Antarmuka ini hanya dapat diakses oleh user dengan role customer service yang berisi menu pengelolaan pendaftaran pasien hewan. Apabila tombol Entry Data Pasien Hewan ditekan maka form Entry Data Pendaftaran Pasien Hewan ditekan akan ditampilkan, apabila tombol Edit Data Pasien Hewan ditekan maka form Edit Data Pasien Hewan akan ditampilkan, apabila tombol Display Data Pasien Hewan ditekan maka form Display Data Pasien Hewan akan ditampilkan. Untuk keluar dari sistem tekan Logout. Untuk kembali ke main page dokter, tekan Back.

#### 4.4.1.1 Antarmuka Entry Data Pasien Hewan

Logo Calico	<b>SILICO</b>
Entry Data Pasien Hewan	
Tanggal Daftar	<input type="text"/>
Nama Pemilik	<input type="text"/>
Alamat	<input type="text"/>
Nama Hewan	<input type="text"/>
Jenis Hewan	<input type="text"/>
<input type="button" value="Submit"/>	<input type="button" value="Cancel"/>

**Gambar 4.23 Rancangan Antarmuka Pengelolaan Pendaftaran Pasien Hewan - Entry Data Pasien Hewan**

Antarmuka ini digunakan dokter untuk menginputkan data pasien hewan baru. Disini customer service dapat mengisikan tanggal daftar, nama pemilik, nama hewan, nama hewan, jenis, dan diagnosa. Jika form telah selesai diisi kemudian tekan tombol Submit untuk memasukkan data ke dalam database. Sebelum data dimasukkan ke dalam database, data akan divalidasi terlebih dahulu. Jika validasi benar maka data akan tersimpan ke database. Jika ada kesalahan dalam inputan data maupun validasi yang benar, maka akan muncul pesan

peringatan. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.4.1.2 Antarmuka Edit Data Pasien Hewan

The screenshot shows a web interface for editing animal patient data. At the top left is a box labeled 'Logo Calico'. To its right is the text 'SILICO'. Below this is a horizontal bar with the text 'Edit Data Pasien Hewan'. The main content area is enclosed in a large rectangular frame. Inside this frame, at the top, is a label 'Id Hewan' next to an empty input field. Below this is a 'Search' button. Further down, there is a sub-container with five rows, each containing a label and an input field: 'Tanggal Daftar', 'Nama Pemilik', 'Alamat', 'Nama Hewan', and 'Jenis Hewan'. At the bottom of the sub-container are two buttons: 'Save' and 'Cancel'.

Gambar 4.24 Rancangan Antarmuka Pengelolaan Pendaftaran Pasien Hewan - Edit Data Pasien Hewan

Antarmuka ini digunakan customer service untuk mengubah data pasien hewan. Pertama user diminta untuk menginputkan id hewan lalu tombol Search ditekan untuk

mencari data dalam database. Jika data ditemukan maka data hewan akan tertampil di field box. Untuk mengupdate data tekan tombol Save. Jika validasi benar maka data akan tersimpan ke database. Jika ada kesalahan dalam inputan data maupun validasi yang benar, maka akan muncul pesan peringatan. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.4.1.3 Antarmuka Display Data Pasien Hewan

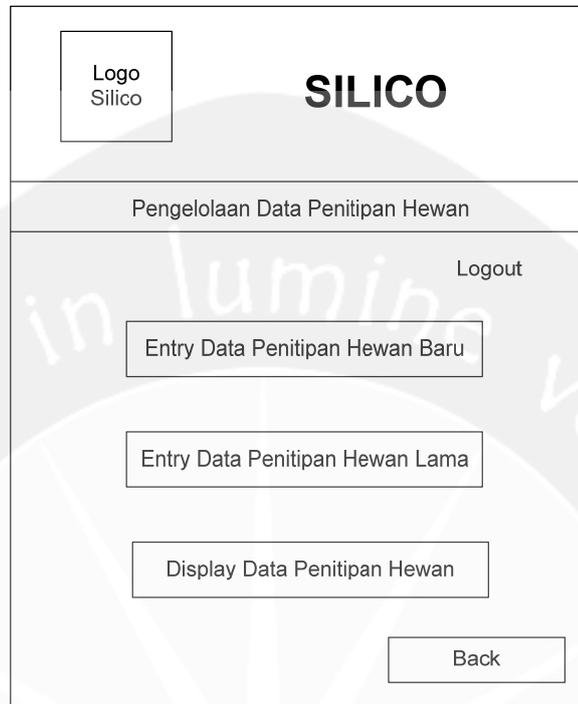
Display Data Pasien Hewan					
Id Hewan	Tanggal Daftar	Nama Pemilik	Alamat	Nama Hewan	Jenis Hewan
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

Back

**Gambar 4.25 Rancangan Antarmuka Pengelolaan Pendaftaran Pasien Hewan - Display Data Pasien Hewan**

Antarmuka ini digunakan customer service untuk menampilkan data pendaftaran pasien hewan. Untuk membatalkan/kembali ke menu utama tekan Back.

#### 4.4.2 Antarmuka Pengelolaan Data Penitipan Hewan



**Gambar 4.26 Rancangan Antarmuka Main Page Customer Service -  
Pengelolaan Data Penitipan Hewan**

Antarmuka ini digunakan untuk melakukan pengelolaan data penitipan hewan. Dapat digunakan oleh user dengan role customer service. Bila Entry Data Penitipan Hewan dipilih maka form Entry Data Penitipan Hewan Lama akan ditampilkan, jika Entry Data Penitipan Hewan Baru dipilih maka form Entry Data Penitipan Hewan Baru akan ditampilkan, jika Display Data Penitipan Hewan dipilih, maka data penitipan hewan akan ditampilkan. Untuk keluar dari sistem tekan Logout. Untuk kembali ke main page customer service tekan Back.

#### 4.4.2.1 Antarmuka Entry Data Penitipan Hewan

Logo Calico		SILICO	
Entry Data Penitipan Hewan Baru			
Jenis Penitipan	<input type="text"/>	Kelamin	<input type="text"/>
Tanggal Titip	<input type="text"/>	Umur	<input type="text"/>
Tanggal Ambil	<input type="text"/>	Ciri Khusus	<input type="text"/>
Nama Pemilik	<input type="text"/>	Berat Badan	<input type="text"/>
Alamat Pemilik	<input type="text"/>	Perkiraan Biaya	<input type="text"/>
Telepon	<input type="text"/>	Pengobatan	<input type="text"/>
Jenis Hewan	<input type="text"/>	Tindakan	<input type="text"/>
Nama Hewan	<input type="text"/>	Down Payment	<input type="text"/>
Ras	<input type="text"/>		
		<input type="button" value="Submit"/>	<input type="button" value="Cancel"/>

**Gambar 4.27 Rancangan Antarmuka Pengelolaan Data Penitipan Hewan – Entry Data Penitipan Hewan Baru**

Antarmuka ini dapat diakses oleh user dengan role customer service. Form ini digunakan untuk hewan yang belum memiliki data di Calico atau belum pernah datang ke Calico. User mengisikan form dengan lengkap kemudian tekan tombol Submit untuk menyimpan data ke dalam database. Jika validasi benar maka data akan tersimpan ke database. Jika ada kesalahan dalam inputan data maupun validasinya benar, maka akan muncul pesan peringatan. Untuk membatalkan pengisian form/untuk kembali ke menu utama tekan Cancel.

#### 4.4.2.2 Antarmuka Entry Data Penitipan Hewan Lama

The screenshot shows the 'SILICO' web application interface for 'Entry Data Penitipan Hewan Lama'. At the top left is the 'Logo Calico'. The main title is 'SILICO'. Below the title is the page header 'Entry Data Penitipan Hewan Lama'. The interface features a search bar for 'Id Hewan' with a 'Search' button. Below this is a large form containing the following fields:

Jenis Penitipan	<input type="text"/>	Kelamin	<input type="text"/>
Tanggal Titip	<input type="text"/>	Umur	<input type="text"/>
Tanggal Ambil	<input type="text"/>	Ciri Khusus	<input type="text"/>
Nama Pemilik	<input type="text"/>	Berat Badan	<input type="text"/>
Alamat Pemilik	<input type="text"/>	Perkiraan Biaya	<input type="text"/>
Telepon	<input type="text"/>	Pengobatan	<input type="text"/>
Jenis Hewan	<input type="text"/>	Tindakan	<input type="text"/>
Nama Hewan	<input type="text"/>	Down Payment	<input type="text"/>
Ras	<input type="text"/>		

At the bottom of the form are 'Submit' and 'Cancel' buttons.

**Gambar 4.28 Rancangan Antarmuka Pengelolaan Data Penitipan Hewan - Entry Data Penitipan Hewan Lama**

Antarmuka ini hanya dapat digunakan oleh user dengan role customer service. Form ini digunakan apabila hewan yang akan dititipkan sudah memiliki data di Calico. Pertama user menginputkan id hewan untuk kemudian dicari datanya dan ditampilkan di field box yang sesuai. Selanjutnya dokter hanya perlu melengkapi field yang masih kosong. Untuk menyimpan data ke database tekan tombol Submit. Jika validasi benar maka data akan tersimpan ke database. Jika ada kesalahan dalam inputan data maupun validasinya benar, maka akan muncul pesan peringatan. Untuk membatalkan pengisian form/kembali ke menu utama tekan Cancel.

#### 4.4.2.3 Antarmuka Display Data Penitipan Hewan

Logo Calico	<b>SILICO</b>					
Display Data Penitipan Hewan						
Id Hewan	Nama Hewan	Jenis Hewan	Nama Pemilik	Jenis Penitipan	Tanggal Titip	Tanggal Ambil
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
						Back

Gambar 4.29 Rancangan Antarmuka Pengelolaan Data Penitipan Hewan - Display Data Penitipan Hewan

Antarmuka ini digunakan untuk menampilkan data penitipan hewan, baik yang sedang dititipkan atau yang sudah diambil. Untuk membatalkan/kembali ke menu utama tekan Back.

#### 4.4.3 Antarmuka Pengelolaan Data Grooming

Logo Calico	<b>SILICO</b>
Pengelolaan Data Grooming	
	Logout
Entry Data Grooming Baru	
Entry Data Grooming Lama	
Display Data Grooming	
Cancel	

**Gambar 4.30 Rancangan Antarmuka Main Page Customer Service -  
Pengelolaan Data Grooming**

Antarmuka ini digunakan user customer service untuk mengelola data grooming. Jika tombol Entry Data Grooming Baru/Lama ditekan maka form entry data grooming akan muncul. Jika form Display Data Grooming ditekan maka halaman data grooming akan ditampilkan. Tekan Logout untuk keluar dari sistem.

#### 4.4.3.1 Antarmuka Entry Data Grooming Baru

The screenshot shows a web interface for entering new grooming data. At the top left is a box for the 'Logo Calico'. The main title is 'SILICO'. Below the title is the subtitle 'Entry Data Grooming Baru'. The form contains several input fields: 'Id Grooming' (text), 'Tanggal Grooming' (calendar icon), 'Jenis Grooming' (dropdown), 'Nama Hewan' (text), 'Jenis Hewan' (dropdown), 'Nama Pemilik' (text), 'Alamat Pemilik' (text), and 'Telepon' (text). At the bottom right are 'Submit' and 'Cancel' buttons.

Gambar 4.31 Rancangan Antarmuka Pengelolaan Data Grooming - Entry Data Grooming Baru

Antarmuka ini digunakan user customer service untuk memasukkan data hewan yang akan digrooming. Form ini digunakan ketika hewan yang digrooming merupakan hewan baru yang belum memiliki data di Calico. Untuk menyimpan data tekan Submit. Jika validasi benar maka data akan tersimpan ke database. Jika ada kesalahan dalam inputan data maupun validasinya benar, maka akan muncul pesan peringatan. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.4.3.2 Antarmuka Entry Data Grooming Lama

The screenshot shows a web application interface for entering grooming data for an old animal. The interface is titled "SILICO" and "Entry Data Grooming Lama". It features a search bar for "Id Hewan" and a "Search" button. Below this is a form with several input fields: "Id Grooming", "Tanggal Grooming", "Jenis Grooming", "Nama Hewan", "Jenis Hewan" (dropdown), "Nama Pemilik", "Alamat Pemilik", and "Telepon". There are "Submit" and "Cancel" buttons at the bottom right.

**Gambar 4.32 Rancangan Antarmuka Pengelolaan Data Grooming - Entry Data Grooming Lama**

Antarmuka ini digunakan user customer service untuk memasukkan data hewan yang akan digrooming. Form ini digunakan ketika hewan yang digrooming merupakan hewan lama yang sudah memiliki data di Calico. User akan menginputkan id hewan untuk kemudian dicari data nama dan jenisnya untuk kemudian ditampilkan di form. Untuk menyimpan data ke dalam database tekan Submit. Jika validasi benar maka data akan tersimpan ke database. Jika ada kesalahan dalam inputan data maupun validasinya benar, maka akan muncul pesan peringatan. Untuk membatalkan/kembali ke menu utama tekan Cancel.

#### 4.4.3.3 Antarmuka Display Data Grooming

Id Grooming	Id Hewan	Tanggal Grooming	Nama Hewan	Jenis Hewan	Jenis Grooming

Back

Gambar 4.33 Antarmuka Pengelolaan Data Grooming - Display Data Grooming

Antarmuka ini digunakan untuk menampilkan data grooming hewan. Untuk membatalkan/kembali ke menu utama tekan Back.

#### 4.5 Antarmuka Pendaftaran User

Username

Password

Email

Submit

Gambar 4.34 Rancangan Antarmuka Pendaftaran User

Antarmuka ini digunakan untuk user mendaftarkan diri menjadi anggota forum. Setelah data dalam form

diisi, tekan tombol Submit untuk memasukkan data dalam database.

#### 4.6 Antarmuka Pengelolaan Forum

##### 4.6.1 Antarmuka Entry Thread Baru

Tanggal Post	User	Judul	Konten

Judul

Konten

Gambar 4.35 Rancangan Antarmuka Pengelolaan Forum - Entry Thread Baru

Antarmuka ini digunakan oleh pengunjung yang telah mendaftar menjadi member forum. User dapat menambahkan thread baru dengan mengisi judul dan konten pada field kemudian menekan tombol Submit. Jika pengunjung belum mendaftar menjadi member, maka pengunjung hanya dapat membaca isi forum tanpa bisa menambahkan thread baru.

#### 4.6.2 Antarmuka Entry Thread Reply

Tanggal Post	User	Judul Post	Reply
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Reply

Submit

Gambar 4.36 Rancangan Antarmuka Pengelolaan Forum - Entry Thread Reply

Antarmuka ini digunakan oleh pengunjung yang telah mendaftar menjadi member forum. User dapat menambahkan reply untuk thread dengan mengisi reply pada field kemudian menekan tombol Submit. Jika pengunjung belum mendaftar menjadi member, maka pengunjung hanya dapat membaca isi forum tanpa bisa menambahkan reply baru.